

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.166

(11/2005)

SERIES J: CABLE NETWORKS AND TRANSMISSION
OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

IPCablecom

**IPCablecom Management Information Base
(MIB) framework**

ITU-T Recommendation J.166



ITU-T Recommendation J.166

IPCablecom Management Information Base (MIB) framework

Summary

This Recommendation describes the framework in which IPCablecom MIBs (Management Information Base) are defined. It provides information on the management requirements of IPCablecom-specified devices and functions and how these requirements are supported in the MIB. It is intended to support and complement the actual MIB Recommendations, which are included as annexes.

Source

ITU-T Recommendation J.166 was approved on 29 November 2005 by ITU-T Study Group 9 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2006

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

	Page
1 Scope	1
2 References.....	1
2.1 Normative references.....	1
2.2 Informative references.....	2
3 Terms and definitions	2
4 Abbreviations and conventions.....	2
4.1 Abbreviations	2
4.2 Conventions.....	2
5 Overview	3
5.1 IPCablecom reference architecture.....	3
5.2 General requirements.....	4
5.3 Functional requirements	7
6 MIBs available in a IPCablecom network.....	8
6.1 DOCSIS MIB modules.....	9
6.2 IF MIB.....	9
6.3 MIB II.....	9
6.4 Ethernet MIB.....	11
6.5 Annex C.....	11
6.6 Annex B.....	12
6.7 Event Management MIB	12
7 IPCablecom MIB module implementation.....	12
7.1 MTA components.....	13
7.2 MIB layering	13
8 Organization of the MIBs	14
8.1 Definition MIB	14
8.2 MTA MIB.....	14
8.3 Signalling MIB	15
8.4 MEM MIB.....	15
8.5 MTA Extension MIB.....	15
8.6 Signalling Extension MIB	15
8.7 Battery Backup MIB.....	15
Annex A – MIB import data	15
Annex B – MTA MIB.....	21
Annex C – Network Call Signalling MIB.....	49
Annex D – Management Event MIB	72
Annex E – Extension MTA MIB	83
Annex F – Signalling Extension MIB.....	85

ITU-T Recommendation J.166

IPCablecom Management Information Base (MIB) framework

1 Scope

This Recommendation describes the framework in which IPCablecom MIBs (Management Information Base) are defined. It provides information on the management requirements of IPCablecom-specified devices and functions and how these requirements are supported in the MIB. It is intended to support and complement the actual MIB Recommendations, which are included as annexes.

2 References

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- ITU-T Recommendation J.112 Annex A (2001), *Digital video broadcasting: DVB interaction channel for cable TV (CATV) distribution systems*.
- ITU-T Recommendation J.112 Annex B (2004), *Data-over-cable service interface specifications: Radio-frequency interface specification*.
- ITU-T Recommendation J.162 (2005), *Network call signalling protocol for the delivery of time-critical services over cable television networks using cable modems*.
- ITU-T Recommendation J.167 (2005), *Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems*.
- ITU-T Recommendation J.170 (2005), *IPcablecom security specification*.
- ITU-T Recommendation J.176 (2002), *IPcablecom management event mechanism MIB*.
- IETF RFC 1907 (1996), *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*.
- IETF RFC 2011 (1996), *SNMPv2 Management Information Base for the Internet Protocol using SMIPv2*.
- IETF RFC 2013 (1996), *SNMPv2 Management Information Base for the User Datagram Protocol using SMIPv2*.
- IETF RFC 2863 (2000), *The Interfaces Group MIB*.
- IETF RFC 2578 (1999), *Structure of Management Information Version 2 (SMIPv2)*.
- IETF RFC 2579 (1999), *Textual Conventions for SMIPv2*.
- ANSI/SCTE 23-3 (2003), *DOCSIS 1.1 Part 3: Operations Support System Interface*.
- ANSI/SCTE 79-2 (2002), *DOCS 2.0 Operations Support System Interface*.

2.2 Informative references

- ITU-T Recommendation J.160 (2005), *Architectural framework for the delivery of time-critical services over cable television networks using cable modems*.
- IETF RFC 1493 (1993), *Definitions of Managed Objects for Bridges*.
- IETF RFC 1643 (1994), *Definitions of Managed Objects for the Ethernet-like Interface Types*.
- IETF RFC 2571 (1999), *An Architecture for Describing SNMP Management Frameworks*.
- IETF RFC 2572 (1999), *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*.
- IETF RFC 2573 (1999), *SNMP Applications*.
- IETF RFC 2574 (1999), *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*.
- IETF RFC 2575 (1999), *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*.

3 Terms and definitions

This Recommendation defines the following terms:

3.1 cable modem: A cable modem is a layer-two termination device that terminates the customer end of the ITU-T J.112/J.122 connection.

3.2 IPCablecom: An ITU-T project that includes an architecture and a series of Recommendations that enable the delivery of real-time services over the cable television networks using cable modems.

3.3 management information base (MIB): The specification of information in a manner that allows standard access through a network management protocol.

4 Abbreviations and conventions

4.1 Abbreviations

This Recommendation uses the following abbreviations:

CM Cable Modem

MIB Management Information Base

MTA Media Terminal Adapter – contains the interface to a physical voice device, a network interface, CODECs, and all signalling and encapsulation functions required for VoIP transport, class features signalling, and QoS signalling.

NCS Network Call Signalling

QoS Quality of Service, guarantees network bandwidth and availability for applications.

4.2 Conventions

Throughout this Recommendation, 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 of this Recommendation.

"MUST NOT"	This phrase means that the item is an absolute prohibition of this Recommendation.
"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 weighed before choosing a different course.
"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behaviour is acceptable or event useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour 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 requires it or because it enhances the product, for example; another vendor may omit the same.

5 Overview

IPCablecom MIB modules are designed to provide necessary functionality defined in IPCablecom Recommendations. MIBs that are developed for IPCablecom support embedded Media Terminal Adapters (MTAs) and in most cases Stand-alone MTAs and provide definitions for call signalling and MTA device provisioning functions. Future IPCablecom development phases will include other functional areas as well as requirements for other IPCablecom components, which will be considered for MIB module development. IPCablecom functional areas that are being studied for future IPCablecom MIB definition include ITU-T Recs J.161, J.162, J.167, J.170 and J.176.

5.1 IPCablecom reference architecture

The conceptual diagram for the IPCablecom architecture is shown in Figure 1. Please refer to ITU-T Rec. J.160 for more detailed information concerning the IPCablecom architecture.

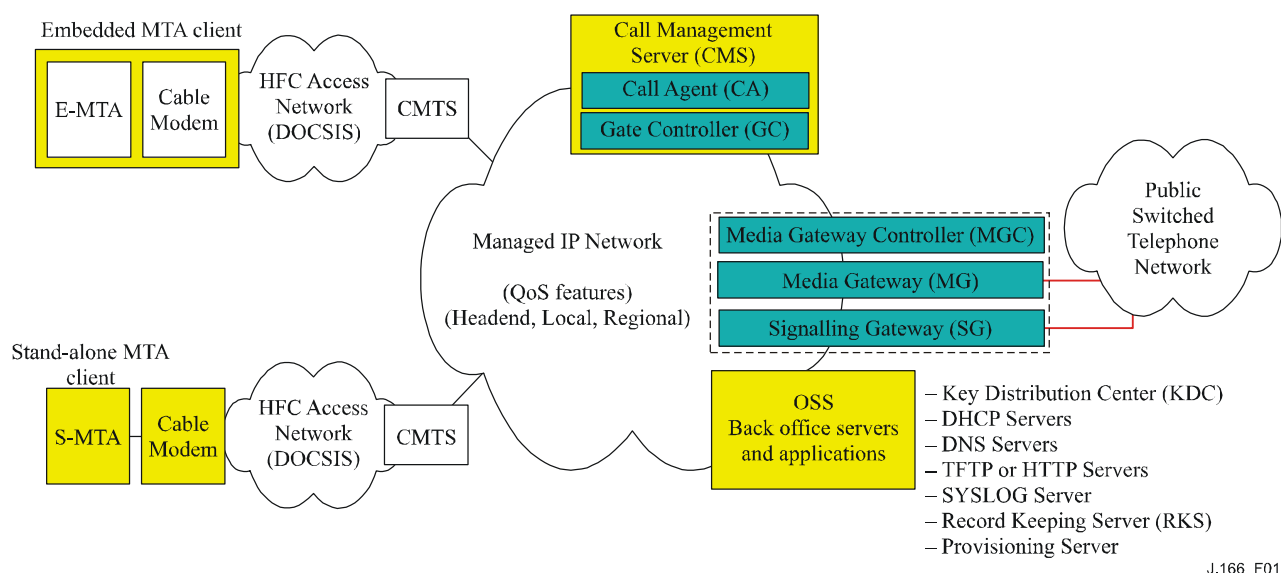


Figure 1/J.166 – IPCablecom network component reference model (partial)

5.2 General requirements

The IPCablecom MIBs Framework Recommendation follows the Internet Standard Management Framework described in RFC 3410. Additionally, the following requirements have been considered in the design of IPCablecom MIB modules:

- IPCablecom devices **MUST** be compliant with ITU-T Rec. J.112 or ITU-T Rec. J.122; Therefore, IPCablecom devices must support DOCSIS MIBs as defined in 6.1;
- Take a minimalist approach for design of the IPCablecom MIB modules, i.e., if other MIB modules define the same functions, then rely on these MIB modules rather than create new ones;
- Organize MIB modules to support both embedded and stand-alone MTAs;
- Organize MIB modules so as to allow functional partitioning of ITU-T Rec. J.112/J.122 (high-speed data) and IPCablecom (voice features);
- ITU-T Rec. J.112 or ITU-T Rec. J.122 within IPCablecom applications requires support of SNMPv3; therefore IPCablecom MIB agents **MUST** comply with SNMPv3;
- IPCablecom MIBs **MUST** comply with SMIV2 as defined in IETF STD 58.

5.2.1 Provisioning and network management service provider

A single physical device (e.g., embedded-MTA) will be completely provisioned and managed by a single business entity. In the case of multiple service providers offering different services on the same device (e.g., data by one provider, voice by another provider), a secondary service provider will act as the "contractor" for the primary provider in the areas of device provisioning and management. See Figure 2.

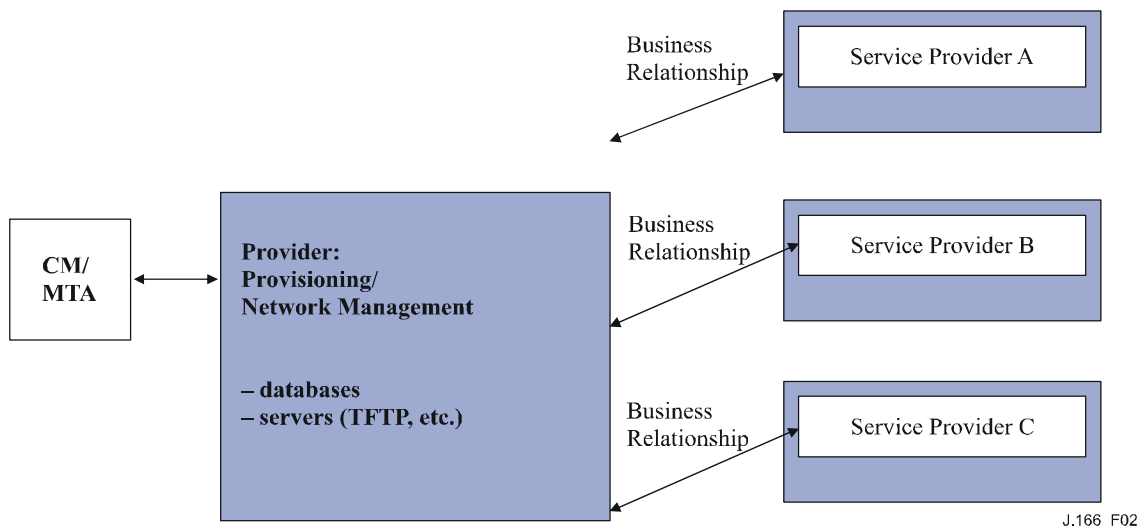


Figure 2/J.166 – Partitioning of management domains

5.2.2 Support for embedded and stand-alone MTAs

The IPCablecom MIBs will provide features for both embedded and stand-alone MTAs. Since Stand-alone MTAs are not required to include any CM related functions, the IPCablecom MIBs, therefore, should be independent of CM and able to provide management support for voice communications functionalities. Cable Modems with Embedded MTAs must adhere to the DOCSIS or eDOCSIS specifications related to the MIBs. The CM part of the-E-MTA **MUST** support eDOCSIS requirements defined in ITU-T Rec. J.126.

Figure 3 describes the possible MIB module implementation for an MTA (Embedded or Stand-alone):

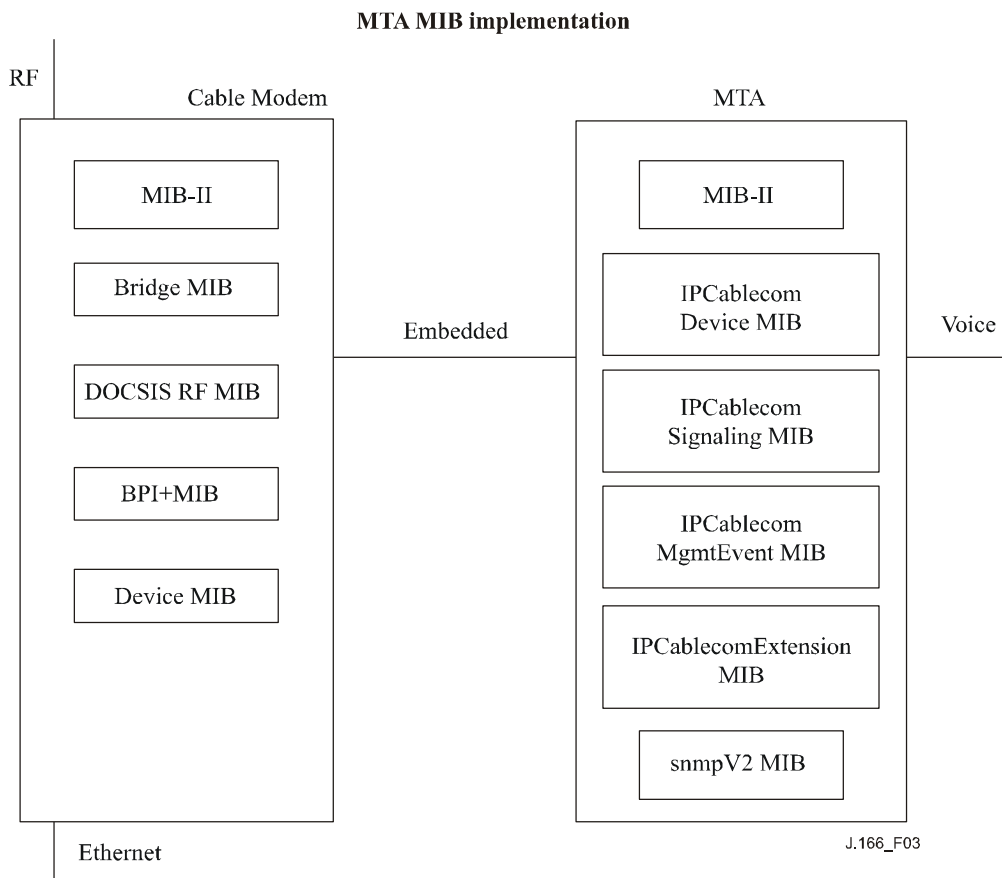


Figure 3/J.166 – Embedded and Stand-alone MTA implementations

5.2.3 Simple network management protocol (SNMP) considerations

SNMPv3 provides an extended User Security Model which implies changes to the way SNMP packets are exchanged between agents and managers. Since MIB modules are used to define the content of the packets, the changes for SNMPv3 do not affect MIB design.

The only requirements imposed are that IPCablecom MIBs MUST conform to SMiv2, which is described in RFCs 2578 and 2579.

The following RFCs provide more information on SNMPv3:

- IETF RFC 3410, Introduction and Applicability Statements for Internet Standard Management Framework;
- IETF RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks;
- IETF RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP);
- IETF RFC 3413, Simple Network Management Protocol (SNMP) Applications;
- IETF RFC 3414, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3);
- IETF RFC 3415, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP).

5.2.3.1 USM requirements

For IPCablecom, the usmUserTable MUST be configured immediately after the AP Reply received from the Provisioning Server with the following entries.

```
usmUserEngineID - the SNMP local engine id
usmUserName - MTA-Prov-xx:xx:xx:xx:xx:xx
usmUserSecurityName - MTA-Prov-xx:xx:xx:xx:xx:xx
usmUserCloneFrom - 0.0
usmUserAuthProtocol - usmHMACMD5AuthProtocol or
                      usmHMACSHAAuthProtocol
usmUserAuthKeyChange - ""
usmUserOwnAuthKeyChange - ""
usmUserPrivProtocol - usmDESPrivProtocol if privacy indicated in AP Reply,
usmNoPrivProtocol if no privacy is indicated in the AP Reply.
UsmUserPrivKeyChange - ""
UsmUserOwnPrivKeyChange - ""
usmUserPublic ""
usmUserStorageType - permanent
usmUserStatus - active
```

The xx:xx:xx:xx:xx:xx in the usmUserName and usmUserSecurityName represents the MAC address of the MTA.

Initial authentication and privacy keys for this user are derived from the AP Reply message.

New users MAY be created by cloning as defined in SNMPv3. This MAY be done through the config file, or later through SNMP Set operations.

5.2.3.2 VACM requirements

The following VACM entries MUST be defined for IPCablecom. Other table entries MAY be implemented at vendor or operator discretion.

VACM views MUST be defined for IPCablecom as described below.

5.2.3.2.1 vacmSecurityToGroup Table

The following configuration of the vacmSecurityToGroup table provides a read/write/create view.

```
vacmSecurityModel - USM
vacmSecurityName - "MTA-Prov-xx:xx:xx:xx:xx:xx"
vacmGroupName - 'PacketCableFullAccess'
vacmSecurityToGroupStorageType - permanent
vacmSecurityToGroupStatus - active
```

5.2.3.2.2 vacmAccessTable

The vacmAccessTable MUST be configured with the following entries. Other table entries MAY be implemented at vendor or operator discretion.

5.2.3.2.2.1 Full access

This configuration allows for read access of all MIB modules in the MTA, write access to IPCablecom MIB modules, and notifications as defined in the IPCablecom MIB modules:

```
vacmGroupName - PacketCableFullAccess
vacmAccessContextPrefix - ""
vacmAccessSecurityModel - USM
vacmAccessSecurityLevel - authPriv or authNoPriv, depending on whether privacy
has been specified
vacmAccessContextMatch - exact
vacmAccessReadViewName - ReadOnlyView
vacmAccessWriteViewName - FullAccessView
```

```
vacmAccess NotifyViewName - NotifyView
vacmAccessStorageType - permanent
vacmAccessStatus - active
```

5.2.3.2.3 MIB View requirements

The FullAccessView MUST consist of the MIB2 system group, the IFMIB, and all IPCablecom defined MIB modules. It MAY include vendor defined MIBs, VACM, USM, and Notifications MIB. The following lists the required OIDs:

```
1.3.6.1.2.1.1          /* MIB-II system group MIB tree */
1.3.6.1.2.1.2.2       /* MIB-II IF MIB tree */
1.3.6.1.4.1.4491.2.2  /* PacketCable Project MIB tree */
1.3.6.1.6.3.13        /* NOTIFY MIB tree */
1.3.6.1.6.3.15        /* USM MIB tree */
1.3.6.1.6.3.16        /* VACM MIB tree */
```

The ReadOnlyView MUST consist of the entire MIB tree contained in the MTA, including IPCablecom defined MIB modules, and vendor defined MIB modules for IPCablecom.

```
1.3.6.1                /* Full Internet MIB Tree*/
```

The NotifyView MUST consist of the MTA MIB tree, MIB-2 System MIB tree and the snmpTrapOID MIB. It MAY include vendor defined MIB modules.

```
1.3.6.1.4.1.4491.2.2.1 /*MTA mib tree*/
1.3.6.1.2.1.1          /* MIB-2 system mib tree */
1.3.6.1.6.3.1.1.4.1.0 /* snmpTrapOID mib*/
```

5.3 Functional requirements

This clause describes management functions that are supported by the IPCablecom MIB modules.

5.3.1 IPCablecom device provisioning

The IPCablecom MIB modules should provide definitions for attributes that are required in the MTA device-provisioning flows. These attributes are specified in the J.167 MTA device provisioning specification and include parameters such as CMS identifier, MTA domain name, MTA server addresses, and MTA capabilities. These attributes are defined as configuration file attributes and/or MIB objects as needed.

5.3.2 Security

The IPCablecom MIB modules provide definitions for attributes that are required for security handshake of the MTA and the provisioning server. These attributes are contained in ITU-T Rec. J.170 and include certificates and signatures.

5.3.3 Voice interfaces (FFS)

The IPCablecom MIB modules should provide a generic external interface to voice service management attributes. This should be done so as to allow a device to implement proprietary mechanisms for internal control and management of voice interfaces.

5.3.4 Packet voice call signalling

The IPCablecom MIB modules should provide attributes that are needed for management of the packet voice call signalling protocol (ITU-T Rec. J.162). Examples of attributes that have to be supported for packet voice call signalling include:

- Dial time-outs;
- Distinctive ring patterns;

- COder-DECoder (Codec) capabilities;
- Signalling configuration for voice communication end points;
- Call agent identifier.

5.3.5 Media Packet Transport

The IPCablecom MIB modules do not provide any managed objects to monitor and manage media packet transport. The RTP and RTCP protocols are used for media transport in IPCablecom. The RTP MIB (IETF RFC 2959) may be used for management of the media transport function of the MTA. Specific details are for future study.

5.3.6 Fault management (FFS)

The IPCablecom MIB modules should provide objects for the management of network faults and failures. Some of these managed objects and management functions are defined in the IPCablecom MTA MIB (J.168) and the IPCablecom Signalling MIB (J.169) ITU-T Recommendations. Further definition of default management is for future study.

5.3.7 Performance management (FFS)

The IPCablecom MIB modules should provide objects for the monitoring of the performance of the network when used for voice communications. Further definition of performance management is for future study.

5.3.8 Event Management

The IPCablecom Management Event MIB module provides the means to define and distribute events generated by the MTA. Refer to the Management Event MIB specification (ITU-T Rec. J.176) for more details.

6 MIBs available in a IPCablecom network

In designing the IPCablecom MIBs, it was necessary to consider other MIBs that are also present in the network and which can provide the required attributes and functions. This clause describes the MIBs that can be present in the IPCablecom MTA device, and which can be used for IPCablecom management functions as needed.

Table 1 lists the MIB modules that must be present in the IPCablecom MTAs. E-MTAs and S-MTAs MUST implement MIB modules present in Table 1.

Table 1/J.166 – MIB modules implemented by E-MTA and S-MTA

IF MIB
MIB II
Ethernet MIB
Bridge MIB
IPCablecom MTA Device MIB
IPCablecom Signalling MIB
IPCablecom Management Event MIB
snmpV2 MIB group

As mentioned before, partitioning of voice and data services and support of both S-MTA and E-MTA has been part of the requirements for design of the IPCablecom MIBs Framework.

Figure 3 in the General Requirements clause describes possible organizations of the MIB modules in order to meet these requirements.

6.1 DOCSIS MIB modules

As described in 5.2, the IPCablecom Embedded MTA must support the DOCSIS (ITU-T Rec. J.112 or J.122) MIB requirements. Refer to the following documents for the normative DOCSIS MIB requirements:

- For J.112, the MIB module requirements are defined in section 3 of ANSI/SCTE 23-3.
- For J.122, the MIB module requirements are defined in section 6 of ANSI/SCTE 79-2.

6.2 IF MIB

This is the interfaces section of the MIB II (RFC 2863), and is needed for definitions of multiple interfaces in the MTA.

6.3 MIB II

Request for Comments (RFC) 1907, RFC 2011, and RFC 2013 define the second version of the Management Information Base (MIB-II) for use with network management protocols in TCP/IP-based Internets. Not all objects in this MIB are deemed necessary for the IPCablecom MTA device. This MIB module only requires the **system**, **interfaces**, **IP**, and **transmission** objects of MIB II to be present in the MTA.

The system object group contact, administrative, location, and service information regarding the managed node.

6.3.1 sysDescr Requirements

The MTA's MIB II sysDescr object MUST conform to the format specified in DOCSIS SCTE 23-3.

6.3.2 sysObjectID Requirements

sysObjectID is defined as follows:

```
sysObjectID OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
ACCESS read-only
STATUS mandatory
DESCRIPTION
```

```
"The vendor's authoritative identification of the network
management subsystem contained in the entity. This
value is allocated within the SMI enterprises subtree
(1.3.6.1.4.1) and provides an easy and unambiguous
means for determining 'what kind of box' is being
managed. For example, if vendor 'Flintstones, Inc.'
was assigned the subtree 1.3.6.1.4.1.4242, it could
assign the identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred
Router'."
```

```
::= { system 2 }
```

By using sysObjectID the manager will be able to determine any enterprise specific MIBs which must be used to manage the embedded MTA.

6.3.3 "ifTable" requirements

IPCablecom ifTable MUST contain information about all IPCablecom endpoints. IfIndex, in case of IPCablecom MTAs MUST start with value of 9 for telephony endpoints and MUST be incremented sequentially and match the physical numbering of the telephony endpoints (Indices 2 through 8 are reserved for future use and the usage of index 1 is defined later in this clause). Each instance of the end-point in an E-MTA MUST have a corresponding entry ("conceptual row") in the "ifTable" MIB Table.

The CableModem part of an embedded MTA MUST adhere to DOCSIS and eDOCSIS requirements for MIB compliancy.

For each "conceptual row" in the "ifTable" table that corresponds to a Telephony Endpoint, the following conceptual columns MUST be used:

- "ifIndex"
- "ifDescr"
- "ifType"
- "ifAdminStatus"
- "ifOperStatus"

Each conceptual row in "ifTable" MUST conform to the "IANAifType-MIB" definition for the IPCablecom interface type:

- "ifType" - voiceOverCable (198)
- "ifDescr" - "Voice Over Cable Interface"

IfIndex 1 is used to recognize the DOCSIS Cable Modem behind which an MTA is connected and the MIB modules involved are indicated in Tables 2 and 3. In the case of an embedded MTA, the tables MUST be adhered to. For stand-alone MTAs, the MTA MAY choose to follow the same. In case a stand-alone MTA cannot display the information, ifIndex 1 MUST be left unused. If the stand-alone MTA is behind a CableHome or other device for its data connectivity, it MAY change the ifDescr to reflect the same.

IPCablecom MTAs MUST implement SCTE 23-3, IETF STD 5 and RFC 2011. IPCablecom implementation MUST conform to the ifTable and ipNetToMediaTable defined below:

Table 2/J.166 – RFC 2863 ifTable, MIB-Object Details for embedded MTA Device Interfaces

RFC 2863 MIB-Object details for MTA Device Interface	MTA Device
IfIndex	1
ifDescr: MUST match the text provided in the next column.	"DOCSIS Embedded Interface"
IfType	other(1)
IfMtu	0
IfSpeed	0
ifPhysAddress	eMTA MAC address
IfAdminStatus: Only up control is required for this interface, down(2) and testing(3) is out of the scope of this specification.	up(1)
ifOperStatus: only up report is required for this object, other options are out of the scope of this specification.	up(1)
IfLastChange	per RFC 2863
ifInOctets: This object is optional, if not implemented, it MUST return 0	(n), 0
ifInNUCastPkts	Deprecated
IfInDiscards	0
IfInErrors	0
IfUnknownProtos	0
ifOutOctets: This object is optional, if not implement MUST return 0	(n), 0
ifOutUCastPkts: This object is optional, if not implemented, it MUST return 0	(n), 0
IfOutNUCastPkts	Deprecated
IfOutDiscards	0
IfOutErrors	0
IfOutQlen	Deprecated
IfSpecific	Deprecated
ifXTable: entries in ifXtable for this type of interface are not required	NA

Table 3/J.166 – RFC 2011 ipNetToMedia MIB-Object Details for eMTA Device Interfaces

RFC 2011 MIB-Object details for MTA Device Interfaces	CM device
IpNetToMediaIfIndex	1
IpNetToMediaPhysAddress	CM MAC Address
IpNetToMediaNetAddress	Acquired CM IP address
IpNetToMediaType	Static(4)
IfIndex	1

6.4 Ethernet MIB

The Ethernet MIB specifies the definitions of managed objects for the Ethernet-like interfaces (RFC 3665).

6.5 Annex C

Annex C contains Network Call Signalling information for provisioning. The data is derived from the IPCablecom NCS Recommendation (ITU-T Rec. J.162).

The IPCablecom Signalling MIB module contains general configuration information that applies to the Network Call Signalling (NCS) protocol on a per MTA device basis. This data only provides the means to provision call signalling parameters on a device basis.

The IPCablecom Signalling MIB module also defines managed objects applicable on a per endpoint basis. The NCS endpoint table (pkcSigEndPntConfigTable) contains specific NCS endpoint configuration information. This data only provides the means to provision network call signalling per endpoint.

6.5.1 Annex C general configuration information

Annex C contains general configuration information that applies to network call signalling on a device basis. This information is also contained in ITU-T Rec. J.162.

This data only provides the means to provision network call signalling on a device basis.

6.5.2 Annex C per endpoint data

Annex C contains a per endpoint table. This table contains general configuration information that applies to network call signalling on a per endpoint basis. This information is also found in the configuration file defined in ITU-T Rec. J.162. This data only provides the means to provision network call signalling per endpoint.

6.6 Annex B

Annex B contains data for provisioning the MTA device and supporting the provisioned functions, specifically Syslog. The data is derived from the IPCablecom ITU-T Rec. J.167, and the CM Device MIB.

Annex B contains general configuration information to provision the MTA on a device basis. These objects support provisioning required servers, and security information.

6.7 Event Management MIB

The IPCablecom Management Event MIB module is defined in Annex D. It provides a common data and format definition for events (informative, alarm, etc.). It also specifies by what means events are transmitted. Use of a common event mechanism facilitates management of the MTA in a multi-vendor environment and provides a standard means to implement IPCablecom specified events.

7 IPCablecom MIB module implementation

This clause describes a reference implementation of the MIBs in an IPCablecom device. Only E-MTA type implementations are considered here.

7.1 MTA components

Figure 4 shows the components of a typical MTA.

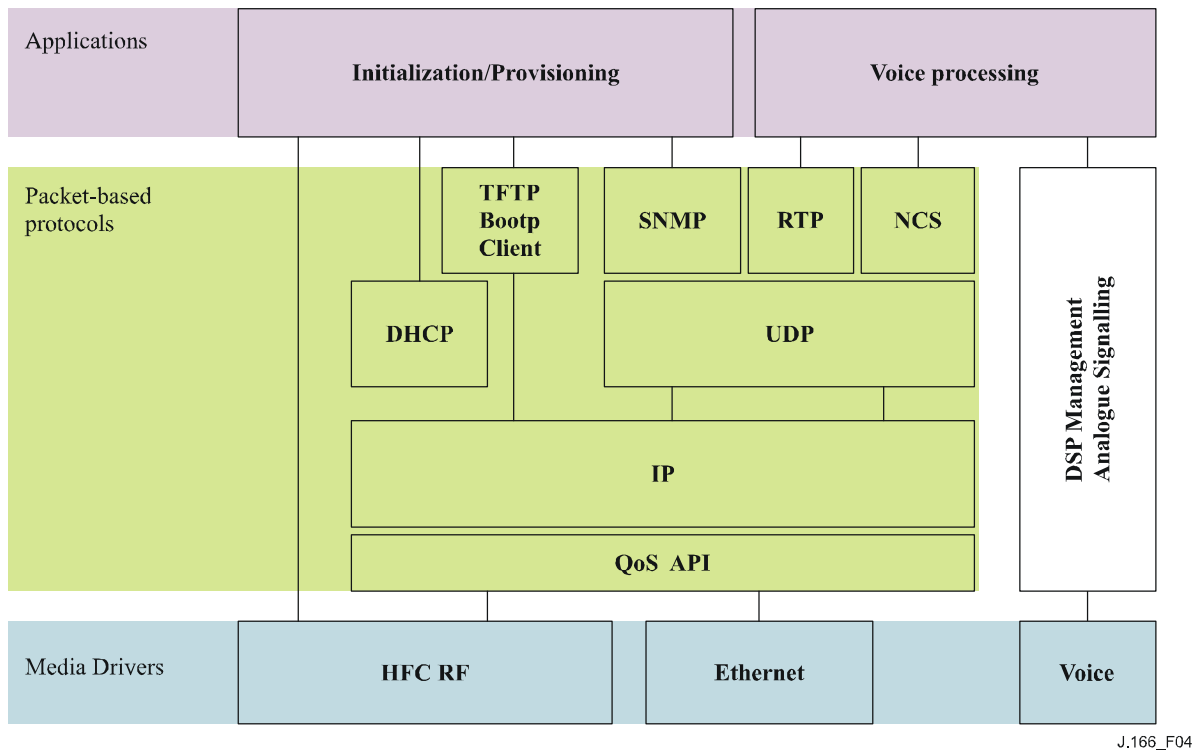


Figure 4/J.166 – MTA components

As shown here, the MTA components can be organized into separate areas, i.e., packet-based protocols, which run on top of IP and the voice subsystem which consists of DSP engines and their associated software. MIBs that are implemented in the MTA have to be organized so as to facilitate this separation. IPCablecom MIB specifies functions for the packet-based protocol section of the MTA. No analogue voice MIBs are specified for the MTA.

NOTE – Please refer to the IPCablecom Security Specification (ITU-T Rec. J.170) for the security protocols.

7.2 MIB layering

Figure 5 describes the MIB layering model. The two stacks represent the packet network and analogue voice sections of the MTA. On the packet network side MIB layering follows the same layering model as the protocol stacks.

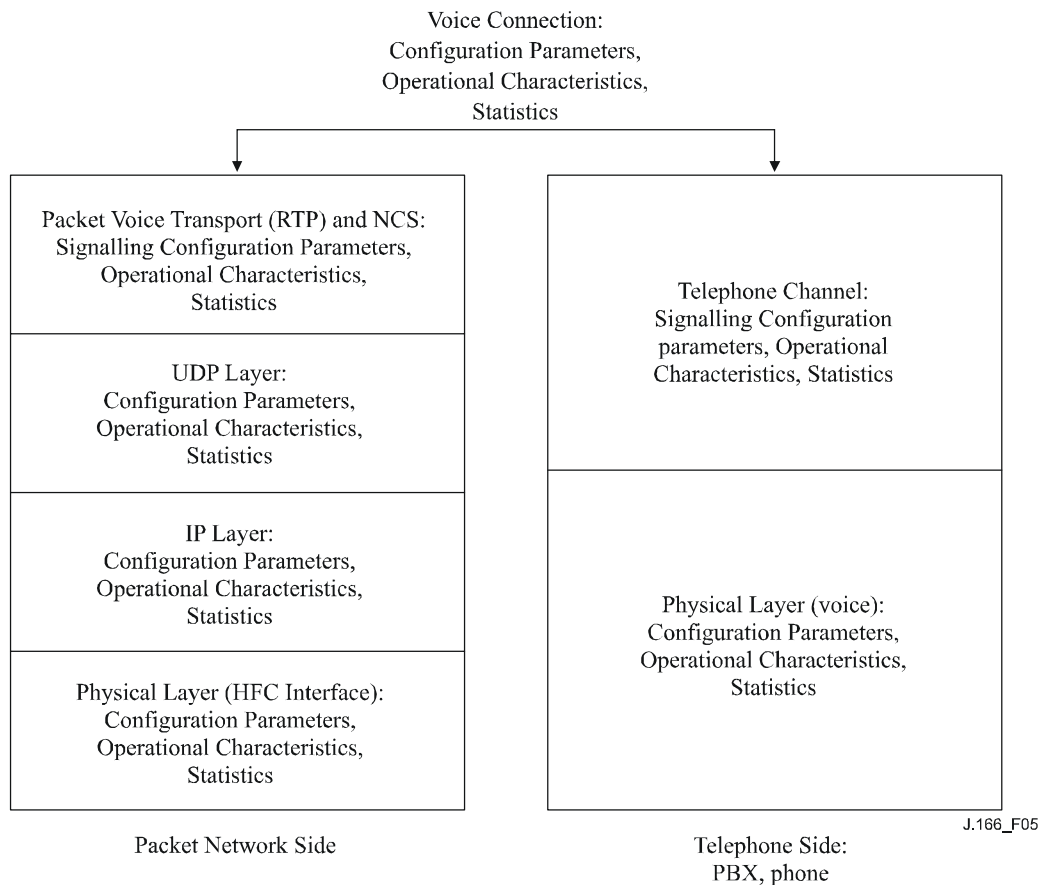


Figure 5/J.166 – MIB layering model

In the context of voice communications, MIBs can be layered into the physical layer attributes which deal with the voice interface and the telephone channel attributes which deal with voice signalling. MIBs for the telephone side of the MTA are for further study.

8 Organization of the MIBs

The IPCablecom MIBs have been gathered together as annexes to this Recommendation so as to keep them all together in one place.

MIBs represent lines of computer code that must be implemented as written. As such the MIBs MUST NOT be translated into other languages.

8.1 Definition MIB

The definition MIB, contained in Annex A, is referenced by several projects including DOCSIS, IPCable2Home, and IPCablecom. It defines the OID registry from which each project assigns its own MIB information modules.

8.2 MTA MIB

The MTA MIB is contained in Annex B. It supersedes ITU-T Rec. J.168. This MIB has been submitted to the IETF for standardization. Once RFC status is achieved, this annex will be deleted and the MIB incorporated by reference.

8.3 Signalling MIB

The signalling MIB is contained in Annex C. It supersedes ITU-T Rec. J.169. This MIB has been submitted to the IETF for standardization. Once RFC status is achieved, this annex will be deleted and the MIB incorporated by reference.

8.4 MEM MIB

The Management Event Mechanism MIB is contained in Annex D. It supersedes ITU-T Rec. J.176.

8.5 MTA Extension MIB

The Media Terminal Adapter Extension MIB is contained in Annex E.

8.6 Signalling Extension MIB

The Signalling Extension MIB is contained in Annex F.

8.7 Battery Backup MIB

The Battery Backup MIB is contained in a separate Recommendation to be approved later. It is a separate document because it is applicable to other cable technologies besides IPCablecom.

Annex A

MIB import data

The Definition MIB MUST be implemented as defined below.

```
CLAB-DEF-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    enterprises
        FROM SNMPv2-SMI
    DocsX509ASN1DEREncodedCertificate
        FROM DOCS-IETF-BPI2-MIB;

cableLabs MODULE-IDENTITY
    LAST-UPDATED "200504081700Z" -- April 8, 2005
    ORGANIZATION "Cable Television Laboratories, Inc."
    CONTACT-INFO
        "Editor: Jean-Francois Mule
        Postal: Cable Television Laboratories, Inc.
        858 Coal Creek Circle
        Louisville, Colorado 80027-9750
        U.S.A.
        Phone: +1 303-661-9100
        Fax: +1 303-661-9199
        E-mail: jfm@cablelabs.com
        mibs@cablelabs.com"

DESCRIPTION
    "This MIB module defines the namespace organization for the
    CableLabs enterprise OID registry.

    Copyright 1999-2005 Cable Television Laboratories, Inc.
    All rights reserved."
```

REVISION "200504081700Z" -- April 8, 2005

DESCRIPTION

"This revision, published as CL-SP-MIB-CLABDEF-I05."

::= { enterprises 4491 }

-- Sub-tree for Registrations

clabFunction OBJECT IDENTIFIER ::= { cableLabs 1 }
clabFuncMib2 OBJECT IDENTIFIER ::= { clabFunction 1 }
clabFuncProprietary OBJECT IDENTIFIER ::= { clabFunction 2 }

-- Sub-tree for Project Definitions

clabProject OBJECT IDENTIFIER ::= { cableLabs 2 }
clabProjDocsis OBJECT IDENTIFIER ::= { clabProject 1 }
clabProjPacketCable OBJECT IDENTIFIER ::= { clabProject 2 }
clabProjOpenCable OBJECT IDENTIFIER ::= { clabProject 3 }
clabProjCableHome OBJECT IDENTIFIER ::= { clabProject 4 }

-- Sub-tree for Global Security Definitions

clabSecurity OBJECT IDENTIFIER ::= { cableLabs 3 }
clabSecCertObject OBJECT IDENTIFIER ::= { clabSecurity 1 }

-- Sub-tree for CableLabs cross project common MIB definitions

clabCommonMibs OBJECT IDENTIFIER ::= { cableLabs 4 }

--

-- CableLabs DOCSIS Project Sub-tree Definitions

--

dsgMIB OBJECT IDENTIFIER

-- DOCSIS Set-top Gateway (DSG) MIB module
-- This object identifier points to the MIB module
-- DOCSIS-SETTOP-GATEWAY-MIB, which is being deprecated by
-- DSG-IF-MIB MIB module (dsgIfMib).
-- Reference:
-- CableLabs DOCSIS Set-top Gateway (DSG) Interface Specification
::= { clabProjDocsis 1 }

docsLoadBalMib OBJECT IDENTIFIER

-- DOCSIS MIB module defining the CMTS configuration parameters to
-- support Load Balancing requirements."
::= { clabProjDocsis 2 }

dsgIfMIB OBJECT IDENTIFIER

-- DOCSIS Set-top Gateway (DSG) MIB module
-- Obsoletes DOCSIS-SETTOP-GATEWAY-MIB Module (dsgMib)
-- defined initially in DOCSIS Set-top Gateway (DSG) Interface
-- Specification SP-DSG-I01-020228.
-- Reference:
-- CableLabs DOCSIS Set-top Gateway (DSG) Interface Specification
::= { clabProjDocsis 3 }

dsgIfStdMib OBJECT IDENTIFIER

-- DOCSIS Set-top Device (DSG) MIB module.
-- Reference:
-- CableLabs DOCSIS Set-top Gateway (DSG) Interface Specification
::= { clabProjDocsis 4 }

docsIfExt2Mib OBJECT IDENTIFIER

-- This MIB module contains the management objects that enhance
-- DOCSIS RFI Interface Extensions. Contains Enhancements to
-- DOCSIS RFI interface MIB module.
-- Reference:
-- CableLabs DOCSIS RFI Interface Specification.
::= { clabProjDocsis 5 }

```

docsTestMIB OBJECT IDENTIFIER
-- DOCSIS Test MIB module supporting programmable test features
-- for DOCSIS 2.0 compliant Cable Modems (CM) and Cable Modems
-- Termination Systems (CMTS).
-- Reference:
-- CableLabs DOCSIS 2.0 Testing MIB Specification
 ::= { clabProjDocsis 12 }

sledMib OBJECT IDENTIFIER
-- eDOCSIS MIB module supporting the Software Loopback Application
-- for eDOCSIS (SLED).
-- Reference:
-- CableLabs eDOCSIS Specification
 ::= { clabProjDocsis 13 }

--
-- CableLabs CableHome Project Sub-tree Definitions
-- Reference
-- CableLabs CableHome Specification
--
cabhPsDevMib OBJECT IDENTIFIER
-- CableHome MIB module defining the basic management objects for
-- the Portal Services logical element of a CableHome compliant
-- Residential Gateway device. The PS device parameters describe
-- general PS Device attributes and behaviour characteristics
 ::= { clabProjCableHome 1 }

cabhSecMib OBJECT IDENTIFIER
-- CableHome MIB module defining the basic management objects for
-- the firewall and other security features of the Portal Services
-- element.
 ::= { clabProjCableHome 2 }

cabhCapMib OBJECT IDENTIFIER
-- CableHome MIB module defining the basic management objects for
-- the CableHome Addressing Portal (CAP) function of the Portal
-- Services element.
 ::= { clabProjCableHome 3 }

cabhCdpMib OBJECT IDENTIFIER
-- This MIB module supplies the basic management objects for the
-- CableHome DHCP Portal (CDP) function of the Portal Services
-- element.
 ::= { clabProjCableHome 4 }

cabhCtpMib OBJECT IDENTIFIER
-- CableHome MIB module supporting the remote LAN diagnostic
-- features provided by the CableHome Test Portal (CTP) function
-- of the Portal Services element.
 ::= { clabProjCableHome 5 }

cabhQosMib OBJECT IDENTIFIER
-- CABLEHOME QOS MIB Module (cabhQosMib).
-- This object identifier points to the MIB module
-- CABH-QOS-MIB, which is being deprecated by
-- CABH-QOS2-MIB module (cabhQos2Mib).
-- Reference:
-- CableLabs CableHome 1.1 Specification
 ::= { clabProjCableHome 6 }

cabhCsaMib OBJECT IDENTIFIER
-- CableHome MIB module defining management objects for the
-- configuration and monitoring of CableHome Commercial Services
-- Annex.

```

```

-- Reference:
-- CableLabs CableOffice Commercial Services Annex MIB
-- Specification
::= { clabProjCableHome 7 }

cabhQos2Mib OBJECT IDENTIFIER
-- Obsoletes CABH-QOS-MIB module (cabhQosMib)
-- defined initially in CABLEHOME 1.1 Interface Specification.
-- This MIB module defines the Quality of Service Management
-- Information Base (MIUB) for CableHome UPnP QOS-compliant
-- devices.
-- Reference:
-- CableLabs CableHome 1.1 Specification
::= { clabProjCableHome 8 }

--
-- CableLabs PacketCable Project Sub-tree Definitions
--
pktcMtaMib OBJECT IDENTIFIER
-- PacketCable MIB module defining the basic management object for
-- the Multimedia Terminal Adapter (MTA) devices compliant with
-- PacketCable requirements.
-- Reference
-- CableLabs PacketCable MTA Device Provisioning Specification
::= { clabProjPacketCable 1 }

pktcSigMib OBJECT IDENTIFIER
-- PacketCable MIB module defining the basic management object for
-- the PacketCable MTA Signalling protocols. This version of the MIB
-- includes common signalling and Network Call Signalling (NCS)
-- related signalling objects.
-- Reference
-- CableLabs PacketCable MTA Device Provisioning Specification
::= { clabProjPacketCable 2 }

pktcEventMib OBJECT IDENTIFIER
-- PacketCable MIB module defining the basic management objects for
-- event reporting.
-- Reference
-- CableLabs PacketCable Management Event Specification
::= { clabProjPacketCable 3 }

pktcSecurity OBJECT IDENTIFIER
-- CableLabs OID reserved for security and used to specify errors
-- that can be returned for the Kerberos KDC - Provisioning
-- Server interface, or the MTA-CMS Kerberized IPsec interface, or
-- the MTA-Provisioning Server Kerberized SNMPv3 interface.
-- CableLabs PacketCable Security Specification
::= { clabProjPacketCable 4 }

pktcLawfulIntercept OBJECT IDENTIFIER
-- CableLabs OID reserved for the PacketCable Electronic
-- Surveillance Protocol (PCESP) between the Delivery Function
-- and Collection Function. This OID is used to define the ASN.1
-- PCESP messages.
-- CableLabs PacketCable Electronic Surveillance Protocol
-- Specification
::= { clabProjPacketCable 5 }

--
-- Sub-tree for PacketCable MIB Enhancements
--

```



```

pktcEnhancements OBJECT IDENTIFIER ::= { clabProjPacketCable 6 }

-- The following MIB OBJECTS are being introduced for
-- incorporation of new MIB objects (MIB enhancements)
-- proposed to the PacketCable MIB group.
-- This includes new MIB objects being introduced
-- as part of the PacketCable MIB Enhancement efforts
-- and as a place holder for future revisions.
-- This sub-division would facilitate easier incorporation
-- of proposed IETF Drafts/RFCs by keeping enhancements
-- independent of RFC/Draft changes.
-- For new MIB tables that use previously used indices, it is
-- recommended that the AUGMENT CLAUSE be used to aid SNMP Operations,
-- as deemed necessary.

pktcEnMtaMib OBJECT IDENTIFIER
  -- PacketCable MIB module enhancements to the basic management
  -- objects defined by the MIB group pktcMtaMib for the Multimedia
  -- Terminal Adapter (MTA) devices compliant with PacketCable
  -- requirements.
  -- Reference:
  -- CableLabs PacketCable MTA Device Provisioning Specification.
  ::= { pktcEnhancements 1 }

pktcEnSigMib OBJECT IDENTIFIER
  -- PacketCable MIB module enhancements to the basic management
  -- objects defined by the MIB group pktcSigMib for the
  -- PacketCable MTA Signalling protocols.
  -- Reference:
  -- CableLabs PacketCable MTA Device Provisioning Specification.
  ::= { pktcEnhancements 2 }

pktcEnEventMib OBJECT IDENTIFIER
  -- PacketCable MIB module enhancements to the basic management
  -- objects defined by the MIB group pktcEventMib for event reporting.
  -- Reference:
  -- CableLabs PacketCable Management Event Specification.
  ::= { pktcEnhancements 3 }

pktcEnSecurityMib OBJECT IDENTIFIER
  -- PacketCable MIB module enhancements to the basic management
  -- objects defined by the reserved MIB group pktcSecurity.
  -- Reference:
  -- CableLabs PacketCable Security Specification.
  ::= { pktcEnhancements 4 }

--
--
-- Definition of CableLabs Security Certificate Objects
--
clabSrvCPrvdrRootCACert OBJECT-TYPE
  SYNTAX      DocsX509ASN1DEREncodedCertificate
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The X509 DER-encoded CableLabs Service Provider Root CA
    Certificate."
  REFERENCE
    "CableLabs CableHome Specification;
    CableLabs PacketCable Security Specification."
  ::= { clabSecCertObject 1 }

```

```

clabCVCRoootCACert OBJECT-TYPE
    SYNTAX      DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The X509 DER-encoded CableLabs CVC Root CA Certificate."
    REFERENCE
        "CableLabs CableHome Specification;
        CableLabs PacketCable Security Specification."
    ::= { clabSecCertObject 2 }

clabCVCCACert OBJECT-TYPE
    SYNTAX      DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The X509 DER-encoded CableLabs CVC CA Certificate."
    REFERENCE
        "CableLabs CableHome Specification;
        CableLabs PacketCable Security Specification."
    ::= { clabSecCertObject 3 }

clabMfgCVCCert OBJECT-TYPE
    SYNTAX      DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The X509 DER-encoded Manufacturer CVC Certificate."
    REFERENCE
        "CableLabs CableHome Specification;
        CableLabs PacketCable Security Specification."
    ::= { clabSecCertObject 4 }

clabMfgCACert OBJECT-TYPE
    SYNTAX      DocsX509ASN1DEREncodedCertificate
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The X509 DER-encoded Manufacturer CA Certificate."
    REFERENCE
        "CableLabs CableHome Specification;
        CableLabs PacketCable Security Specification."
    ::= { clabSecCertObject 5 }

--
-- CableLabs cross project common MIB sub-tree definitions
--

clabUpsMib OBJECT IDENTIFIER
    -- CableLabs cross project MIB module defining the basic management
    -- objects for the configuration and monitoring of the battery
    -- backup and UPS functionality for CableLabs compliant devices.
    ::= { clabCommonMibs 1 }

END

```

Annex B

MTA MIB

The MTA MIB MUST be implemented as defined below.

```
PKTC-MTA-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        MODULE-IDENTITY,
        OBJECT-TYPE,
        Integer32, Counter32,
        BITS,IpAddress, NOTIFICATION-TYPE          FROM SNMPv2-SMI
        TruthValue, RowStatus, DisplayString,
        MacAddress, TEXTUAL-CONVENTION             FROM SNMPv2-TC
        OBJECT-GROUP, MODULE-COMPLIANCE,
        NOTIFICATION-GROUP                        FROM SNMPv2-CONF
        clabProjPacketCable                      FROM CLAB-DEF-MIB
        ifIndex                                   FROM IF-MIB
        SnmpAdminString                          FROM SNMP-FRAMEWORK-MIB
        sysDescr                                  FROM SNMPv2-MIB;

pktcMtaMib MODULE-IDENTITY
    LAST-UPDATED      "200501280000Z" -- January 28, 2005
    ORGANIZATION      "Packet Cable OSS Group"
    CONTACT-INFO
        "Sumanth Channabasappa
        Postal: Cable Television Laboratories, Inc.
        858 Coal Creek Circle
        Louisville, Colorado 80027-9750
        U.S.A.
        Phone:   +1 303-661-9100
        Fax:     +1 303-661-9199
        E-mail:  mibs@cablelabs.com"

    DESCRIPTION
        "This MIB module supplies the basic management objects
        for the MTA Device
        Acknowledgements:
        Angela Lyda           -   Arris Interactive
        Chris Melle           -   AT&T Broadband Labs
        Sasha Medvinsky       -   Motorola
        Roy Spitzer           -   Telogy Networks, Inc.
        Rick Vetter           -   Motorola
        Eugene Nechamkin      -   BroadCom Corp.
        Satish Kumar          -   Texas Instruments
        Copyright 1999-2005 Cable Television Laboratories, Inc.
        All rights reserved."
    REVISION "200501280000Z "
    DESCRIPTION
        "This revision, published as part of the PacketCable 1.5
        MIB MTA Specification I01."
    ::= { clabProjPacketCable 1 }

-- Textual conventions
X509Certificate ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "An X509 digital certificate encoded as an ASN.1 DER object."
    SYNTAX OCTET STRING (SIZE (0..4096))
--
-- PacketCable 1.5 only supports Embedded MTAs
--
```

```

-----
--
-- The MTA MIB only supports a single provisioning server.
--
-----

pktcMtaMibObjects      OBJECT IDENTIFIER ::= { pktcMtaMib 1 }
pktcMtaDevBase         OBJECT IDENTIFIER ::= { pktcMtaMibObjects 1 }
pktcMtaDevServer       OBJECT IDENTIFIER ::= { pktcMtaMibObjects 2 }
pktcMtaDevSecurity     OBJECT IDENTIFIER ::= { pktcMtaMibObjects 3 }

--
-- The following group describes the base objects in the MTA
--
pktcMtaDevResetNow    OBJECT-TYPE
    SYNTAX              TruthValue
    MAX-ACCESS          read-write
    STATUS              current
    DESCRIPTION
        "Setting this object to true(1) causes the device to reset.
        Reading this object always returns false(2).  When
        pktcMtaDevResetNow is set to true, the following actions
        occur:
        1. All connections (if present) are flushed locally
        2. All current actions such as ringing immediately
           terminate
        3. Requests for notifications such as notification based
           on digit map recognition are flushed
        4. All endpoints are disabled.
        5. The provisioning flow is started at step MTA - 1."
    ::= { pktcMtaDevBase 1 }

pktcMtaDevSerialNumber OBJECT-TYPE
    SYNTAX              SnmpAdminString(SIZE (0..128))
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
        "This object specifies the manufacturer's serial number
        for this MTA. The value of this object MUST be identical
        to the value specified in DHCP option 43 sub-option 4. "
    REFERENCE
        "PacketCable MTA Device Provisioning Specification;
        RFC 2132, DHCP Options and BOOTP Vendor Extensions"
    ::= { pktcMtaDevBase 2 }

pktcMtaDevHardwareVersion OBJECT-TYPE
    SYNTAX              SnmpAdminString(SIZE (0..48))
    MAX-ACCESS          read-only
    STATUS              obsolete
    DESCRIPTION
        "The manufacturer's hardware version for this MTA."
    ::= { pktcMtaDevBase 3 }

pktcMtaDevMacAddress  OBJECT-TYPE
    SYNTAX              MacAddress
    MAX-ACCESS          read-only
    STATUS              current
    DESCRIPTION
        "This object specifies the telephony MAC address for
        this device. The value of this object MUST be identical
        to the value specified in DHCP option 43 sub-option 11. "
    REFERENCE
        "PacketCable MTA Device Provisioning Specification;
        RFC 2132, DHCP Options and BOOTP Vendor Extensions"

```

```

 ::= { pktcMtaDevBase 4 }
pktcMtaDevFQDN          OBJECT-TYPE
    SYNTAX               SnmpAdminString
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "The Fully Qualified Domain Name for this MTA."
 ::= { pktcMtaDevBase 5 }

pktcMtaDevEndPntCount   OBJECT-TYPE
    SYNTAX                Integer32 (1..255)
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "The physical end points for this MTA."
 ::= { pktcMtaDevBase 6 }

pktcMtaDevEnabled       OBJECT-TYPE
    SYNTAX                TruthValue
    MAX-ACCESS            read-write
    STATUS                current
    DESCRIPTION
        "This object contains the MTA Admin Status of this device.
        If this object is set to 'true', the MTA is
        administratively enabled and the MTA MUST be able to
        interact with PacketCable entities such as CMS,
        Provisioning Server, KDC, other MTAs and MGs on all
        PacketCable interfaces.
        If this object is set to 'false', the MTA is
        administratively disabled and the MTA MUST perform the
        following actions for all endpoints:
        - Shutdown all media sessions if present,
        - Shutdown NCS signalling by following the Restart in
          Progress procedures in the PacketCable NCS
          specification.
        Additionally, the MTA MUST maintain the SNMP Interface for
        management. Also, the MTA MUST NOT continue Kerberized Key
        Management with CMSes until this object is set to 'true'.
        Note: MTAs MUST renew the CMS kerberos tickets according
        to the PacketCable Security Specification"
    REFERENCE
        "PacketCable Security Specification;
        PacketCable MTA Device Provisioning Specification"
 ::= { pktcMtaDevBase 7 }

pktcMtaDevTypeIdentifier OBJECT-TYPE
    SYNTAX                SnmpAdminString
    MAX-ACCESS            read-only
    STATUS                current
    DESCRIPTION
        "This is a copy of the device type identifier used in the
        DHCP option 60 exchanged between the MTA and the DHCP
        server."
 ::= { pktcMtaDevBase 8 }

pktcMtaDevProvisioningState OBJECT-TYPE
    SYNTAX                INTEGER {
        pass                (1),
        inProgress          (2),
        failConfigFileError (3),
        passWithWarnings    (4),
        passWithIncompleteParsing (5),
        failureInternalError (6),
        failOtherReason     (7)
    }

```

```

}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This object indicates the completion state of the
    MTA device provisioning process.

    pass:
    If the configuration file could be parsed successfully
    and the MTA is able to reflect the same in its
    MIB, the MTA MUST return the value 'pass'.

    inProgress:
    If the MTA is in the process of being provisioned,
    the MTA MUST return the value 'inProgress'.

    failConfigFileError:
    If the configuration file was in error due to incorrect
    values in the mandatory parameters, the MTA MUST reject
    the configuration file and the MTA MUST return the value
    'failConfigFileError'.

    passWithWarnings:
    If the configuration file had proper values for all the
    mandatory parameters but has errors in any of the optional
    parameters (this includes any vendor specific OIDs which
    are incorrect or not known to the MTA), the MTA MUST
    return the value 'passWithWarnings'.

    passWithIncompleteParsing:
    If the configuration file is valid, but the MTA cannot
    reflect the same in its configuration (for example, too
    many entries caused memory exhaustion), it must accept
    the CMS configuration entries related and the MTA MUST
    return the value 'passWithIncompleteParsing'.

    failureInternalError:
    If the configuration file cannot be parsed due to an
    internal error, the MTA MUST return the value
    'failureInternalError'.

    failureOtherReason:
    If the MTA cannot accept the configuration file for any
    other reason than the ones stated above, the MTA MUST
    return the value 'failureOtherReason'.

    When a final SNMP INFORM is sent as part of Step 25 of
    the MTA Provisioning process, this parameter is also
    included in the final INFORM message."
REFERENCE
    "PacketCable MTA Device Provisioning Specification"
 ::= { pktcMtaDevBase 9 }

pktcMtaDevHttpAccess OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This indicates whether HTTP file access is supported for
        MTA configuration file transfer."
    ::= { pktcMtaDevBase 10 }

pktcMtaDevProvisioningTimer OBJECT-TYPE
    SYNTAX Integer32 (0..30)

```

```

UNITS          "minutes"
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION   "This object defines the time interval for the
               provisioning flow to complete. The MTA MUST finish
               all provisioning operations starting from the moment when
               an MTA receives its DHCP ACK and ending at the moment when
               the MTA downloads its configuration file (e.g., MTA5 to
               MTA23 for Secure Flow) within the period of time set by
               this object. Failure to comply with this condition
               constitutes the provisioning flow failure. If the
               object is set to 0, the MTA MUST ignore the provisioning
               timer condition."
REFERENCE     "PacketCable MTA Device Provisioning Specification."
DEFVAL { 10 }
 ::= { pktcMtaDevBase 11 }

pktcMtaDevProvisioningCounter OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   "This object is the count of the number of times the
               provisioning cycle has looped through step MTA-1 since
               the last reboot."
 ::= { pktcMtaDevBase 12 }

--

pktcMtaDevErrorOidsTable OBJECT-TYPE
SYNTAX SEQUENCE OF PktcMtaDevErrorOidsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION   "If pktcMtaDevProvisioningState is reported with anything
               other than a pass(1) then this table is populated with the
               necessary information, each pertaining to observations of
               the configuration file. Even if different parameters
               share the same error ( Ex: All Realm Names are invalid ),
               all recognized errors must be reported as different
               instances."
 ::= { pktcMtaDevBase 13 }

pktcMtaDevErrorOidsEntry OBJECT-TYPE
SYNTAX PktcMtaDevErrorOidsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION   "This contains the necessary information an MTA must
               attempt to provide in case the configuration file
               is not parsed and/or accepted in its entirety."
INDEX { pktcMtaDevErrorOidIndex }
 ::= { pktcMtaDevErrorOidsTable 1 }

PktcMtaDevErrorOidsEntry ::= SEQUENCE {
    pktcMtaDevErrorOidIndex      Integer32,
    pktcMtaDevErrorOid          SnmpAdminString,
    pktcMtaDevErrorGiven        SnmpAdminString,
    pktcMtaDevErrorReason       SnmpAdminString
}

```

```

pktcMtaDevErrorOidIndex OBJECT-TYPE
    SYNTAX Integer32(1..1024)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This is the index to pktcMtaDevErrorOidsEntry.
        This is an integer value and will start from the value 1
        and be incremented for each error encountered in the
        configuration file. The indices need not necessarily
        reflect the order of error occurrences in the
        configuration file."
    ::= { pktcMtaDevErrorOidsEntry 1 }

pktcMtaDevErrorOid OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This is the OID associated with the particular error. If
        the error was not due to an identifiable OID, then this
        can be populated with impartial identifiers, in hexadecimal
        or numeric format."
    ::= { pktcMtaDevErrorOidsEntry 2 }

pktcMtaDevErrorGiven OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "If the error was due to the value associated
        with the corresponding pktcMtaDevErrorOid, then this
        contains the value of the OID as interpreted by the MTA in
        the configuration file provided. If the error was not due
        to the value of an OID this must be set to an empty
        string. This is provided to eliminate errors due to
        misrepresentation/misinterpretation of data."
    ::= { pktcMtaDevErrorOidsEntry 3 }

pktcMtaDevErrorReason OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This indicates the reason for the error,
        as per the MTAs interpretation, in human readable form.
        Example include:
        VALUE NOT IN RANGE,
        VALUE DOES NOT MATCH TYPE
        UNSUPPORTED VALUE
        LAST 4 BITS MUST BE SET TO ZERO,
        OUT OF MEMORY, CANNOT STORE etc.
        This MAY also contain vendor specific errors
        for vendor specific OIDs and any proprietary error
        codes/messages which can help diagnose errors
        better, in a manner the vendor deems fit."
    ::= { pktcMtaDevErrorOidsEntry 4 }

pktcMtaDevSwCurrentVers OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current

```


DESCRIPTION

"This object identifies the software version currently operating in the MTA. The MTA MUST return a string descriptive of the current software load. This object should use the syntax defined by the individual vendor to identify the software version. The data presented in this object MUST be identical with the software version information contained in the sysDescr MIB Object of the MTA. The value of this object MUST be identical to the value specified in DHCP option 43 sub-option 6."

REFERENCE

"PacketCable MTA Device Provisioning Specification; RFC 2132, DHCP Options and BOOTP Vendor Extensions" ::= { pktcMtaDevBase 14 }

-- The following group describes server access and parameters used for -- initial provisioning and bootstrapping.

-- *****
-- *****This object is obsolete*****
-- *****

pktcMtaDevServerBootState OBJECT-TYPE

SYNTAX INTEGER { operational (1), disabled (2), waitingForDhcpOffer (3), waitingForDhcpResponse (4), waitingForConfig (5), refusedByCmts (6), other (7), unknown (8) }

MAX-ACCESS read-only
STATUS obsolete

DESCRIPTION

"If operational(1), the device has completed loading and processing of configuration parameters and the CMTS has completed the Registration exchange. If disabled(2) then the device was administratively disabled, possibly by being refused network access in the configuration file. If waitingForDhcpOffer(3) then a DHCP Discover has been transmitted and no offer has yet been received. If waitingForDhcpResponse(4) then a DHCP Request has been transmitted and no response has yet been received. If waitingForConfig(5) then a request to the config parameter server has been made and no response received. If refusedByCmts(6) then the Registration Request/Response exchange with the CMTS failed. "

REFERENCE

"DOCSIS Radio Frequency Interface Specification" ::= { pktcMtaDevServer 1 }

-- *****
-- *****This object is obsolete*****
-- *****

pktcMtaDevServerDhcp OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS read-only
STATUS obsolete

```

DESCRIPTION
    "The IP address of the DHCP server that assigned an IP
    address to this device. Returns 0.0.0.0 if DHCP was not
    used for IP address assignment."
 ::= { pktcMtaDevServer 2 }
--
pktcMtaDevServerDns1 OBJECT-TYPE
    SYNTAX      IPAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The IP address of the primary DNS server to be used by the
        MTA to resolve the FQDNs and IP addresses."
 ::= { pktcMtaDevServer 3 }

pktcMtaDevServerDns2 OBJECT-TYPE
    SYNTAX      IPAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The IP address of the Secondary DNS server to be used by
        the MTA to resolve the FQDNs and IP addresses. Contains
        0.0.0.0 if there is no Secondary DNS server specified
        for the MTA under consideration."
 ::= { pktcMtaDevServer 4 }

pktcMtaDevConfigFile OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object specifies the MTA device configuration file
        information, including the access method, the server
        name and the configuration file name. The value of this
        object is the Uniform Resource Locator (URL) of the
        configuration file for TFTP or HTTP download.
        If this object value is a TFTP URL, it must be formatted
        as defined in RFC 3617.
        If this object value is an HTTP URL, it must be formatted
        as defined in RFC 2616.
        If the MTA SNMP Enrollment mechanism is used, then the MTA
        must download the file provided by the Provisioning Server
        during provisioning via an SNMP SET on this object.
        If the MTA SNMP Enrollment mechanism is not used, this
        object MUST contain the URL value corresponding to the
        'siaddr' and 'file' fields received in the DHCP ACK to
        locate the configuration file:  the 'siaddr' & 'file'
        fields represent the host and file of the TFTP URL.
        In this case, the MTA MUST return an
        'inconsistentValue' error in response to SNMP SET
        operations.  The MTA MUST return a zero-length string if
        the server address (host part of the URL) is unknown."
    REFERENCE
        "RFC 3617, URI Scheme for TFTP; RFC 2616, HTTP 1.1"
 ::= { pktcMtaDevServer 5 }

pktcMtaDevSnmpEntity OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the FQDN of the SNMP entity of the
        Provisioning Server. When the MTA SNMP Enrollment
        Mechanism is used, this object represents the server the

```

MTA communicates with, to receive the configuration file URL from, and, to send the enrollment notification to. The SNMP entity is also the destination entity for all the provisioning notifications. It may be also used for post-provisioning SNMP operations. During the provisioning phase, this SNMP entity FQDN is supplied to the MTA via the DHCP option 122 sub-option 3 as defined in RFC 3495."

REFERENCE

"PacketCable MTA Device Provisioning Specification;
RFC 3495, DHCP Option for CableLabs Client Configuration."
 ::= { pktcMtaDevServer 6 }

pktcMtaDevProvConfigHash OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(16|20))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object contains the hash value of the contents of the config file.
If the authentication algorithm is MD5, the length is 128 bits. If the authentication algorithm is SHA-1, the length is 160 bits. The hash calculation MUST follow the requirements defined in the PacketCable Security specification.
When the MTA SNMP Enrollment mechanism is used, this hash value is calculated and sent to the MTA prior to sending the config file. This object value is then provided by the Provisioning server via an SNMP SET operation.
When the MTA SNMP Enrollment mechanism is not in use, the hash value is provided in the configuration file itself and it is also calculated by the MTA. This object value MUST represent the hash value calculated by the MTA.
When the MTA SNMP Enrollment mechanism is not in use, the MTA must reject all SNMP SET operations on this object and return an 'inconsistentValue' error."

REFERENCE

"PacketCable MTA Device Provisioning Specification;
PacketCable Security Specification."
 ::= { pktcMtaDevServer 7 }

pktcMtaDevProvConfigKey OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0|8))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object contains the key used to encrypt/decrypt the configuration file when secure SNMPv3 provisioning is used.
It is sent to the MTA prior to sending the config file. If the privacy algorithm is null, the length is 0. If the privacy algorithm is DES, the length is 64 bits. This object must not be used in non secure provisioning mode.
In non secure provisioning modes, the MTA MUST return an 'inconsistentValue' in response to SNMP SET operations, and, the MTA MUST return a 'genErr' error in response to SNMP GET operations."

::= { pktcMtaDevServer 8 }

```

pktcMtaDevProvSolicitedKeyTimeout OBJECT-TYPE
    SYNTAX      Integer32 (15..600)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object defines a Kerberos Key Management timer on the
        MTA. It is the time period during which the MTA saves the
        nonce and Server Kerberos Principal Identifier to match an
        AP Request and its associated AP Reply response from the
        Provisioning Server.
        After the timeout has been exceeded, the client discards
        this (nonce, Server Kerberos Principal Identifier) pair,
        after which it will no longer accept a matching AP Reply.
        This timer only applies when the Provisioning Server
        initiated key management for SNMPv3 (with a
        Wake Up message). This object should not be used in non
        secure provisioning modes. In non secure provisioning
        modes, the MTA MUST return an 'inconsistentValue' in
        response to SNMP SET operations, and the MTA MUST
        return a 'genErr' error in response to SNMP GET
        operations."
    DEFVAL { 120 }
    ::= { pktcMtaDevServer 9 }

```

```

-----
--
-- Unsolicited Key Updates are based on an exponential backoff
-- mechanism with two timers for AS replies. The fast timers have a
-- maximum timer (pktcMtaDevProvUnsolicitedKeyMaxTimeout seconds) and
-- a nominal timer (pktcMtaDevProvUnsolicitedKeyNomTimeout seconds)
-- from which the backoff timer determinations are made.
--
-----
--
-- Timeouts for unsolicited key management updates are only pertinent
-- before the first SNMPv3 message is sent between the MTA and the
-- Provisioning server and before the configuration file is loaded.
--
-----

```

```

pktcMtaDevProvUnsolicitedKeyMaxTimeout OBJECT-TYPE

    SYNTAX      Integer32 (15..600)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object defines the timeout value that applies to
        an MTA-initiated AP-REQ/REP key management exchange with
        the Provisioning Server in SNMPv3 provisioning.
        It is the maximum timeout value and it may not be exceeded
        in the exponential back-off algorithm. If the DHCP option
        code 122 sub-option 5 is provided to the MTA, it overwrites
        this value.
        In non secure provisioning mode, the MTA MUST return
        a 'genErr' error in response to SNMP GET operations."

```

REFERENCE
"PacketCable Security Specification"
DEFVAL {600}
::= { pktcMtaDevServer 10 }

pktcMtaDevProvUnsolicitedKeyNomTimeout OBJECT-TYPE
SYNTAX Integer32 (15..600)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object defines the starting value of the timeout
for the AP-REQ/REP Backoff and Retry mechanism
with exponential timeout in SNMPv3 provisioning.
If the DHCP option code 122 sub-option 5 is provided to
the MTA, it overwrites this value.
In non secure provisioning mode, the MTA MUST return
a 'genErr' error in response to SNMP GET operations."
REFERENCE
"PacketCable Security Specification"
DEFVAL {30}
::= { pktcMtaDevServer 11 }

pktcMtaDevProvUnsolicitedKeyMeanDev OBJECT-TYPE
SYNTAX Integer32 (15..600)
UNITS "seconds"
MAX-ACCESS read-only
STATUS obsolete
DESCRIPTION
"This is the mean deviation for the round trip delay
timings."
REFERENCE
"PacketCable Security Specification"
::= { pktcMtaDevServer 12 }

pktcMtaDevProvUnsolicitedKeyMaxRetries OBJECT-TYPE

SYNTAX Integer32 (1..32)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object contains a retry counter that applies to
an MTA-initiated AP-REQ/REP key management exchange with
the Provisioning Server in secure SNMPv3 provisioning.
It is the maximum number of retries before the MTA stops
attempting to establish a Security Association with
Provisioning Server.
If the DHCP option code 122 sub-option 5 is provided to
the MTA, it overwrites this value.
In non secure provisioning mode, the MTA MUST return
a 'genErr' error in response to SNMP GET operations."
REFERENCE
"PacketCable Security Specification"
DEFVAL {8}
::= { pktcMtaDevServer 13 }

pktcMtaDevProvKerbRealmName OBJECT-TYPE
SYNTAX SnmpAdminString (SIZE(1..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"For Secure provisioning this object contains the name of
the associated provisioning Kerberos realm acquired during
the MTA4 provisioning step (DHCP Ack)."

Additionally this object value is used as an index into the pktcMtaDevRealmTable. In which case, the upper case ASCII representation of the associated Kerberos realm name MUST be used by both the Manager (SNMP entity) and the MTA. The Kerberos realm name for the Provisioning Server is supplied to the MTA via DHCP option code 122 sub-option 6 as defined in RFC 3495.

For non secure provisioning modes, the value of this object MUST contain the value supplied in the DHCP ACK message (option code 122 sub-option 6)."

REFERENCE

"PacketCable MTA Device Provisioning Specification;
RFC 3495, DHCP Option for CableLabs Client Configuration."
::= { pktcMtaDevServer 14 }

pktcMtaDevProvState OBJECT-TYPE

SYNTAX INTEGER {
operational (1),
waitingForSnmpSetInfo (2),
waitingForTftpAddrResponse (3),
waitingForConfigFile (4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This object defines the MTA provisioning state.

If the state is:

'operational(1)', the device has completed the loading and processing of the initialization parameters.

'waitingForSnmpSetInfo(2)', the device is waiting on its configuration file download access information. Note that this state is only reported when the MTA SNMP enrollment mechanism is used.

'waitingForTftpAddrResponse(3)', the device has sent a DNS request to resolve the server providing the configuration file and it is awaiting for a response. Note that this state is only reported when the MTA SNMP enrollment mechanism is used.

'waitingForConfigFile(4)', the device has sent a request via TFTP or HTTP for the download of its configuration file and it is awaiting for a response or the file download is in progress."

REFERENCE

"PacketCable MTA Device Provisioning Specification,
PacketCable Security Specification"
::= { pktcMtaDevServer 15 }

pktcMtaDevServerDhcp1 OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The IP address of the primary DHCP server which would cater to the MTA during its provisioning. Contains 255.255.255.255 if there was no preference given with respect to the DHCP servers for MTA provisioning."

::= { pktcMtaDevServer 16 }

pktcMtaDevServerDhcp2 OBJECT-TYPE

SYNTAX IpAddress

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION "The IP address of the Secondary DHCP server
which could cater to the MTA during its provisioning.
Contains 0.0.0.0 if there is no specific secondary DHCP
server to be considered during MTA provisioning."
 ::= { pktcMtaDevServer 17 }

```

```

pktcMtaDevTimeServer OBJECT-TYPE
SYNTAX IPAddress
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"IP address of the Time Server from which to obtain the
time. Contains 0.0.0.0 if the Time Protocol is not used for
time synchronization."
 ::= { pktcMtaDevServer 18 }

```

```

--
-- The following group describes the security objects in the MTA
--

```

```

pktcMtaDevManufacturerCertificate OBJECT-TYPE
SYNTAX X509Certificate
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This object contains the MTA Manufacturer Certificate.
The object value must be the ASN.1 DER encoding of the MTA
manufacturer's X.509 public key certificate. The MTA
Manufacturer Certificate is issued to each MTA
manufacturer and is installed into each MTA at the time of
manufacture or with a secure code download. The specific
requirements related to this certificate are defined in
the PacketCable Security specification."
REFERENCE
"PacketCable Security Specification."
 ::= { pktcMtaDevSecurity 1 }

```

```

pktcMtaDevCertificate OBJECT-TYPE
SYNTAX X509Certificate
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"ASN.1 DER encoding of the MTA's X.509 public-key
certificate issued by the manufacturer and installed
into the embedded-MTA in the factory. This certificate,
called MTA Device Certificate, contains the MTA's MAC
address. It cannot be updated by the provisioning server."
 ::= { pktcMtaDevSecurity 2 }

```

```

--*****
--***** THIS OBJECT IS OBSOLETE *****
--*****

```

```

pktcMtaDevSignature OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..256))
MAX-ACCESS read-only
STATUS obsolete

```

DESCRIPTION

"A unique signature created by the MTA for each SNMP Inform or SNMP Trap or SNMP GetResponse message exchanged prior to enabling SNMPv3 security ASN.1 encoded Digital signature in the Cryptographic message syntax (includes nonce). "

::= { pktcMtaDevSecurity 3 }

pktcMtaDevCorrelationId OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Random value generated by the MTA for use in registration authorization. It is for use only in the MTA initialization messages and for MTA configuration file download "

::= { pktcMtaDevSecurity 4 }

--
-- pktcMtaDevSecurityTable
--
-- The pktcMtaDevSecurityTable shows security association information
-- relating to a particular MTA endpoint. The MTA endpoint is indexed
-- with ifIndex.
--

--*****
--***** THIS TABLE IS OBSOLETE *****
--*****

pktcMtaDevSecurityTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcMtaDevSecurityEntry

MAX-ACCESS not-accessible

STATUS obsolete

DESCRIPTION

"Contains per endpoint security information."

::= { pktcMtaDevSecurity 5 }

pktcMtaDevSecurityEntry OBJECT-TYPE

SYNTAX PktcMtaDevSecurityEntry

MAX-ACCESS not-accessible

STATUS obsolete

DESCRIPTION

"List of security attributes for a single PacketCable endpoint interface."

INDEX { ifIndex }

::= { pktcMtaDevSecurityTable 1 }

PktcMtaDevSecurityEntry ::= SEQUENCE {

pktcMtaDevServProviderCertificate X509Certificate,

pktcMtaDevTelephonyCertificate X509Certificate,

pktcMtaDevKerberosRealm OCTET STRING,

pktcMtaDevKerbPrincipalName DisplayString,

pktcMtaDevServGracePeriod Integer32,

pktcMtaDevLocalSystemCertificate X509Certificate,

pktcMtaDevKeyMgmtTimeout1 Integer32,

pktcMtaDevKeyMgmtTimeout2 Integer32

}

pktcMtaDevServProviderCertificate OBJECT-TYPE

SYNTAX X509Certificate

MAX-ACCESS read-write


```

STATUS      obsolete
DESCRIPTION
    "ASN.1 DER encoding of the Telephony Service
    Provider's X.509 public-key certificate, called
    Telephony Service Provider Certificate. It serves
    as the root of the intra-domain trust hierarchy.
    Each MTA is configured with this certificate so
    that it can authenticate TGSSs owned by the same
    service provider. The provisioning server needs
    the ability to update this certificate in the MTAs
    via both SNMP and configuration files"
 ::= { pktcMtaDevSecurityEntry 1 }

```

```

pktcMtaDevTelephonyCertificate OBJECT-TYPE
SYNTAX      X509Certificate
MAX-ACCESS  read-write
STATUS      obsolete
DESCRIPTION
    "ASN.1 DER encoding of the MTA's X.509 public-key
    certificate issued by the Service Provider with either
    the Service Provider CA or a Local System CA. This
    certificate, called MTA Telephony Certificate, contains
    the same public key as the MTA Device Certificate issued
    by the manufacturer. It is used to authenticate the
    identity of the MTA to the TGS (during PKINIT exchanges).
    The provisioning server needs the ability to update this
    certificate in the MTAs via both SNMP and configuration
    files"
 ::= { pktcMtaDevSecurityEntry 2 }

```

```

pktcMtaDevKerberosRealm OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..1280))
MAX-ACCESS  read-write
STATUS      obsolete -- moved to realm table
DESCRIPTION
    "Specifies a Kerberos realm (i.e., administrative domain),
    required for Packet Cable key management."
 ::= { pktcMtaDevSecurityEntry 3 }

```

```

pktcMtaDevKerbPrincipalName OBJECT-TYPE
SYNTAX      DisplayString (SIZE(0..40))
MAX-ACCESS  read-write
STATUS      obsolete
DESCRIPTION
    "Kerberos principal name for the Call Agent. This
    information is required in order for the MTA to obtain
    Call Agent Kerberos tickets. This principal name does not
    include the realm, which is specified as a separate field
    in this configuration file. A Single Kerberos principal
    name MAY be shared among several Call Agents."
 ::= { pktcMtaDevSecurityEntry 4 }

```

```

pktcMtaDevServGracePeriod OBJECT-TYPE
SYNTAX      Integer32 (15..600)
UNITS       "minutes"
MAX-ACCESS  read-write
STATUS      obsolete -- moved to realm table
DESCRIPTION
    "The MTA MUST obtain a new Kerberos ticket (with a PKINIT
    exchange), many minutes before the old ticket expires.
    The minimum allowable value is 15 mins. The default is
    30 mins."
DEFVAL     { 30 }
 ::= { pktcMtaDevSecurityEntry 5 }

```

```

pktcMtaDevLocalSystemCertificate OBJECT-TYPE
    SYNTAX      X509Certificate
    MAX-ACCESS  read-write
    STATUS      obsolete
    DESCRIPTION
        "The Telephony Service Provider CA may delegate the
        issuance of certificates to a regional Certification
        Authority called Local System CA (with the corresponding
        Local System Certificate). This parameter is the ASN.1
        DER encoding of the Local System Certificate. It MUST have
        a non-empty value when the MTA Telephony certificate is
        signed by a Local System CA. Otherwise, the value MUST
        be of length 0."
 ::= { pktcMtaDevSecurityEntry 6 }

```

```

pktcMtaDevKeyMgmtTimeout1 OBJECT-TYPE
    SYNTAX      Integer32 (15..600)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      obsolete -- moved to cms table
    DESCRIPTION
        "This timeout applies only when the MTA initiated key
        management. It is the period during which the MTA will
        save a nonce (inside the sequence number field) from the
        sent out AP Request and wait for the matching AP Reply
        from the CMS."
    REFERENCE
        "PacketCable Security Specification"
 ::= { pktcMtaDevSecurityEntry 7 }

```

```

pktcMtaDevKeyMgmtTimeout2 OBJECT-TYPE
    SYNTAX      Integer32 (15..600)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      obsolete -- changed to adaptive backoff and moved to
                  -- cms table
    DESCRIPTION
        "This timeout applies only when the CMS initiated key
        management (with a Wake Up or Rekey message).
        It is the period during which the MTA will
        save a nonce (inside the sequence number field) from
        the sent out AP Request and wait for the matching AP
        Reply from the CMS."
    REFERENCE
        "PacketCable Security Specification"
 ::= { pktcMtaDevSecurityEntry 8 }

```

```

--
--      Ticket Granting Server information
--
--*****
--***** THIS TABLE IS OBSOLETE *****
--*****

```

```

pktcMtaDevTgsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PkctlMtaDevTgsEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete -- Secure Provisioning ECR
    DESCRIPTION
        "Contains per endpoint Ticket Granting Server information."
 ::= { pktcMtaDevSecurity 8 }

```

```

pktcMtaDevTgsEntry OBJECT-TYPE
    SYNTAX      PktcMtaDevTgsEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete -- Secure Provisioning ECR
    DESCRIPTION
        "List of Tgs attributes for a single packet cable
        endpoint interface."
    INDEX { ifIndex, pktcMtaDevTgsIndex }
    ::= { pktcMtaDevTgsTable 1 }

PktcMtaDevTgsEntry ::= SEQUENCE {
    pktcMtaDevTgsIndex      Integer32,
    pktcMtaDevTgsLocation   DisplayString,
    pktcMtaDevTgsStatus     RowStatus
}

pktcMtaDevTgsIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      obsolete -- Secure Provisioning ECR
    DESCRIPTION
        "Index into the TGS table for TGS locations.
        IfType specifies the endpoint, TgsIndex specifies a TGS."
    ::= { pktcMtaDevTgsEntry 1 }

pktcMtaDevTgsLocation OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    MAX-ACCESS  read-create
    STATUS      obsolete -- Secure Provisioning ECR
    DESCRIPTION
        "Name of the TGS Ticket Granting Server, which is the
        Kerberos Server. This parameter is a FQDN or Ipv4 address.
        There may be multiple entries of this type. The order
        in which these entries are listed is the priority order
        in which the MTA will attempt to contact them for this
        endpoint."
    ::= { pktcMtaDevTgsEntry 2 }

pktcMtaDevTgsStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      obsolete -- Secure Provisioning ECR
    DESCRIPTION
        "This object contains the Row Status associated with
        the pktcMtaDevTgsTable."
    ::= { pktcMtaDevTgsEntry 3 }

pktcMtaDevTelephonyRootCertificate OBJECT-TYPE
    SYNTAX      X509Certificate
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "ASN.1 DER encoding of the IP Telephony Root X.509
        public-key certificate stored in the MTA non-volatile
        memory and updateable with a code download. This
        certificate is used to validate the initial AS Reply
        from the KDC received during the MTA initialization."
    ::= { pktcMtaDevSecurity 9 }

-----
--
--   Procedures for setting up security associations:
--
--   A security association may be setup either via configuration or via

```

```

-- NCS signalling.
--
-- I. Security association set up via configuration.
--
-- The realm must be configured first. Associated with the
-- realm is a KDC. The realm table (pktcMtaDevRealmTable)
-- indicates information about realm (e.g., name,
-- organization name) and parameters associated with KDC
-- communications (e.g., grace periods, AS request/AS
-- reply adaptive backoff parameters).
--
-- Once the realm is established, one or more servers may be
-- defined in the realm. For PacketCable, these are
-- Call Management Servers (CMSs). Associated with each CMS
-- entry in the pktcMtaDevCmsTable is an explicit reference
-- to a Realm via the realm index
-- (pktcMtaDevCmsKerbRealmName), the FQDN of the CMS,
-- and parameters associated with IPsec management with the
-- CMS (e.g., clock skew, AP request/
-- AP reply adaptive backoff parameters).
--
--
--
-- II. Security association setup via NCS signalling
--
--
-- NOTE - The following process is done automatically by the
-- MTA. The NCS is not involved in creating signaled entries.
-- The current CMS signalling association being used by an
-- endpoint is marked as active in CMS MAP table. If NCS
-- signalling requests a change of signalling association to
-- a different FQDN, the MTA checks the current CMS MAP
-- table entries for the affected endpoint. If the entry
-- exists in the CMS MAP table, the current CMS MAP table
-- entry is marked inactive and the newly chosen CMS MAP
-- table entry is marked active.
--
-- If the entry does not exist in the CMS MAP table, the
-- CMS table is checked to determine whether or not it
-- contains the CMS specified by CMS signalling (possibly
-- a redirection). If the desired CMS entry is defined,
-- then a corresponding entry is created and an entry in
-- the CMS MAP table is created. If the MTA does not
-- have current associations with that CMS, it will now
-- perform key management to establish required security
-- associations. Once the desired CMS entry is established,
-- the current CMS MAP table entry is marked inactive and
-- the newly created CMS MAP table entry is marked active.
-- Otherwise the current CMS MAP table entry remains
-- active and the newly created CMS MAP table entry is marked
-- inactive.
--
-- If the entry does not exist in the CMS MAP table and the
-- CMS entry does not exist in the CMS table, a new CMS
-- table entry should be created. This CMS entry should use
-- the same realm as used by this endpoint. The default
-- values for the clock skew and AP request/AP reply adaptive
-- backoff parameters should be used. The MTA will now
-- perform key management to establish required security
-- associations. Once the desired CMS entry is established,
-- the current CMS MAP table entry is marked inactive and
-- the newly created CMS MAP table entry is marked active.

```

```

--      Otherwise the current CMS MAP table entry remains
--      active and the newly created CMS MAP table entry is
--      marked inactive.
--
--      III.  When the MTA receives wake-up or rekey messages from a CMS,
--            it performs key management based on the corresponding entry
--            in the CMS table.  If the matching CMS entry does not exist,
--            it must ignore the wake-up or rekey messages.
--
--=====

```

```

--=====
--
--      pktcMtaDevRealmTable
--
--      The pktcMtaDevRealmTable shows the KDC realms.  The table is
--      indexed withpktcMtaDevRealmName.  The Realm Table is used in conjunction
--      with any server which needs a security association with an
--      server MTA.  The table (today the CMS) has a security association.
--      Each server-MTA security association is associated with a
--      single Realm.  This allows for multiple realms, each
--      with its own security association.
--
--=====

```

```

pktcMtaDevRealmTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcMtaDevRealmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Contains per Kerberos realm security parameters."
    ::= { pktcMtaDevSecurity 16 }

```

```

pktcMtaDevRealmEntry  OBJECT-TYPE
    SYNTAX      PktcMtaDevRealmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of security parameters for a single Kerberos realm."
    INDEX { IMPLIED pktcMtaDevRealmName }
    ::= { pktcMtaDevRealmTable 1 }

```

```

PktcMtaDevRealmEntry ::= SEQUENCE {
    pktcMtaDevRealmName          SnmpAdminString,
    pktcMtaDevRealmPkinitGracePeriod  Integer32,
    pktcMtaDevRealmTgsGracePeriod    Integer32,
    pktcMtaDevRealmOrgName          OCTET STRING,
    pktcMtaDevRealmUnsolicitedKeyMaxTimeout  Integer32,
    pktcMtaDevRealmUnsolicitedKeyNomTimeout  Integer32,
    pktcMtaDevRealmUnsolicitedKeyMeanDev    Integer32,
    pktcMtaDevRealmUnsolicitedKeyMaxRetries  Integer32,
    pktcMtaDevRealmStatus          RowStatus
}

```

```

pktcMtaDevRealmName  OBJECT-TYPE
    SYNTAX      SnmpAdminString(SIZE(1..255))
    MAX-ACCESS  not-accessible
    STATUS      current

```

DESCRIPTION

"The corresponding Kerberos Realm name. This is used as an index into pktcMtaDevRealmTable. When used as an index, used by both the Manager(SNMPv3 Entity) and the MTA."
 ::= { pktcMtaDevRealmEntry 1 }

pktcMtaDevRealmPkinitGracePeriod OBJECT-TYPE

SYNTAX Integer32 (15..600)

UNITS "minutes"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"For the purposes of the key management with an Application Server (CMS or Provisioning Server), the MTA MUST obtain a new Kerberos ticket (with a PKINIT exchange), many minutes before the old ticket expires. The minimum allowable value is 15 mins. The default is 30 mins. This parameter MAY also be used with other Kerberized applications."

DEFVAL { 30 }

::= { pktcMtaDevRealmEntry 2 }

pktcMtaDevRealmTgsGracePeriod OBJECT-TYPE

SYNTAX Integer32 (1..600)

UNITS "minutes"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"When the MTA implementation uses TGS Request/TGS Reply Kerberos messages for the purpose of the key management with an Application Server (CMS or Provisioning Server), the MTA MUST obtain a new service ticket for the Application Server (with a TGS Request), many minutes before the old ticket expires. The minimum allowable value is 1 min. The default is 10 mins. This parameter MAY also be used with other Kerberized applications."

DEFVAL { 10 }

::= { pktcMtaDevRealmEntry 3 }

pktcMtaDevRealmOrgName OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (1..64))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value of the X.500 organization name attribute in the subject name of the Service provider certificate"

::= { pktcMtaDevRealmEntry 4 }

--
-- Unsolicited Key Updates are based on an exponential backoff
-- mechanism with two timers for AS replies. The backoff timers has a
-- maximum value of pktcMtaDevRealmUnsolicitedKeyMaxTimeout seconds
-- and a nominal timer has a pktcMtaDevRealmUnsolicitedKeyNomTimeout
-- seconds from which the backoff timer determinations are made.
-- After pktcMtaDevRealmUnsolicitedMaxRetries have occurred no more
-- attempts are made.
--

```

pktcMtaDevRealmUnsolicitedKeyMaxTimeout OBJECT-TYPE
    SYNTAX Integer32 (1..600)
    UNITS "seconds"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This timeout applies only when the MTA initiated key
        management. The maximum timeout is the value which may not
        be exceeded in the exponential backoff algorithm. If
        provided, DHCP-Option-122-Sub-option 4 overrides this value."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 30 }
    ::= { pktcMtaDevRealmEntry 5 }

```

```

pktcMtaDevRealmUnsolicitedKeyNomTimeout OBJECT-TYPE
    SYNTAX Integer32 (100..600000)
    UNITS "milliseconds"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Defines the starting value of the timeout for the AS-REQ/REP
        Backoff and Retry mechanism with exponential timeout. If
        provided, DHCP-Option-122-Sub-option 4 overrides this
        value."
    REFERENCE
        "PacketCable Security Specification,
        PacketCable Provisioning Specification"
    DEFVAL { 10000 }
    ::= { pktcMtaDevRealmEntry 6 }

```

```

pktcMtaDevRealmUnsolicitedKeyMeanDev OBJECT-TYPE
    SYNTAX Integer32 (1..600)
    UNITS "seconds"
    MAX-ACCESS read-only
    STATUS obsolete
    DESCRIPTION
        "This is a measurement of the mean deviation for the round
        trip delay timings."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 2 }
    ::= { pktcMtaDevRealmEntry 7 }

```

```

pktcMtaDevRealmUnsolicitedKeyMaxRetries OBJECT-TYPE
    SYNTAX Integer32 (0..1024)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is the maximum number of retries before the MTA
        gives up attempting to establish a security association.
        If provided, DHCP-Option-122-Sub-option 4 overrides this
        value."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 5 }
    ::= { pktcMtaDevRealmEntry 8 }

```

```

pktcMtaDevRealmStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current

```

```

DESCRIPTION
    "This object contains the Row Status associated with
    the pktcMtaDevRealmTable."
 ::= { pktcMtaDevRealmEntry 9 }

```

```

-----
--
-- pktcMtaDevCmsTable
--
-- The pktcMtaDevCmsTable shows the IPsec key management policy
-- relating to a particular CMS. The table is indexed with
-- pktcMtaDevCmsFQDN.
--
-----

```

```

pktcMtaDevCmsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcMtaDevCmsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Contains per CMS key management policy."
    ::= { pktcMtaDevSecurity 17 }

```

```

pktcMtaDevCmsEntry OBJECT-TYPE
    SYNTAX      PktcMtaDevCmsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of key management parameters for a single MTA-CMS
        interface."
    INDEX { IMPLIED pktcMtaDevCmsFqdn }
    ::= { pktcMtaDevCmsTable 1 }

```

```

PktcMtaDevCmsEntry ::= SEQUENCE {
    pktcMtaDevCmsFqdn          SnmpAdminString,
    pktcMtaDevCmsKerbRealmName SnmpAdminString,
    pktcMtaDevCmsSolicitedKeyTimeout Integer32,
    pktcMtaDevCmsMaxClockSkew Integer32,
    pktcMtaDevCmsUnsolicitedKeyMaxTimeout Integer32,
    pktcMtaDevCmsUnsolicitedKeyNomTimeout Integer32,
    pktcMtaDevCmsUnsolicitedKeyMeanDev Integer32,
    pktcMtaDevCmsUnsolicitedKeyMaxRetries Integer32,
    pktcMtaDevCmsStatus      RowStatus,
    pktcMtaDevCmsIpsecCtrl   TruthValue
}

```

```

pktcMtaDevCmsFqdn OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..255))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the fully qualified
        domain name of the CMS.
        When used as an index, the upper case ASCII
        representation of the associated CMS FQDN
        MUST be used by both the SNMP Manager and the MTA."
    ::= { pktcMtaDevCmsEntry 1 }

```

```

pktcMtaDevCmsKerbRealmName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..255))
    MAX-ACCESS  read-create
    STATUS      current

```



```

DESCRIPTION
    "The Kerberos Realm Name of the associated CMS. This is
    the index into the pktcMtaDevRealmTable.
    When used as an index, the upper case ASCII
    representation of the associated CMS FQDN
    must be used by both the SNMP Manager and the MTA "
 ::= { pktcMtaDevCmsEntry 2 }

```

```

pktcMtaDevCmsMaxClockSkew    OBJECT-TYPE
    SYNTAX      Integer32 (1..1800)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This is the maximum allowable clock skew between the
        MTA and CMS"
    DEFVAL { 300 }
    ::= { pktcMtaDevCmsEntry 3 }

```

```

pktcMtaDevCmsSolicitedKeyTimeout  OBJECT-TYPE
    SYNTAX Integer32 (100..30000)
    UNITS "milliseconds"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This timeout applies only when the CMS initiated key
        management(with a Wake Up or Rekey message). It is the
        period during which the MTA will save a nonce (inside the
        sequence number field) from the sent out AP Request and
        wait for the matching AP Reply from the CMS."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 1000 }
    ::= { pktcMtaDevCmsEntry 4 }

```

```

-----
--
-- Unsolicited Key Updates are based on an exponential backoff
-- mechanism with two timers for AP replies. The
-- backoff timers have a maximum value of
-- pktcMtaDevCmsUnsolicitedKeyMaxTimeout
-- seconds and a nominal timer has
-- pktcMtaDevCmsUnsolicitedKeyNomTimeout seconds from which the
-- backoff timer determinations are made. After
-- pktcMatDevCmsUnsolicitedMaxRetries have occurred no more
-- attempts are made.
--
-----

```

```

pktcMtaDevCmsUnsolicitedKeyMaxTimeout  OBJECT-TYPE

    SYNTAX      Integer32 (1..600)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This timeout applies only when the MTA initiated key.
        The maximum management timeout is the value which may not
        be exceeded in the exponential backoff algorithm."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 8 }
    ::= { pktcMtaDevCmsEntry 5 }

```

```

pktcMtaDevCmsUnsolicitedKeyNomTimeout OBJECT-TYPE
    SYNTAX Integer32 (100..30000)
    UNITS "milliseconds"
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Defines the starting value of the timeout for the
        AP-REQ/REP Backoff and Retry mechanism with exponential
        timeout for CMS."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 500 }
    ::= { pktcMtaDevCmsEntry 6 }

pktcMtaDevCmsUnsolicitedKeyMeanDev OBJECT-TYPE
    SYNTAX Integer32 (1..600)
    UNITS "seconds"
    MAX-ACCESS read-only
    STATUS obsolete
    DESCRIPTION
        "This is the measurement of the mean deviation for the
        round trip delay timings."
    REFERENCE
        "PacketCable Security Specification"
    ::= { pktcMtaDevCmsEntry 7 }

pktcMtaDevCmsUnsolicitedKeyMaxRetries OBJECT-TYPE

    SYNTAX Integer32 (0..1024)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This is the maximum number of retries before the MTA
        gives up attempting to establish a security association."
    REFERENCE
        "PacketCable Security Specification"
    DEFVAL { 5 }
    ::= { pktcMtaDevCmsEntry 8 }

pktcMtaDevCmsStatus OBJECT-TYPE

    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "This object contains the Row Status associated with the
        pktcMtaDevCmsTable."
    ::= { pktcMtaDevCmsEntry 9 }

pktcMtaDevCmsIpsecCtrl OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of 'true(1)' indicates that IPSEC and IPSEC
        KeyManagement MUST be used to communicate with the CMS.
        The value of 'false(2)' indicates that IPSEC Signalling
        Security is disabled for both the IPSEC Key Management and
        IPSECprotocol (for the specific CMS)."
    DEFVAL { true }
    ::= { pktcMtaDevCmsEntry 10 }

```

```

-----
--
--      pktcMtaCmsMapTable
--*** this table is obsolete ***
--
--
-- The pktcMtaCmsMapTable contains the signalling associations
-- between MTA endpoints and CMSs. It maps the endpoint to
-- zero or more entries in pktcMtaDevCmsTable.
--
-- The table contains the following indexes and rows:
--
-- ifIndex - the index of the physical port
--
-- pktcMtaCmsMapCmsIndex - the index of the CMS entry in the
-- pktcMtaDevCmsTable. Valid indices are equal to current
-- pktcMtaDevCmsIndex values.
--
-- pktcMtaCmsMapOperStatus - this value indicates which signalling
-- association the endpoint is actively using
--
-- pktcMtaCmsMapAdminStatus - this flag indicates whether or not
-- an endpoint should use a particular CMS and its security
-- association. By setting this flag to inhibit, this associated
-- CMS cannot provide signalling to the referenced endpoint.
--
-- pktcMtaCmsMapRowStatus - allows for the creation and deletion of
-- endpoint mappings via the NMS
--
--
-----

```

```

pktcMtaCmsMapTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcMtaCmsMapEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "Contains per endpoint CMS signalling associations."
    ::= { pktcMtaDevSecurity 18 }

```

```

pktcMtaCmsMapEntry OBJECT-TYPE
    SYNTAX      PktcMtaCmsMapEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "List of signalling associations."
    INDEX { ifIndex, pktcMtaCmsMapCmsFqdn }
    ::= { pktcMtaCmsMapTable 1 }

```

```

PktcMtaCmsMapEntry ::= SEQUENCE {
    pktcMtaCmsMapCmsFqdn DisplayString,
    pktcMtaCmsMapOperStatus INTEGER,
    pktcMtaCmsMapAdminStatus INTEGER,
    pktcMtaCmsMapRowStatus RowStatus
}

```

```

pktcMtaCmsMapCmsFqdn OBJECT-TYPE
    SYNTAX      DisplayString (SIZE(1..255))
    MAX-ACCESS  not-accessible
    STATUS      obsolete

```

DESCRIPTION
 "The index for the associated CMS. Valid indices are equal to current pktcMtaDevCmsFqdn values."
 ::= { pktcMtaCmsMapEntry 1 }

pktcMtaCmsMapOperStatus OBJECT-TYPE

SYNTAX INTEGER {
 inactive (1),
 active (2)
 }

MAX-ACCESS read-only

STATUS obsolete

DESCRIPTION

"The operational status of signalling association. The meaning of the status is as follows:

inactive - signalling is not currently active

active - signalling is active."

::= { pktcMtaCmsMapEntry 2 }

pktcMtaCmsMapAdminStatus OBJECT-TYPE

SYNTAX INTEGER {
 inhibit (1),
 allow (2)
 }

MAX-ACCESS read-create

STATUS obsolete

DESCRIPTION

"The administrative status for signalling over the indicated security association. The meaning of the status is as follows:

inhibit - signalling is not currently allowed

allow - signalling is allowed."

::= { pktcMtaCmsMapEntry 3 }

pktcMtaCmsMapRowStatus OBJECT-TYPE

SYNTAX RowStatus
 MAX-ACCESS read-create
 STATUS obsolete

DESCRIPTION

"This object is used for creating and deleting an entry in this table via an element manager."

::= { pktcMtaCmsMapEntry 4 }

pktcMtaDevResetKrbTickets OBJECT-TYPE

SYNTAX BITS {
 invalidateProvOnReboot (0),
 invalidateAllCmsOnReboot (1)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object defines a Kerberos Ticket Control Mask that instructs the MTA to invalidate the specific Application Server Kerberos Ticket(s) that are stored locally in the MTA NVRAM (non-volatile or persistent memory).

If the MTA does not store Kerberos tickets in NVRAM, it MUST ignore setting of this object, and MUST report a BITS value of zero when the object is read.

If the MTA supports Kerberos tickets storage in NVRAM, the object value is encoded as follows:

- Setting the invalidateProvOnReboot bit (bit 0) to 1

means that the MTA MUST invalidate the Kerberos Application Ticket(s) for the Provisioning Application

at the next MTA reboot (if secure SNMP provisioning mode

- is used). In non secure provisioning modes, the MTA MUST return an 'inconsistentValue' in response to SNMP SET operations with a bit 0 set to 1.
- Setting the invalidateAllCmsOnReboot bit (bit 1) to 1 means that the MTA MUST invalidate the Kerberos Application Ticket(s) for all CMSEs currently assigned to the MTA endpoints."

REFERENCE

"PacketCable Security Specification"

DEFVAL {{ }}

::= { pktcMtaDevSecurity 19 }

--

-- notification group is for future extension.

--

pktcMtaNotificationPrefix OBJECT IDENTIFIER ::= { pktcMtaMib 2 }

pktcMtaNotification OBJECT IDENTIFIER ::= {

 pktcMtaNotificationPrefix 0 }

pktcMtaConformance OBJECT IDENTIFIER ::= { pktcMtaMib 3 }

pktcMtaCompliances OBJECT IDENTIFIER ::= { pktcMtaConformance 1 }

pktcMtaGroups OBJECT IDENTIFIER ::= { pktcMtaConformance 2 }

--

-- Notification Group

--

pktcMtaDevProvisioningEnrollment NOTIFICATION-TYPE

OBJECTS {

 sysDescr,
 pktcMtaDevSwCurrentVers,
 pktcMtaDevTypeIdentifier,
 pktcMtaDevMacAddress,
 pktcMtaDevCorrelationId

}

STATUS current

DESCRIPTION

"This INFORM notification is issued by the MTA to initiate the PacketCable provisioning process when the MTA SNMP enrollment mechanism is used.

It contains the system description, the current software version, the MTA device type identifier, the MTA MAC address (obtained in the MTA ifTable in the ifPhysAddress object that corresponds to the ifIndex 1) and a correlation ID."

::= { pktcMtaNotification 1 }

pktcMtaDevProvisioningStatus NOTIFICATION-TYPE

OBJECTS {

 pktcMtaDevMacAddress,
 pktcMtaDevCorrelationId,
 pktcMtaDevProvisioningState

}

STATUS current

DESCRIPTION

"This INFORM notification may be issued by the MTA to confirm the completion of the PacketCable provisioning process, and to report its provisioning completion status.

It contains the MTA MAC address (obtained in the MTA ifTable in the ifPhysAddress object that corresponds to the ifIndex 1), a correlation ID and the MTA provisioning state as defined in pktcMtaDevProvisioningState."

::= { pktcMtaNotification 2 }

-- compliance statements

```

pktcMtaBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for devices that implement
        MTA feature."
    MODULE --pktcMtaMib
-- unconditionally mandatory groups
    MANDATORY-GROUPS {
        pktcMtaGroup,
        pktcMtaNotificationGroup
    }
    ::= { pktcMtaCompliances 3 }
pktcMtaGroup OBJECT-GROUP
    OBJECTS {
        pktcMtaDevResetNow,
        pktcMtaDevSerialNumber,
        pktcMtaDevMacAddress,
        pktcMtaDevFQDN,
        pktcMtaDevEndPntCount,
        pktcMtaDevEnabled,
        pktcMtaDevTypeIdentifier,
        pktcMtaDevProvisioningState,
        pktcMtaDevHttpAccess,
        pktcMtaDevCertificate,
        pktcMtaDevCorrelationId,
        pktcMtaDevManufacturerCertificate,
        pktcMtaDevServerDhcp1,
        pktcMtaDevServerDhcp2,
        pktcMtaDevServerDns1,
        pktcMtaDevServerDns2,
        pktcMtaDevTimeServer,
        pktcMtaDevConfigFile,
        pktcMtaDevSnmpEntity,
        pktcMtaDevRealmPkinitGracePeriod,
        pktcMtaDevRealmTgsGracePeriod,
        pktcMtaDevRealmOrgName,
        pktcMtaDevRealmUnsolicitedKeyMaxTimeout,
        pktcMtaDevRealmUnsolicitedKeyNomTimeout,
        pktcMtaDevRealmUnsolicitedKeyMaxRetries,
        pktcMtaDevRealmStatus,
        pktcMtaDevCmsKerbRealmName,
        pktcMtaDevCmsUnsolicitedKeyMaxTimeout,
        pktcMtaDevCmsUnsolicitedKeyNomTimeout,
        pktcMtaDevCmsUnsolicitedKeyMaxRetries,
        pktcMtaDevCmsSolicitedKeyTimeout,
        pktcMtaDevCmsMaxClockSkew,
        pktcMtaDevCmsStatus,
        pktcMtaDevProvUnsolicitedKeyMaxTimeout,
        pktcMtaDevProvUnsolicitedKeyNomTimeout,
        pktcMtaDevProvUnsolicitedKeyMaxRetries,
        pktcMtaDevProvKerbRealmName,
        pktcMtaDevProvSolicitedKeyTimeout,
        pktcMtaDevProvConfigHash,
        pktcMtaDevProvConfigKey,
        pktcMtaDevProvState,
        pktcMtaDevProvisioningTimer,
        pktcMtaDevTelephonyRootCertificate,
        pktcMtaDevErrorOid,
        pktcMtaDevErrorGiven,
        pktcMtaDevErrorReason,
        pktcMtaDevSwCurrentVers,
        pktcMtaDevResetKrbTickets,
        pktcMtaDevCmsIpsecCtrl,
        pktcMtaDevProvisioningCounter
    }
}

```

```

STATUS      current
DESCRIPTION
    "Group of objects for PacketCable MTA MIB."
 ::= { pktcMtaGroups 1 }
pktcMtaNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS {
    pktcMtaDevProvisioningStatus,
    pktcMtaDevProvisioningEnrollment
}
STATUS      current
DESCRIPTION
    "These notifications deal with change in status of
    MTA Device."
 ::= { pktcMtaGroups 2 }

pktcMtaObsoleteGroup    OBJECT-GROUP
OBJECTS {
    pktcMtaDevHardwareVersion,
    pktcMtaDevSignature,
    pktcMtaDevServProviderCertificate,
    pktcMtaDevTelephonyCertificate,
    pktcMtaDevKerberosRealm,
    pktcMtaDevKerbPrincipalName,
    pktcMtaDevServGracePeriod,
    pktcMtaDevLocalSystemCertificate,
    pktcMtaDevKeyMgmtTimeout1,
    pktcMtaDevTgsLocation,
    pktcMtaDevTgsStatus,
    pktcMtaDevServerBootState,
    pktcMtaCmsMapOperStatus,
    pktcMtaCmsMapAdminStatus,
    pktcMtaCmsMapRowStatus,
    pktcMtaDevRealmUnsolicitedKeyMeanDev,
    pktcMtaDevCmsUnsolicitedKeyMeanDev,
    pktcMtaDevProvUnsolicitedKeyMeanDev,
    pktcMtaDevServerDhcp,
    pktcMtaDevKeyMgmtTimeout2
}
STATUS      obsolete
DESCRIPTION
    "Group of obsolete objects for PacketCable MTA MIB."
 ::= { pktcMtaGroups 3 }

END

```

Annex C

Network Call Signalling MIB

The NCS MIB MUST be implemented as defined below.

```
PKTC-SIG-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```

    MODULE-IDENTITY,
    OBJECT-TYPE,
    Integer32,
    IPAddress,

```

```

BITS
    FROM SNMPv2-SMI
TEXTUAL-CONVENTION,
RowStatus,
TruthValue
    FROM SNMPv2-TC
OBJECT-GROUP,
MODULE-COMPLIANCE
    FROM SNMPv2-CONF
SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
clabProjPacketCable
    FROM CLAB-DEF-MIB
ifIndex
    FROM IF-MIB;

```

```

pktcSigMib MODULE-IDENTITY
    LAST-UPDATED      "200501280000Z" -- January 28, 2005
    ORGANIZATION      "CableLabs -- PacketCable OSS Group"
    CONTACT-INFO
        "Sumanth Channabasappa
        Postal: CableLabs, Inc.
              858 Coal Creek Circle
              Louisville, CO 80027-9750
              U.S.A.
        Phone: +1 303-661-9100
        Fax:   +1 303-661-9199
        E-mail: mibs@cablelabs.com"

```

```

DESCRIPTION
    "This MIB module supplies the basic management
    object for the PacketCable Signalling
    protocols. This version of the MIB includes
    common signalling and Network Call Signalling
    (NCS) related signalling objects.
    Acknowledgements:
    Angela Lyda           Arris Interactive
    Sasha Medvinsky      Motorola
    Roy Spitzer          Telogy Networks, Inc.
    Rick Vetter          Motorola
    Itay Sherman         Texas Instruments
    Klaus Hermanns       Cisco Systems
    Eugene Nechamkin     Broadcom Corp.
    Satish Kumar         Texas Instruments
    Copyright 1999-2005 Cable Television Laboratories, Inc.
    All rights reserved."

```

```

REVISION "200501280000Z"
DESCRIPTION
    "This revision, published as part of the PacketCable
    1.5 Signalling MIB I01 Specification."
 ::= { clabProjPacketCable 2 }

```

```

PktcCodecType ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION

```

```

    "Textual Convention defines various types of
    CODECs that MAY be supported. The list of CODECs
    MUST be consistent with the Codec RTP MAP Parameters
    Table in the PacketCable CODEC specification. In-line
    embedded comments below contain the Literal Codec Name
    for each CODEC. The Literal Codec Name corresponds to
    the second column of the Codec RTP MAP Parameters Table.
    The Literal Codec Name Column contains the CODEC name
    that is used in the LCD of the NCS messages CRCX/MDCX,
    and is also used to identify the CODEC in the CMS

```


Provisioning Specification. The RTP Map Parameter Column of the Codec RTP MAP Parameters Table contains the string used in the media attribute line ('a=') of the SDP parameters in NCS messages."

REFERENCE

"PacketCable CODEC Specification"

SYNTAX INTEGER {
 other (1),
 unknown (2),
 g729 (3), -- G729
 reserved (4), -- reserved for future use
 g729E (5), -- G729E
 pcmu (6), -- PCMU
 g726at32 (7), -- G726-32
 g728 (8), -- G728
 pcma (9), -- PCMA
 g726at16 (10), -- G726-16
 g726at24 (11), -- G726-24
 g726at40 (12), -- G726-40
 ilbc (13), -- iLBC
 bv16 (14) -- BV16
}

PkctRingCadence ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This object represents a ring cadence in bit string format. The ring cadence representation starts with the first 1 in the pattern (the leading 0s in the MSB are padding and are to be ignored). Each bit represents 100ms of tone; 1 is tone, 0 is no tone. 64 bits MUST be used for cadence representation, LSB 4 bits are used for representing repeatable characteristics. 0000 means repeatable, and 1000 means non repeatable. During SNMP SET operations 64 bits MUST be used, otherwise MTA MUST reject the value. As an example, the hex representation of a ring cadence of 0.5 secs on; 4 secs off; repeatable would be:0x0001F00000000000."

SYNTAX BITS {
 interval1 (0),
 interval2 (1),
 interval3 (2),
 interval4 (3),
 interval5 (4),
 interval6 (5),
 interval7 (6),
 interval8 (7),
 interval9 (8),
 interval10 (9),
 interval11 (10),
 interval12 (11),
 interval13 (12),
 interval14 (13),
 interval15 (14),
 interval16 (15),
 interval17 (16),
 interval18 (17),
 interval19 (18),
 interval20 (19),
 interval21 (20),
 interval22 (21),
 interval23 (22),
 interval24 (23),
 interval25 (24),

```

    interval26 (25),
    interval27 (26),
    interval28 (27),
    interval29 (28),
    interval30 (29),
    interval31 (30),
    interval32 (31),
    interval33 (32),
    interval34 (33),
    interval35 (34),
    interval36 (35),
    interval37 (36),
    interval38 (37),
    interval39 (38),
    interval40 (39),
    interval41 (40),
    interval42 (41),
    interval43 (42),
    interval44 (43),
    interval45 (44),
    interval46 (45),
    interval47 (46),
    interval48 (47),
    interval49 (48),
    interval50 (49),
    interval51 (50),
    interval52 (51),
    interval53 (52),
    interval54 (53),
    interval55 (54),
    interval56 (55),
    interval57 (56),
    interval58 (57),
    interval59 (58),
    interval60 (59),
    interval61 (60),
    interval62 (61),
    interval63 (62),
    interval64 (63)
}

```

```

PktcSigType ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "These are the various types of signalling that
        may be supported.
        ncs - network call signalling a derivation of MGCP
        (Media Gateway Control Protocol) version 1.0
        dcs - distributed call signalling a derivation
        of SIP (Session Initiation Protocol) RFC 3261"
    SYNTAX INTEGER {
        other(1),
        unknown(2),
        ncs(3),
        dcs(4)
    }

```

```

pktcSigMibObjects OBJECT IDENTIFIER
    ::= { pktcSigMib 1 }
pktcSigDevConfigObjects OBJECT IDENTIFIER
    ::= { pktcSigMibObjects 1 }
pktcNcsEndPntConfigObjects OBJECT IDENTIFIER
    ::= { pktcSigMibObjects 2 }

```

```

pktcSigEndPntConfigObjects OBJECT IDENTIFIER
                               ::= { pktcSigMibObjects 3 }
pktcDcsEndPntConfigObjects OBJECT IDENTIFIER
                               ::= { pktcSigMibObjects 4 }

--
-- The pktcSigDevCodecTable defines the codecs supported by this
-- Media Terminal Adapter (MTA). There is one entry for each
-- codecs supported.
--

pktcSigDevCodecTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PktcSigDevCodecEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table describes the MTA supported codec types."
    ::= { pktcSigDevConfigObjects 1 }

pktcSigDevCodecEntry OBJECT-TYPE
    SYNTAX PktcSigDevCodecEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "List of supported codecs types for the MTA."
    INDEX { pktcSigDevCodecIndex }
    ::= { pktcSigDevCodecTable 1 }

PktcSigDevCodecEntry ::= SEQUENCE {
    pktcSigDevCodecIndex Integer32,
    pktcSigDevCodecType PktcCodecType,
    pktcSigDevCodecMax Integer32
}

pktcSigDevCodecIndex OBJECT-TYPE
    SYNTAX Integer32 (1..16383)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The index value which uniquely identifies an entry
        in the pktcSigDevCodecTable."
    ::= { pktcSigDevCodecEntry 1 }

pktcSigDevCodecType OBJECT-TYPE
    SYNTAX PktcCodecType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A codec type supported by this MTA."
    ::= { pktcSigDevCodecEntry 2 }

pktcSigDevCodecMax OBJECT-TYPE
    SYNTAX Integer32(1..16383)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The maximum number of simultaneous sessions of the
        specific codec that the MTA can support"
    ::= { pktcSigDevCodecEntry 3 }

```



```

pktcSigTosFormatSelector OBJECT-TYPE
    SYNTAX      INTEGER {
        ipv4TOSOctet(1),
        dscpCodepoint(2)
    }
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The format of the default signalling and media
        Type of Service (TOS) values."
    DEFVAL { ipv4TOSOctet }
    ::= { pktcSigDevConfigObjects 10 }

--
--      pktcSigCapabilityTable - This table defines the valid signalling
--      types supported by this MTA.
--

pktcSigCapabilityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcSigCapabilityEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table describes the signalling types by this MTA."
    ::= { pktcSigDevConfigObjects 11 }

pktcSigCapabilityEntry OBJECT-TYPE
    SYNTAX      PktcSigCapabilityEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Entries in pktcMtaDevSigCapabilityTable - List of
        supported signalling types, versions and vendor extensions
        for this MTA. Each entry in the list provides for one
        signalling type and version combination. If the device
        supports multiple versions of the same signalling type -
        it will require multiple entries."
    INDEX { pktcSignallingIndex }
    ::= { pktcSigCapabilityTable 1 }

PktcSigCapabilityEntry ::= SEQUENCE {
    pktcSignallingIndex      Integer32,
    pktcSignallingType       PktcSigType,
    pktcSignallingVersion    SnmpAdminString,
    pktcSignallingVendorExtension SnmpAdminString
}

pktcSignallingIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The index value which uniquely identifies
        an entry in the pktcSigCapabilityTable."
    ::= { pktcSigCapabilityEntry 1 }

pktcSignallingType OBJECT-TYPE
    SYNTAX      PktcSigType
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The Type identifies the type of signalling
        used, this can be NCS, DCS, etc. This value

```

```

        has to be associated with a single signalling
        version - reference pktcMtaDevSignallingVersion."
 ::= { pktcSigCapabilityEntry 2 }

pktcSignallingVersion      OBJECT-TYPE
SYNTAX      SnmpAdminString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Provides the version of the signalling type -
    reference pktcSignallingType. Examples
    would be 1.0 or 2.33 etc."
 ::= { pktcSigCapabilityEntry 3 }

pktcSignallingVendorExtension  OBJECT-TYPE
SYNTAX      SnmpAdminString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The vendor extension allows vendors to
    provide a list of additional capabilities,
    vendors can decide how to encode these
    Extensions, although space separated text is
    suggested."
 ::= { pktcSigCapabilityEntry 4 }

pktcSigDefNcsReceiveUdpPort  OBJECT-TYPE
SYNTAX      Integer32 (1025..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object contains the MTA User Datagram Protocol
    (UDP) receive port that is being used for NCS call
    signalling. This object should only be changed by the
    configuration file."
REFERENCE
    "Refer to NCS specification"
DEFVAL { 2427 }
 ::= { pktcSigDevConfigObjects 12 }

pktcSigServiceClassNameUS  OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..15))
MAX-ACCESS  read-write
STATUS      obsolete
DESCRIPTION
    "This object contains a string indicating the Service
    Class name to create an Upstream Service (US) Flow for
    NCS. If the object has an empty string value then the
    upstream NCS SF is not created and the best effort
    SF is used for upstream NCS data. The creation of the NCS
    SF primary occurs before Voice Communication Service is
    activated on the device. If this object is set to a
    non-empty (non-zero length) string, the MTA MUST create
    the NCS SF if it does not currently exist and the
    pktcSigServiceClassNameMask object has a non-zero value.
    If this object is subsequently set to an empty
    (zero-length)string , the MTA MUST delete the NCS SF
    if it exists. Setting this object to a different value
    does not cause the Upstream Service Flow to be
    re-created. The string MUST contain printable ASCII
    characters. The length of the string does not include a
    terminating zero. The MTA MUST append a terminating zero
    when the MTA creates the service flow. "
 ::= { pktcSigDevConfigObjects 13 }

```

```

pktcSigServiceClassNameDS    OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..15))
    MAX-ACCESS  read-write
    STATUS      obsolete
    DESCRIPTION
        "This object contains a string indicating the Service
        Class Name to create a Downstream Service Flow for NCS.
        If the object has an empty string value then the
        NCS SF is not created and the best effort primary SF is
        used for downstream NCS data. The creation of the NCS SF
        occurs before Voice Communication Service is activated on
        the device. If this object is set to a non-empty (non-zero
        length) string, the MTA MUST create the NCS SF if it does
        not currently exist and the pktcSigServiceClassNameMask
        object has a non-zero value. If this object is
        subsequently set to an empty (zero-length) string, the MTA
        MUST delete the NCS SF if it exists. Setting this object
        to a different value does not cause the Downstream Service
        Flow to be re-created. The string MUST contain printable
        ASCII characters. The length of the string does not include
        a terminating zero. The MTA MUST append a terminating
        zero when the MTA creates the service flow. "
    ::= { pktcSigDevConfigObjects 14 }

pktcSigServiceClassNameMask  OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-write
    STATUS      obsolete
    DESCRIPTION
        "This object contains a value for the Call Signalling
        Network Mask. The value is used as the NCS Call Signalling
        classifier mask. The object is used to delete the NCS SF
        when set to zero. When the object is set to a non-zero
        value by the SNMP Manager, the NCS SF are to be created."
    DEFVAL { 0 }
    ::= { pktcSigDevConfigObjects 15 }

pktcSigNcsServiceFlowState  OBJECT-TYPE
    SYNTAX      INTEGER {
        notactive  (1),
        active    (2),
        error      (3)
    }
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
        "This object contains a status value of the Call Signalling
        Service Flow.
        - 'notactive' indicates that the NCS SF is not being used,
        and has not tried to be created,
        - 'active' indicates that the NCS SF is in use,
        - 'error' indicates that the NCS SF creation resulted in
        an error and the best effort channel is used for NCS
        Signalling."
    ::= { pktcSigDevConfigObjects 16 }

pktcSigDevRlCadence         OBJECT-TYPE
    SYNTAX      PkctRingCadence
    MAX-ACCESS  read-write
    STATUS      current

```



```

        the specific signalling tables such as
        pktcNcsEndPntConfigTable."
 ::= { pktcNcsEndPntConfigObjects 1 }

```

```

pktcNcsEndPntConfigEntry OBJECT-TYPE
    SYNTAX      PktcNcsEndPntConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Entries in pktcNcsEndPntConfigTable - Each entry
        describes what signalling type a particular endpoint uses."
    INDEX { ifIndex }
 ::= { pktcNcsEndPntConfigTable 1 }

```

```

PktcNcsEndPntConfigEntry ::= SEQUENCE {
    pktcNcsEndPntConfigCallAgentId      SnmpAdminString,
    pktcNcsEndPntConfigCallAgentUdpPort Integer32,
    pktcNcsEndPntConfigPartialDialTO    Integer32,
    pktcNcsEndPntConfigCriticalDialTO   Integer32,
    pktcNcsEndPntConfigBusyToneTO       Integer32,
    pktcNcsEndPntConfigDialToneTO       Integer32,
    pktcNcsEndPntConfigMessageWaitingTO Integer32,
    pktcNcsEndPntConfigOffHookWarnToneTO Integer32,
    pktcNcsEndPntConfigRingingTO        Integer32,
    pktcNcsEndPntConfigRingBackTO       Integer32,
    pktcNcsEndPntConfigReorderToneTO    Integer32,
    pktcNcsEndPntConfigStutterDialToneTO Integer32,
    pktcNcsEndPntConfigTSMMax           Integer32,
    pktcNcsEndPntConfigMax1             Integer32,
    pktcNcsEndPntConfigMax2             Integer32,
    pktcNcsEndPntConfigMax1QEnable      TruthValue,
    pktcNcsEndPntConfigMax2QEnable      TruthValue,
    pktcNcsEndPntConfigMWD              Integer32,
    pktcNcsEndPntConfigTdinit           Integer32,
    pktcNcsEndPntConfigTdmin            Integer32,
    pktcNcsEndPntConfigTdmax            Integer32,
    pktcNcsEndPntConfigRtoMax           Integer32,
    pktcNcsEndPntConfigRtoInit          Integer32,
    pktcNcsEndPntConfigLongDurationKeepAlive Integer32,
    pktcNcsEndPntConfigThist            Integer32,
    pktcNcsEndPntConfigStatus           RowStatus,
    pktcNcsEndPntConfigCallWaitingMaxRep Integer32,
    pktcNcsEndPntConfigCallWaitingDelay Integer32,
    pktcNcsEndPntStatusCallIpAddress    IpAddress,
    pktcNcsEndPntStatusError            INTEGER
}

```

```

pktcNcsEndPntConfigCallAgentId OBJECT-TYPE
    SYNTAX      SnmpAdminString(SIZE (3..255))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object contains a string indicating the call agent
        name(e.g.: ca@abc.def.com). The call agent name
        after the character '@', MUST be a fully qualified
        domain name and MUST have a corresponding
        pktcMtaDevCmsFqdn entry in the pktcMtaDevCmsTable. For
        each particular end-point, the MTA MUST use the current
        value of this object to communicate with the corresponding
        CMS. The MTA MUST update this object with the value of the
        'Notified Entity' parameter of the NCS message. If the
        Notified Entity parameter does not contain a CallAgent
        port, the MTA MUST update this object with default value
        of 2727. Because of the high importance of this object to

```

the ability of the MTA to maintain reliable NCS communication with the CMS, it is highly recommended not to change this object's value through management station during normal operations."

::= { pktcNcsEndPntConfigEntry 1 }

pktcNcsEndPntConfigCallAgentUdpPort OBJECT-TYPE
SYNTAX Integer32 (1025..65535)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the current value of the User Datagram Protocol (UDP) receive port on which the call agent will receive NCS signalling from the endpoint. For each particular end-point, the MTA MUST use the current value of this object to communicate with the corresponding CMS. The MTA MUST update this object with the value of the 'Notified Entity' parameter of the NCS message. If the Notified Entity parameter does not contain a CallAgent port, the MTA MUST update this object with default value of 2727. Because of the high importance of this object to the ability of the MTA to maintain reliable NCS communication with the CMS, it is highly recommended not to change this object's value through management station during normal operations."
REFERENCE
"Refer to NCS specification"
DEFVAL { 2727 }
::= { pktcNcsEndPntConfigEntry 2 }

pktcNcsEndPntConfigPartialDialTO OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains maximum value of the partial dial time out."
REFERENCE
"Refer to PacketCable NCS specification"
DEFVAL { 16 }
::= { pktcNcsEndPntConfigEntry 3 }

pktcNcsEndPntConfigCriticalDialTO OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the maximum value of the critical dial time out."
REFERENCE
"Refer NCS specification"
DEFVAL { 4 }
::= { pktcNcsEndPntConfigEntry 4 }

pktcNcsEndPntConfigBusyToneTO OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This object contains the default timeout value for busy tone. The MTA MUST NOT update this object with the value provided in the NCS Message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

REFERENCE

"Refer to NCS specification"

DEFVAL { 30 }

::= { pktcNcsEndPntConfigEntry 5 }

pktcNcsEndPntConfigDialToneTO OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object contains the default timeout value for dial tone. The MTA MUST NOT update this object with the value provided in the NCS Message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

REFERENCE

"Refer to NCS specification "

DEFVAL { 16 }

::= { pktcNcsEndPntConfigEntry 6 }

pktcNcsEndPntConfigMessageWaitingTO OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object contains the default timeout value for message waiting indicator. The MTA MUST NOT update this object with the value provided in the NCS Message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

REFERENCE

"Refer to NCS specification"

DEFVAL { 16 }

::= { pktcNcsEndPntConfigEntry 7 }

pktcNcsEndPntConfigOffHookWarnToneTO OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object contains the default timeout value for the off hook Warning tone. The MTA MUST NOT update this object with the value provided in the NCS Message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

REFERENCE

"Refer to NCS specification"

```

DEFVAL { 0 }
 ::= { pktcNcsEndPntConfigEntry 8 }

pktcNcsEndPntConfigRingingTO      OBJECT-TYPE
SYNTAX      Integer32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object contains the default timeout value for
    ringing. The MTA MUST NOT update this object with
    the value provided in the NCS Message (if present).
    If the value of the object is modified by the
    SNMP Management Station, the MTA MUST use the new value
    as a default only for a new signal requested by the NCS
    message."
REFERENCE
    "Refer to NCS specification"
DEFVAL      { 180 }
 ::= { pktcNcsEndPntConfigEntry 9 }

pktcNcsEndPntConfigRingBackTO     OBJECT-TYPE
SYNTAX      Integer32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object contains the default timeout value for ring
    back. The MTA MUST NOT update this object with
    the value provided in the NCS Message (if present).
    If the value of the object is modified by the
    SNMP Management Station, the MTA MUST use the new value as
    a default only for a new signal requested by the NCS
    message."
REFERENCE
    "Refer to NCS specification"
DEFVAL      { 180 }
 ::= { pktcNcsEndPntConfigEntry 10 }

pktcNcsEndPntConfigReorderToneTO  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object contains the default timeout value for
    reorder tone. The MTA MUST NOT update this
    object with the value provided in the NCS Message (if
    present). If the value of the object is modified
    by the SNMP Management Station, the MTA MUST use the new
    value as a default only for a new signal requested by
    the NCS message."
REFERENCE
    "Refer to NCS specification"
DEFVAL      { 30 }
 ::= { pktcNcsEndPntConfigEntry 11 }

pktcNcsEndPntConfigStutterDialToneTO  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current

```

DESCRIPTION
"This object contains the default timeout value for
stutter dial tone. The MTA MUST NOT update this
object with the value provided in the NCS Message (if
present). If the value of the object is modified
by the SNMP Management Station, the MTA MUST use the new
value as a default only for a new signal requested by the
NCS message."

REFERENCE
"Refer to NCS specification"
DEFVAL { 16 }
::= { pktcNcsEndPntConfigEntry 12 }

pktcNcsEndPntConfigTSMMax OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the max time in seconds since the
sending of the initial datagram."
REFERENCE
"Refer to NCS specification"
DEFVAL { 20 }
::= { pktcNcsEndPntConfigEntry 13 }

pktcNcsEndPntConfigMax1 OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the suspicious error threshold
for signalling messages."
REFERENCE
"Refer to NCS specification"
DEFVAL { 5 }
::= { pktcNcsEndPntConfigEntry 14 }

pktcNcsEndPntConfigMax2 OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the disconnect error
threshold for signalling messages."
REFERENCE
"Refer to NCS specification"
DEFVAL { 7 }
::= { pktcNcsEndPntConfigEntry 15 }

pktcNcsEndPntConfigMax1QEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object enables/disables the Max1 Domain Name
Server (DNS) query operation when Max1 expires."
DEFVAL { true }
::= { pktcNcsEndPntConfigEntry 16 }

pktcNcsEndPntConfigMax2QEnable OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION
"This object enables/disables the Max2 DNS query
operation when Max2 expires."
DEFVAL { true }
::= { pktcNcsEndPntConfigEntry 17 }

pktcNcsEndPntConfigMWD OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Maximum Waiting Delay (MWD) contains the maximum
number of seconds a MTA waits after a restart."
REFERENCE
"Refer to NCS specification"
DEFVAL { 600 }
::= { pktcNcsEndPntConfigEntry 18 }

pktcNcsEndPntConfigTdinit OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the initial number of seconds
a MTA waits after a disconnect."
REFERENCE
"Refer to NCS specification"
DEFVAL { 15 }
::= { pktcNcsEndPntConfigEntry 19 }

pktcNcsEndPntConfigTadmin OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the minimum number of seconds a
MTA waits after a disconnect."
REFERENCE
"Refer to NCS specification"
DEFVAL { 15 }
::= { pktcNcsEndPntConfigEntry 20 }

pktcNcsEndPntConfigTdmax OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"This object contains the maximum number of seconds
a MTA waits after a disconnect."
REFERENCE
"Refer to NCS specification"
DEFVAL { 600 }
::= { pktcNcsEndPntConfigEntry 21 }

pktcNcsEndPntConfigRtoMax OBJECT-TYPE
SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION
 "This object contains the maximum number of seconds
 for the retransmission timer."
 REFERENCE
 "Refer to NCS specification"
 DEFVAL { 4 }
 ::= { pktcNcsEndPntConfigEntry 22 }

pktcNcsEndPntConfigRtoInit OBJECT-TYPE
 SYNTAX Integer32
 UNITS "milliseconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "This object contains the initial number of seconds
 for the retransmission timer."
 REFERENCE
 "Refer to NCS specification"
 DEFVAL { 200 }
 ::= { pktcNcsEndPntConfigEntry 23 }

pktcNcsEndPntConfigLongDurationKeepAlive OBJECT-TYPE
 SYNTAX Integer32
 UNITS "minutes"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "Specifies a timeout value in minutes for sending
 long duration call notification message."
 REFERENCE
 "Refer to NCS specification"
 DEFVAL { 60 }
 ::= { pktcNcsEndPntConfigEntry 24 }

pktcNcsEndPntConfigThist OBJECT-TYPE
 SYNTAX Integer32
 UNITS "seconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "Timeout period in seconds before no response is
 declared."
 REFERENCE
 "Refer to NCS specification"
 DEFVAL { 30 }
 ::= { pktcNcsEndPntConfigEntry 25 }

pktcNcsEndPntConfigStatus OBJECT-TYPE
 SYNTAX RowStatus
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "This object contains the Row Status associated with
 the pktcNcsEndPntConfigTable."
 ::= { pktcNcsEndPntConfigEntry 26 }

pktcNcsEndPntConfigCallWaitingMaxRep OBJECT-TYPE
 SYNTAX Integer32 (0..10)
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "This object contains the default value of the maximum
 number of repetitions of the call waiting tone that the

MTA will play from a single CMS request. The MTA MUST NOT update this object with the information provided in the NCS Message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

DEFVAL { 1 }
 ::= { pktcNcsEndPntConfigEntry 27 }

pktcNcsEndPntConfigCallWaitingDelay OBJECT-TYPE

SYNTAX Integer32 (1..100)

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object contains the delay between repetitions of the call waiting tone that the MTA will play from a single CMS request."

DEFVAL { 10 }

::= { pktcNcsEndPntConfigEntry 28 }

pktcNcsEndPntStatusCallIpAddress OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains the IP address of the CMS currently being used for this endpoint. This IP address is used to create the appropriate security association."

::= { pktcNcsEndPntConfigEntry 29 }

pktcNcsEndPntStatusError OBJECT-TYPE

SYNTAX INTEGER {
 operational (1),
 noSecurityAssociation (2),
 disconnected (3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains the error status for this interface. The operational state indicates that all operations necessary to put the line in service have occurred and CMS has acknowledged the RSIP message successfully.

If 'pktcMtaDevCmsIpsecCtrl' is enabled for the associated Call Agent, the noSecurityAssociation status indicates that no Security Association (SA) yet exists for this endpoint. Otherwise, the state is unused.

The disconnected status indicates one of the following two:

1. If 'pktcMtaDevCmsIpsecCtrl' is disabled then no security association is involved with this endpoint: the NCS signalling Software is in process of establishing the NCS signalling Link via an RSIP exchange.

2. Otherwise, pktcMtaDevCmsIpsecCtrl is enabled, the security Association has been established and the NCS signalling Software is in process of establishing the NCS signalling Link via an RSIP exchange."

::= { pktcNcsEndPntConfigEntry 30 }

--

-- notification group is for future extension.

--

```

pktcSigNotificationPrefix OBJECT IDENTIFIER ::= { pktcSigMib 2 }
pktcSigNotification OBJECT IDENTIFIER ::= {
  pktcSigNotificationPrefix 0 }
pktcSigConformance OBJECT IDENTIFIER ::= { pktcSigMib 3 }
pktcSigCompliances OBJECT IDENTIFIER ::= { pktcSigConformance 1 }
pktcSigGroups OBJECT IDENTIFIER ::= { pktcSigConformance 2 }

-- compliance statements

pktcSigBasicCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for devices that implement Signalling
    on the MTA."

MODULE -- pktcSigMib

-- unconditionally mandatory groups

MANDATORY-GROUPS {
  pktcSigGroup
}
GROUP pktcNcsGroup
DESCRIPTION
  "This group is mandatory for any MTA implementing NCS
  signalling"
::={ pktcSigCompliances 1 }

-- units of conformance

pktcSigGroup OBJECT-GROUP
  OBJECTS {
    pktcSigDevCodecType,
    pktcSigDevCodecMax,
    pktcSigDevEchoCancellation,
    pktcSigDevSilenceSuppression,
    pktcSigDevConnectionMode,
    pktcSigDevR0Cadence,
    pktcSigDevR6Cadence,
    pktcSigDevR7Cadence,
    pktcSigDefCallSigTos,
    pktcSigDefMediaStreamTos,
    pktcSigTosFormatSelector,
    pktcSigSignallingType,
    pktcSigSignallingVersion,
    pktcSigSignallingVendorExtension,
    pktcSigEndPntCapabilityIndex,
    pktcSigDefNcsReceiveUdpPort,
    pktcSigDevR1Cadence,
    pktcSigDevR2Cadence,
    pktcSigDevR3Cadence,
    pktcSigDevR4Cadence,
    pktcSigDevR5Cadence,
    pktcSigDevRgCadence,
    pktcSigDevRsCadence,
    pktcSigDevRtCadence
  }
  STATUS current
  DESCRIPTION
    "Group of objects for the common portion of the
    PacketCable Signalling MIB."
  ::= { pktcSigGroups 1 }

```

```

pktcNcsGroup OBJECT-GROUP
  OBJECTS {
    pktcNcsEndPntConfigCallAgentId,
    pktcNcsEndPntConfigCallAgentUdpPort,
    pktcNcsEndPntConfigPartialDialTO,
    pktcNcsEndPntConfigCriticalDialTO,
    pktcNcsEndPntConfigBusyToneTO,
    pktcNcsEndPntConfigDialToneTO,
    pktcNcsEndPntConfigMessageWaitingTO,
    pktcNcsEndPntConfigOffHookWarnToneTO,
    pktcNcsEndPntConfigRingingTO,
    pktcNcsEndPntConfigRingBackTO,
    pktcNcsEndPntConfigReorderToneTO,
    pktcNcsEndPntConfigStutterDialToneTO,
    pktcNcsEndPntConfigTSMMax,
    pktcNcsEndPntConfigMax1,
    pktcNcsEndPntConfigMax2,
    pktcNcsEndPntConfigMax1QEnable,
    pktcNcsEndPntConfigMax2QEnable,
    pktcNcsEndPntConfigMWD,
    pktcNcsEndPntConfigTdinit,
    pktcNcsEndPntConfigTdmin,
    pktcNcsEndPntConfigTdmax,
    pktcNcsEndPntConfigRtoMax,
    pktcNcsEndPntConfigRtoInit,
    pktcNcsEndPntConfigLongDurationKeepAlive,
    pktcNcsEndPntConfigThist,
    pktcNcsEndPntConfigStatus,
    pktcNcsEndPntConfigCallWaitingMaxRep,
    pktcNcsEndPntConfigCallWaitingDelay,
    pktcNcsEndPntStatusCallIpAddress,
    pktcNcsEndPntStatusError
  }
  STATUS current
  DESCRIPTION
    "Group of objects for the NCS portion of the
    PacketCable Signalling MIB. This is mandatory for
    NCS signalling."
    ::= { pktcSigGroups 2 }

pktcSigObsoleteGroup OBJECT-GROUP
  OBJECTS {
    pktcSigServiceClassNameUS,
    pktcSigServiceClassNameDS,
    pktcSigServiceClassNameMask,
    pktcSigNcsServiceFlowState
  }
  STATUS obsolete
  DESCRIPTION
    " Collection of obsolete objects for PacketCable
    Signalling MIB."
    ::= { pktcSigGroups 3 }
END

```

Annex D

Management Event MIB

The Management Event MIB MUST be implemented as defined below.

```
PKTC-EVENT-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    NOTIFICATION-TYPE,
    BITS
        FROM SNMPv2-SMI
    DateAndTime
        FROM SNMPv2-TC
    clabProjPacketCable
        FROM CLAB-DEF-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    OBJECT-GROUP,
    MODULE-COMPLIANCE,
    NOTIFICATION-GROUP
        FROM SNMPv2-CONF
    ifPhysAddress
        FROM IF-MIB
    InetAddressType,
    InetAddress,
    InetPortNumber
        FROM INET-ADDRESS-MIB ;

pktcEventMib MODULE-IDENTITY
    LAST-UPDATED "200501280000Z -- 01/28/2005"
    ORGANIZATION "Cable Television Laboratories, Inc"
    CONTACT-INFO
        "Sumanth Channabasappa
        Postal: Cable Television Laboratories, Inc.
        858 Coal Creek Circle
        Louisville, Colorado 80027
        U.S.A.
        Phone: +1 303-661-9100
        Fax: +1 303-661-9199
        E-mail: mibs@cablelabs.com"

    DESCRIPTION
        "This MIB module supplies the basic management objects
        for event reporting

        Acknowledgements:
            Eugene Nechamkin - Broadcom Corp
            John Berg - CableLabs, Inc.
            Kevin Marez - Motorola, Inc.
            Satish Kumar - Texas Instruments
            Venkatesh Sunkad - CableLabs, Inc."
    ::= { clabProjPacketCable 3 }

--
--
pktcDevEventControl OBJECT IDENTIFIER ::= { pktcEventMib 1 }
pktcDevEventThrottle OBJECT IDENTIFIER ::= { pktcEventMib 2 }
pktcDevEventStatus OBJECT IDENTIFIER ::= { pktcEventMib 3 }
pktcDevEventDescr OBJECT IDENTIFIER ::= { pktcEventMib 4 }
pktcDevEventLog OBJECT IDENTIFIER ::= { pktcEventMib 5 }
pktcDevEvNotification OBJECT IDENTIFIER ::= { pktcEventMib 6 }
--
```

```

---
---  Event Reporting control objects
---
pktcDevEvControl  OBJECT-TYPE
    SYNTAX          BITS {
                    resetEventLogTable(0),
                    resetEventDescrTable(1)
                    }    MAX-ACCESS  read-write
    STATUS          current
    DESCRIPTION
        "This MIB object defines the actions related to the event
        log configuration.

        The MTA MUST take the appropriate action whenever
        a bit is set to a value of '1'.

        Setting the resetEventLogTable(0) bit to
        a value of '1' clears the entire event log
        (Deletes all entries in pktcDevEventLogTable).

        Setting resetEventDescrTable(1) to a value of '1'
        resets the pktcDevEventDescrTable to the
        factory default values.

        Setting a control bit to a value of '0' MUST not result in
        any action.

        Reading this MIB object MUST always return '00'."
 ::= { pktcDevEventControl 1 }

pktcDevEvSyslogAddressType  OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This MIB Object defines the address type of the
        Syslog server.
        PacketCable devices implementing this MIB MUST
        support an InetAddressType of ipv4(1).
        PacketCable devices MAY optionally implement other
        address types.

        If an unsupported InetAddressType is used to set
        this object, the PacketCable device MUST reject it
        and report an SNMP error stating 'wrong value'.

        If an SNMP SET results in a type that does not match
        the value contained in the MIB Object
        pktcDevEvSyslogAddress, the PacketCable device MUST
        reject the SNMP SET with an 'inconsistent value'
        error."
 ::= { pktcDevEventControl 2 }

pktcDevEvSyslogAddress  OBJECT-TYPE
    SYNTAX          InetAddress
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This MIB Object contains the IP address of the
        Syslog server. If this is set to either 0.0.0.0 or
        255.255.255.255 the device MUST inhibit syslog
        transmission.
        The use of FQDNs is syntactically allowed, but
        discouraged since a failure to resolve them in a

```

timely manner may leave the device without access to the Syslog daemon during critical network events. The type of address this object represents is defined by the MIB Object pktcDevEvSyslogAddressType.

If an SNMP SET results in a type that does not match that indicated by the MIB Object pktcDevEvSyslogAddressType, the PacketCable device MUST reject the SNMP SET with an 'inconsistent value' error."

```
::= { pktcDevEventControl 3 }
```

```
pktcDevEvSyslogUdpPort OBJECT-TYPE
```

```
SYNTAX      InetPortNumber
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

"This MIB Object contains the UDP Port Number of the Syslog Server. The PacketCable device must send the Syslog messages to this port on the Syslog Server."

```
DEFVAL { 514 }
```

```
::= { pktcDevEventControl 4 }
```

```
--
```

```
-- Event throttling control
```

```
--
```

```
pktcDevEvThrottleAdminStatus OBJECT-TYPE
```

```
SYNTAX      INTEGER {  
            unconstrained(1),  
            maintainBelowThreshold(2),  
            stopAtThreshold(3),  
            inhibited(4)  
            }
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

"This MIB Object controls the throttling of the transmitted messages upon generation of an event (SNMP/Syslog).

A value of unconstrained(1) causes event messages to be transmitted without regard to the threshold settings.

A value of maintainBelowThreshold(2) causes event messages to be suppressed if the number of transmissions would otherwise exceed the threshold.

A value of stopAtThreshold(3) causes event message transmission to cease at the threshold, and not resume until directed to do so.

A value of inhibited(4) causes all event message transmission to be suppressed.

An event causing both an SNMP and a Syslog message is still treated as a single event.

Writing to this object resets the thresholding state.

Refer to MIB Objects pktcDevEvThrottleThreshold and pktcDevEvThrottleInterval for information on throttling."


```
DEFVAL { unconstrained }
 ::= { pktcDevEventThrottle 1 }
```

pktcDevEvThrottleThreshold OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB Object contains the number of events per
pktcDevEvThrottleInterval to be transmitted before
throttling.

An event causing both a SNMP and a syslog message is
still treated as a single event."

DEFVAL { 2 }

::= { pktcDevEventThrottle 2 }

pktcDevEvThrottleInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB Object contains the interval over which
the throttle threshold applies." DEFVAL { 1 }

::= { pktcDevEventThrottle 3 }

-- Status Reporting

pktcDevEvTransmissionStatus OBJECT-TYPE

SYNTAX BITS {
syslogThrottled(0),
snmpThrottled(1),
validSyslogServerAbsent(2),
validSnmpManagerAbsent(3),
syslogTransmitError(4),
snmpTransmitError(5)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object reflects the status of the event
transmission.

If a bit corresponding to a state is set to a value
of:

'1', it indicates that the state is true

'0', it indicates that the state is false

'Event throttling' is based on thresholds and the current
setting of pktcDevEvThrottleAdminStatus.

'Server/Manager' indicators must be based on the
availability of valid Syslog server/SNMP managers.

'Transmit Errors' must only be used in cases where the
PacketCable Device can identify unavailable servers."

```

 ::= { pktcDevEventStatus 1 }

---
-- Event Descriptions
---

pktcDevEventDescrTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcDevEventDescrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB table contains all the possible events
         that can be generated by the device. This includes
         both PacketCable defined and vendor-specific events."
    ::= { pktcDevEventDescr 1 }

pktcDevEventDescrEntry OBJECT-TYPE
    SYNTAX      PktcDevEventDescrEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in this table is created for each
         event the PacketCable Device implementing this
         MIB is capable of reporting."
    INDEX { pktcDevEventDescrId, pktcDevEventDescrEnterprise }
    ::= { pktcDevEventDescrTable 1 }

PktcDevEventDescrEntry ::= SEQUENCE {
    pktcDevEventDescrId      Unsigned32,
    pktcDevEventDescrEnterprise Unsigned32,
    pktcDevEventDescrFacility INTEGER,
    pktcDevEventDescrLevel  INTEGER,
    pktcDevEventDescrReporting BITS,
    pktcDevEventDescrText   SnmpAdminString
}

pktcDevEventDescrId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB Object contains the event identifier for the
         specific event to which the priority and display
         strings belong.
         The event identifier can either be PacketCable defined
         or vendor-specific."
    ::= { pktcDevEventDescrEntry 1 }

pktcDevEventDescrEnterprise OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This MIB Object provides the IANA enterprise number of
         the Organization defining the event. Thus, all PacketCable
         defined events will contain the CableLabs IANA enterprise
         number and for vendor-specific events it will contain
         the IANA enterprise number of the defining organization."
    ::= { pktcDevEventDescrEntry 2 }

```

pktcDevEventDescrFacility OBJECT-TYPE

```
SYNTAX      INTEGER {
            kernel(0),
            user(1),
            mail(2),
            daemon(3),
            auth(4),
            syslog(5),
            lpr(6),
            news(7),
            uucp(8),
            cron(9),
            authPriv(10),
            ftp(11),
            ntp(12),
            security(13),
            console(14),
            clockDaemon(15),
            local0(16),
            local1(17),
            local2(18),
            local3(19),
            local4(20),
            local5(21),
            local6(22),
            local7(23)
        }
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains the facility for the event. For PacketCable events this MUST be set to local0(16)."

::= { pktcDevEventDescrEntry 3 }

pktcDevEventDescrLevel OBJECT-TYPE

```
SYNTAX      INTEGER {
            emergency(0),
            alert(1),
            critical(2),
            error(3),
            warning(4),
            notice(5),
            info(6),
            debug(7)
        }
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB Object contains the priority level that is controlled by this entry. The levels are described as:

emergency(0) - A condition that makes the system unusable.
alert(1) - A service-affecting condition for which immediate action must be taken.
critical(2) - A service-affecting critical condition.
error(3) - An error condition.
warning(4) - A warning condition.
notice(5) - A normal but significant condition.
info(6) - An informational message.
debug(7) - A debug message."

```
::= { pktcDevEventDescrEntry 4 }
```

```
pktcDevEventDescrReporting OBJECT-TYPE
```

```
SYNTAX      BITS {  
             local(0),  
             syslog(1),  
             snmpTrap(2),  
             snmpInform(3)  
            }
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This MIB Object defines the action to be taken on  
occurrence of this event class.
```

```
Setting a bit to a value of '1' indicates that the  
corresponding action will be taken upon occurrence of  
this event, provided the required parameters are present.  
(e.g.: Syslog Server for Syslog messages, SNMP targets for  
SNMP traps and SNMP INFORMs etc). If none of the bits  
are set then no action is taken upon occurrence of the  
event.
```

```
The default value of this MIB Object is dependent on the  
value of the MIB Object 'pktcDevEventDescrLevel', for the  
corresponding event.
```

```
For the following values of 'pktcDevEventDescrLevel':  
    emergency(0), alert(1), critical(2) and error(3),  
the PacketCable device MUST set the bits for local(0),  
syslog(1) and snmpInform(3) to a value of '1' and the rest  
to a value of '0'.
```

```
For all the remaining values of 'pktcDevEventDescrLevel',  
the PacketCable device MUST set the bits for local(0) and  
syslog(1) to a value of '1' and the rest to a value of  
'0'."
```

```
::= { pktcDevEventDescrEntry 5 }
```

```
pktcDevEventDescrText OBJECT-TYPE
```

```
SYNTAX      SnmpAdminString(SIZE (0..127))
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This MIB Object contains event display  
string providing a human-readable description of the  
event."
```

```
::= { pktcDevEventDescrEntry 6 }
```

```
---
```

```
-- Events generated
```

```
---
```

```
pktcDevEventLogTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF PktcDevEventLogEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This MIB table contains a log of the events  
generated by the PacketCable device.  
A description of all the events that can be  
generated by the device can be obtained from the  
MIB table 'pktcDevEventDescrTable'."
```

```
::= { pktcDevEventLog 1 }
```

```

pktcDevEventLogEntry OBJECT-TYPE
    SYNTAX      PktcDevEventLogEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry in this table describes an event that
        has occurred, indexed in the chronological order of
        generation. The details of the event are borrowed
        from the parameters associated with the corresponding
        event entry in 'pktcDevEventDescrTable', at the
        time of the event generation.
        While all entries created as such can be cleared using
        the MIB Object pktcDevEvControl, the Event entries
        themselves cannot be individually deleted."

```

```

INDEX { pktcDevEvLogIndex }
 ::= { pktcDevEventLogTable 1 }

```

```

PktcDevEventLogEntry ::= SEQUENCE {
    pktcDevEvLogIndex      Unsigned32,
    pktcDevEvLogTime       DateAndTime,
    pktcDevEvLogEnterprise Unsigned32,
    pktcDevEvLogId         Unsigned32,
    pktcDevEvLogText       SnmpAdminString,
    pktcDevEvLogEndpointName SnmpAdminString,
    pktcDevEvLogType       BITS,
    pktcDevEvLogTargetInfo SnmpAdminString,
    pktcDevEvLogCorrelationId Unsigned32,
    pktcDevEvLogAdditionalInfo SnmpAdminString
}

```

```

pktcDevEvLogIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This MIB Object provides relative ordering of the
        objects in the event log.
        This object will always increase except when
        (a) the log is reset via pktcDevEvControl,
        (b) the device reboots and does not implement non-volatile
        storage for this log,
        (c) it reaches the value 2^31.
        The next entry for all the above cases is 0.
        This also serves as an indicator of event sequence."
    ::= { pktcDevEventLogEntry 1 }

```

```

pktcDevEvLogTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This MIB Object provides a human-readable description
        of the time at which the event occurred."
    ::= { pktcDevEventLogEntry 2 }

```

```

pktcDevEvLogEnterprise OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current

```

DESCRIPTION

"This MIB Object provides the IANA enterprise number of the Organization defining the event. Thus, all PacketCable defined events will contain the CableLabs IANA enterprise number and for vendor-specific events it will contain the IANA enterprise number of the defining organization."

::= { pktcDevEventLogEntry 3 }

pktcDevEvLogId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains the event identifier for the specific event to which the priority and display strings belong. The event identifier can either be PacketCable defined or vendor-specific."

::= { pktcDevEventLogEntry 4 }

pktcDevEvLogText OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains the contents of pktcDevEventDescrText, corresponding to the event, at the moment of generation."

::= { pktcDevEventLogEntry 5 }

pktcDevEvLogEndpointName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object provides the endpoint identifier followed by the PacketCable MTA's Fully Qualified Domain Name (FQDN) and the IP Address (IP) of the PacketCable MTA device.

This will be denoted as follows:

aaln/n:<FQDN>/<IP>, where 'n' is the Endpoint number.

or

<FQDN>/<IP> if it is not specific to an endpoint."

::= { pktcDevEventLogEntry 6 }

pktcDevEvLogType OBJECT-TYPE

SYNTAX BITS {
local(0),
syslog(1),
trap(2),
inform(3)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains the kind of actions taken by the PacketCable device when the event under consideration occurred.

A bit with a value of 1 indicates the corresponding action was taken. Setting it to a value of 0 indicates that the corresponding action was not taken.

An event may trigger one or more actions (e.g.: Syslog and SNMP) or may remain as a local event since transmissions could be disabled or inhibited as defined by the Throttle MIB Objects."

```
::= { pktcDevEventLogEntry 7 }
```

pktcDevEvLogTargetInfo OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains a comma separated list of the actions taken, along with the target IP address for the generated event.

The syntax is as:

<action-1/IP:port>,<action-2/IP:port>,<action-3/IP:port>

Where <action-n/IP> is to be denoted as follows:

For Syslog events:

syslog/<IP address of the Syslog Server:port>

For SNMP traps:

snmpTrap/<IP address of the SNMP Server:port>

For SNMP INFORMS:

snmpInform/<IP address of the SNMP Server:port>

If there are multiple targets for the same type (SNMP Traps sent to multiple IP addresses) or if there are multiple messages sent to the same IP (Syslog and SNMP sent to the same IP address) they need to be reported individually."

```
::= { pktcDevEventLogEntry 8 }
```

pktcDevEvLogCorrelationId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This MIB Object contains the correlation ID generated by the MTA as per section 5.4.5 of [7] that was being used by the MTA when the event was generated."

```
::= { pktcDevEventLogEntry 9 }
```

pktcDevEvLogAdditionalInfo OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB Object contains additional, useful information in relation to the corresponding event that a PacketCable device might wish to report (for example: parameterized data or debugging information). The format is vendor-specific.

However, the PacketCable device is not required to implement this functionality."

```
::= { pktcDevEventLogEntry 10 }
```

```

---
-- Notifications
---

pktcDevEvNotificationIndex OBJECT IDENTIFIER ::=
    { pktcDevEvNotification 0 }

pktcDevEvInform NOTIFICATION-TYPE
    OBJECTS {pktcDevEvLogIndex, pktcDevEvLogTime,
pktcDevEvLogEnterprise, pktcDevEvLogId,
pktcDevEvLogEndpointName, pktcDevEvLogCorrelationId, ifPhysAddress}
    STATUS current
    DESCRIPTION
        "This Notification MIB Objects contains the Inform
        contents for event reporting "
    ::= { pktcDevEvNotificationIndex 1 }

pktcDevEvTrap NOTIFICATION-TYPE
    OBJECTS {pktcDevEvLogIndex, pktcDevEvLogTime,
pktcDevEvLogEnterprise, pktcDevEvLogId,
pktcDevEvLogEndpointName, pktcDevEvLogCorrelationId, ifPhysAddress}
    STATUS current
    DESCRIPTION
        "This Notification MIB Objects contains the Trap contents
        for event reporting "
    ::= { pktcDevEvNotificationIndex 2 }

---
-- Conformance/Compliance
---

pktcEventConformance OBJECT IDENTIFIER ::= { pktcEventMib 7 }
pktcEventCompliances OBJECT IDENTIFIER ::= { pktcEventConformance 1 }
pktcEventGroups OBJECT IDENTIFIER ::= { pktcEventConformance 2 }

pktcEventBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for devices that implement
        Event reporting feature."
    MODULE --pktcEventMib

MANDATORY-GROUPS {
    pktcEventGroup,
    pktcEventNotificationGroup
}
-- units of conformance
::= { pktcEventCompliances 3 }

pktcEventGroup OBJECT-GROUP
    OBJECTS {
        pktcDevEvControl,
        pktcDevEvSyslogAddressType,
        pktcDevEvSyslogAddress,
        pktcDevEvSyslogUdpPort,
        pktcDevEvThrottleAdminStatus,
        pktcDevEvThrottleThreshold,
        pktcDevEvThrottleInterval,
        pktcDevEvTransmissionStatus,
        pktcDevEventDescrEnterprise,
        pktcDevEventDescrFacility,
        pktcDevEventDescrLevel,
    }

```



```

    pktcDevEventDescrReporting,
    pktcDevEventDescrText,
    pktcDevEvLogIndex,
    pktcDevEvLogTime,
    pktcDevEvLogEnterprise,
    pktcDevEvLogId,
    pktcDevEvLogText,
    pktcDevEvLogEndpointName,
    pktcDevEvLogType,
    pktcDevEvLogTargetInfo,
    pktcDevEvLogCorrelationId,
    pktcDevEvLogAdditionalInfo
}

STATUS          current
DESCRIPTION
    "Group of MIB objects for PacketCable Management Event
    MIB."
 ::= { pktcEventGroups 1 }

pktcEventNotificationGroup NOTIFICATION-GROUP
NOTIFICATIONS { pktcDevEvInform, pktcDevEvTrap }
STATUS          current
DESCRIPTION
    "Group of MIB objects for notifications related to
    change in status of the MTA Device."
 ::= { pktcEventGroups 2 }
END

```

Annex E

Extension MTA MIB

The Extension MTA MIB MUST be implemented as defined below.

```

PKTC-EN-MTA-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE          FROM SNMPv2-SMI
    OBJECT-GROUP, MODULE-COMPLIANCE     FROM SNMPv2-CONF
    pktcEnhancements                    FROM CLAB-DEF-MIB;

pktcEnMtaMib MODULE-IDENTITY
    LAST-UPDATED      "200501280000Z - January 28, 2005"
    ORGANIZATION      "Cable Television Laboratories, Inc"

    CONTACT-INFO
        "Sumanth Channabasappa
        Postal: Cable Television Laboratories, Inc.
        858 Coal Creek Circle
        Louisville, Colorado 80027-9750
        U.S.A.
        Phone: +1 303-661-9100
        Fax: +1 303-661-9199
        E-mail: mibs@cablelabs.com"

```

DESCRIPTION

"This MIB module enhances the basic management objects defined for the PacketCable MTA Device by the MIB group pktcMtaMib.

Acknowledgements:

Rodney Osborne - Arris Interactive
Eugene Nechamkin - BroadCom Corporation
Satish Kumar - Texas Instruments
Jean-Francois Mule - CableLabs
Venkatesh Sunkad - CableLabs

Copyright 1999-2005 Cable Television Laboratories, Inc.
All rights reserved."

REVISION "200501280000Z"

DESCRIPTION

"This revision is being published as part of the PacketCable MTA MIBs enhancements for PacketCable 1.5."

::= { pktcEnhancements 1 }

--

-- PacketCable Enhanced MTA MIB Objects

--

pktcEnMtaMibObjects OBJECT IDENTIFIER ::= { pktcEnMtaMib 1 }
pktcEnMtaDevBase OBJECT IDENTIFIER ::= { pktcEnMtaMibObjects 1 }
pktcEnMtaDevServer OBJECT IDENTIFIER ::= { pktcEnMtaMibObjects 2 }
pktcEnMtaDevSecurity OBJECT IDENTIFIER ::= { pktcEnMtaMibObjects 3 }

--

-- Enhanced notification group.

--

pktcEnMtaNotificationPrefix OBJECT IDENTIFIER ::= { pktcEnMtaMib 2 }
pktcEnMtaNotification OBJECT IDENTIFIER ::= { pktcEnMtaNotificationPrefix
0 }
pktcEnMtaConformance OBJECT IDENTIFIER ::= { pktcEnMtaMib 3 }
pktcEnMtaCompliances OBJECT IDENTIFIER ::= { pktcEnMtaConformance 1 }
pktcEnMtaGroups OBJECT IDENTIFIER ::= { pktcEnMtaConformance 2 }

--

-- Enhancement MIB Objects

--

pktcEnMtaDevMltplGrantsPerInterval OBJECT-TYPE
SYNTAX INTEGER {
enablempifunctionality(1),
disablempifunctionality(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This object is used to control the Multiple grants functionality on a PacketCable MTA.
To indicate enabling of this functionality, a value of enablempifunctionality(1) is used.
To indicate disabling of this functionality, a value of disablempifunctionality(2) is used."

```

DEFVAL {disablempifunctionality}
 ::= { pktcEnMtaDevBase 1}

--
-- Compliance statements
--
pktcEnMtaBasicCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices that implement
        MTA feature."
    MODULE      --PKTC-EN-MTA-MIB

--
-- Mandatory groups
--
    MANDATORY-GROUPS {
        pktcEnMtaGroup
    }
 ::= { pktcEnMtaCompliances 3 }

pktcEnMtaGroup OBJECT-GROUP
    OBJECTS {
        pktcEnMtaDevMltplGrantsPerInterval
    }
    STATUS      current
    DESCRIPTION
        "Group of Enhanced objects for the PacketCable MTA MIB."
 ::= { pktcEnMtaGroups 1 }

END

```

Annex F

Signalling Extension MIB

The Extension Signalling MIB MUST be implemented as defined below.

```

PKTC-EN-SIG-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32, BITS FROM SNMPv2-SMI
    ifIndex FROM IF-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    TruthValue
        FROM SNMPv2-TC
    OBJECT-GROUP,
    MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    pktcEnhancements
        FROM CLAB-DEF-MIB
    pktcNcsEndPntConfigEntry
        FROM PKTC-SIG-MIB;

```

```

pktcEnSigMib  MODULE-IDENTITY
    LAST-UPDATED      "200528010000Z" -- January 28, 2005
    ORGANIZATION      "Cable Television Laboratories, Inc "
    CONTACT-INFO
        "Sumanth Channabasappa
        Postal: Cable Television Laboratories, Inc.
        858 Coal Creek Circle
        Louisville, Colorado 80027-9750
        U.S.A.
        Phone: +1 303-661-9100
        Fax: +1 303-661-9199
        E-mail: mibs@cablelabs.com"

```

```

DESCRIPTION
    "This MIB module enhances the basic management
    objects defined for PacketCable Signalling
    protocols by the MIB group pktcSigMib.

```

```

Acknowledgements:
Rodney Osborne - Arris Interactive
Eugene Nechamkin - Broadcom Corporation
Satish Kumar - Texas Instruments
Jean-Francois Mule - CableLabs

```

```

Copyright 1999-2004 Cable Television Laboratories, Inc.
All rights reserved."

```

```

REVISION "2005028010000Z"

```

```

DESCRIPTION
    "This revision is being published as part of the PacketCable
    Signalling MIBs enhancements for PacketCable 1.5."
 ::= { pktcEnhancements 2 }

```

```

--
-- Enhanced MIB Objects and Divisions.
--

```

```

pktcEnSigMibObjects      OBJECT IDENTIFIER
                          ::= { pktcEnSigMib 1 }
pktcEnSigDevConfigObjects OBJECT IDENTIFIER
                          ::= { pktcEnSigMibObjects 1 }
pktcEnNcsEndPntConfigObjects OBJECT IDENTIFIER
                          ::= { pktcEnSigMibObjects 2 }
pktcEnSigEndPntConfigObjects OBJECT IDENTIFIER
                          ::= { pktcEnSigMibObjects 3 }
pktcEnDcsEndPntConfigObjects OBJECT IDENTIFIER
                          ::= { pktcEnSigMibObjects 4 }

```

```

--
-- Enhanced Notification groups.
--

```

```

pktcEnSigNotificationPrefix OBJECT IDENTIFIER
                          ::= { pktcEnSigMib 2 }
pktcEnSigNotification      OBJECT IDENTIFIER
                          ::= { pktcEnSigNotificationPrefix 0 }
pktcEnSigConformance       OBJECT IDENTIFIER
                          ::= { pktcEnSigMib 3 }
pktcEnSigCompliances       OBJECT IDENTIFIER
                          ::= { pktcEnSigConformance 1 }

```

```

pktcEnSigGroups          OBJECT IDENTIFIER
                           ::= { pktcEnSigConformance 2 }

pktcEnNcsMinimumDtmfPloyout OBJECT-TYPE
    SYNTAX Unsigned32 (0 | 40..100)
    UNITS "milliseconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This object defines the minimum playout time for
        the DTMF digit when IETF RFC 2833 DTMF Relay is used
        for the egress gateway.
        If the value set via this pktcEnNcsMinimumDtmfPloyout
        object is different from that specified in RFC2833 packet,
        then the MTA MUST use the maximum of the two values.
        For example:
        If the RFC 2833 packet specifies 23 ms and if the object
        pktcEnNcsMinimumDtmfPloyout is set to 40 ms then
        the egress gateway must use a value of 40 ms.
        Similarly if the RFC 2833 packet specifies
        60 ms and if the object pktcEnNcsMinimumDtmfPloyout
        is set to 40 ms then the egress gateway must use a
        value of 60 ms."
    REFERENCE
        "PacketCable(tm) Codec Specification"
    DEFVAL {0}
    ::= { pktcEnSigDevConfigObjects 1}

--
-- The following table enhances the NCS End Point Config Table
-- (pktcNcsEndPntConfigTable) defined in pktSigMib.
--
--
pktcEnNcsEndPntConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PktcEnNcsEndPntConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table augments pktcNcsEndPntConfigTable."
    ::= { pktcEnNcsEndPntConfigObjects 1 }

pktcEnNcsEndPntConfigEntry OBJECT-TYPE
    SYNTAX PktcEnNcsEndPntConfigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An enhancement to pktcNcEndPntConfigTable - where each
        entry describes endpoint characteristics."
    AUGMENTS { pktcNcsEndPntConfigEntry }
    ::= { pktcEnNcsEndPntConfigTable 1 }

PktcEnNcsEndPntConfigEntry ::=
    SEQUENCE {
        pktcEnNcsEndPntQuarantineState INTEGER,
        pktcEnNcsEndPntHookState INTEGER,
        pktcEnNcsEndPntFaxDetection TruthValue,
        pktcEnNcsEndPntStatusReportCtrl INTEGER
    }

```

```

pktcEnNcsEndPntQuarantineState OBJECT-TYPE
    SYNTAX INTEGER {
        normal                (1),
        notification          (2),
        lockstep              (3),
        extendedlockstep      (4)
    }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object reflects the state of the Endpoint.
        When the endpoint is in notification, lockstep or
        values 'extended lockstep' states, the E-MTA MUST report
        the value of notification(2), lockstep(3) or extendedlockstep(4),
        respectively. Else, the endpoint MUST report a value
        of normal(1).
        'Extended Lockstep' is defined as the state when the
        E-MTA is in the lockstep state for longer than 2 minutes.
        For more description about the states refer
        to the PacketCable Network Based Call signalling
        specification."
    REFERENCE
        "PacketCable(tm) Network-Based Call Signalling Protocol
        Specification,"
 ::= { pktcEnNcsEndPntConfigEntry 1 }

```

```

pktcEnNcsEndPntHookState OBJECT-TYPE
    SYNTAX INTEGER {
        onHook                (1),
        onHookPlusNCSActivity (2),
        offHook               (3)
    }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "This object reflects the 'hook state' and 'NCS Activity'
        of an endpoint.
        'NCS Activity', by definition includes: an active
        timeout signal, active brief signal or existence of
        an NCS connection.
        The onHook(1) state indicates that the endpoint is
        'on hook' and the absence of 'NCS Activity' on that
        endpoint.
        The onHookPlusNCSActivity(2) indicates that the endpoint
        is 'on hook' and the presence of 'NCS Activity' on that
        endpoint.
        The offHook(3) state indicates that the endpoint is
        'off hook'."
    REFERENCE
        "PacketCable(tm) Network-Based Call Signalling Protocol
        Specification"
 ::= { pktcEnNcsEndPntConfigEntry 2 }

```

```

pktcEnNcsEndPntFaxDetection OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS    read-create
    STATUS        current

```

DESCRIPTION

"This MIB object is used to configure the distinctive fax calling tone (CNG) detection feature on an MTA endpoint with reference to the analog interface. When set to true, the MTA MUST enable the detection of CNG tones on the specific endpoint. When set to false, the MTA MUST disable the detection of CNG tones on the specific endpoint. If a connection already exists on the endpoint when this MIB Object is modified, then the setting needs to take effect on the next connection."

DEFVAL {false}
 ::= { pktcEnNcsEndPntConfigEntry 3}

pktcEnNcsEndPntStatusReportCtrl OBJECT-TYPE

SYNTAX INTEGER {
 unsupported (1),
 reportActualStatus (2),
 reportEndPointAsActive (3)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object is used to control the Endpoint Status Reporting, if the feature is supported by the MTA and is configurable. The term 'Endpoint Status Reporting' refers to any information that the MTA may provide to External Systems for use in a particular reporting mechanism (Ex: Home Alarm Systems). The definition of the External Systems and reporting mechanism are beyond the scope of this definition (In the example of Home Alarm Systems, this MIB Object will allow Management Stations to temporarily disable outage reporting on an EndPoint during planned downtime).

If supported, the MTA MUST:

- reflect the actual endpoint status when the value is set to 'reportActualStatus(2)'
- reflect the EndPoint status as being active when the value is set to 'reportEndPointAsActive(3)', irrespective of the actual status.

If unsupported, the MTA MUST set this value to 'unsupported(1)' and reject any attempt to set this MIB object using SNMP SET to any other value."

::= { pktcEnNcsEndPntConfigEntry 4}

pktcEnEndPntInfoTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcEnEndPntInfoTableEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table includes any additional information associated with PacketCable EndPoints. The number of entries in this table represents the number of available PacketCable EndPoints."

::= { pktcEnNcsEndPntConfigObjects 2 }

pktcEnEndPntInfoTableEntry OBJECT-TYPE

SYNTAX PktcEnEndPntInfoTableEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in this table MUST be created for each PacketCable EndPoint.
The index needs to be the corresponding index in the ifTable for the associated PacketCable EndPoint."

INDEX { ifIndex }
 ::= { pktcEnEndPntInfoTable 1 }

PktcEnEndPntInfoTableEntry ::=

SEQUENCE {
 pktcEnEndPntFgnPotSupport BITS,
 pktcEnEndPntFgnPotDescr SnmpAdminString,
 pktcEnEndPntClrFgnPotTsts BITS,
 pktcEnEndPntRunFgnPotTsts BITS,
 pktcEnEndPntFgnTestValidity BITS,
 pktcEnEndPntFgnTestResults BITS
}

pktcEnEndPntFgnPotSupport OBJECT-TYPE

SYNTAX BITS {
 fgnPotDetection (0),
 hazardousFgnPotDetection (1)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object indicates the capabilities of the MTA to detect various conditions related to the presence of foreign potential on an endpoint.
The MTA MUST set a value of '1' for each bit corresponding to a supported functionality and a value of '0' for each bit corresponding to an unsupported functionality."

::= { pktcEnEndPntInfoTableEntry 1 }

pktcEnEndPntFgnPotDescr OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object provides information related to the various tests for each detection mechanism supported by the MTA. While the actual contents are vendor-specific, the recommended format is:

[<Capability>:<Test References>:<Other Info>]...

Example:

<fgnPotDetection>:<test XYZ, Reference 'Document'>:<NA>;

<hazardousFngPotDetection>:<Test ABC, References>:<NA>

"

::= { pktcEnEndPntInfoTableEntry 2 }

pktcEnEndPntClrFgnPotTsts OBJECT-TYPE

SYNTAX BITS {
 clrFgnPotentialResults (0),
 clrHazardousPotResults (1)
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This MIB object is used to clear the current test results of supported conditions indicated by

'pktcEnEndPntFgnPotSupport'.
 Setting a bit to a value of '1' clears the corresponding results in the MIB Object 'pktcEnEndPntFgnTestResults' and the validity as indicated by the MIB object 'pktcEnEndPntFgnTestValidity' for the supported conditions only (i.e, the MTA MUST set the corresponding bits to a value of '0' in the indicated tables).
 If an SNMP SET attempts to set a bit corresponding to an unsupported condition to a value of '1', then the MTA MUST reject the entire SNMP SET and report an 'inconsistent value' error.
 For all unsupported scenarios, the corresponding bits MUST be set to a value of '0'.
 Whenever one or more tests are enabled by the MIB Object 'pktcEnEndPntRunFgnPotTests', the MTA MUST also reset the corresponding bits in this MIB Object to a value of '0'.
 ::= { pktcEnEndPntInfoTableEntry 3}

```

pktcEnEndPntRunFgnPotTsts      OBJECT-TYPE
    SYNTAX BITS {
        runFgnPotentialTsts (0),
        runHazardousPotTsts (1)
    }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This MIB object is used to initiate one or more test cases
        associated with a supported foreign potential detection.
        Thus, whenever one or more BITS corresponding to supported
        foreign scenario potential detection mechanisms are set to
        a value of '1', the MTA MUST enable those tests.
        Once the tests are executed, the MTA MUST:
        - set the corresponding bit to a value of '0'
        - update the corresponding BITS in the MIB Objects
        'pktcEnEndPntFgnTestValidity' and
        'pktcEnEndPntFgnTestResults'.
        If an SNMP SET attempts to set a bit corresponding to an
        unsupported condition to a value of '1', then the MTA MUST
        reject the entire SNMP SET and report an 'inconsistent
        value' error.
        Whenever a test is being run on an EndPoint the MTA MUST
        set the corresponding 'ifOperStatus' MIB Object to a
        value of 'testing(3)' for the whole duration of the test.
        When the test is completed, the MTA MUST set the
        ifOperStatus to the value corresponding to the current
        state of the line.

        Note: Whenever multiple tests are run, the ordering of the
        tests or the results is vendor dependent and need not
        necessarily follow the ordering of BITS in this MIB
        Object."
    ::= { pktcEnEndPntInfoTableEntry 4}

```

```

pktcEnEndPntFgnTestValidity  OBJECT-TYPE
    SYNTAX BITS {
        fgnPotTstValidity (0),
        hazardousPotTstValidity (1)
    }
    MAX-ACCESS      read-only
    STATUS          current

```

```

DESCRIPTION
    "This MIB object is used to indicate the validity of the
    corresponding test cases that were initiated using the MIB
    Object 'pktcEnEndPntRunFgnPotTests'.
    An MTA MUST:
    - return a value of '1' if the tests were run successfully
    and the results are valid.
    - return a value of '0' if a particular test was not
    initiated or if the tests could not be run successfully
    and hence the results are invalid.
    Note: The MTA MUST set all the BITS to '0' as soon as one
    or more test cases are initiated."
 ::= { pktcEnEndPntInfoTableEntry 5}

```

```

pktcEnEndPntFgnTestResults OBJECT-TYPE

```

```

    SYNTAX BITS {
        fgnPotentialResults (0),
        hazardousPotResults (1)
    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This MIB object is used to indicate the results of the
        corresponding test cases that were initiated using the MIB
        Object 'pktcEnEndPntRunFgnPotTests'.
        An MTA MUST:
        - set the corresponding bit to a value of '1' if the tests
        indicated the presence of a foreign potential as per the
        associated test case.
        - set the corresponding bit to a value of '0' if the tests
        indicated the absence of a foreign potential as per the
        associated test case.
        Note: The MTA MUST set all the BITS to '0' as soon as one or
        more test cases are initiated."
    ::= { pktcEnEndPntInfoTableEntry 6}

```

```

--
-- Compliance statements
--

```

```

pktcSigBasicCompliance MODULE-COMPLIANCE

```

```

    STATUS          current
    DESCRIPTION
        "The compliance statement for devices that implement
        PacketCable defined Signalling on an MTA."

```

```

    MODULE PKTC-EN-SIG-MIB

```

```

--
-- Mandatory groups
--

```

```

MANDATORY-GROUPS {
    pktcEnSigGroup
}
GROUP pktcEnNcsGroup
DESCRIPTION
    "This group is mandatory for any MTA implementing
    PacketCable signalling."
 ::= { pktcEnSigCompliances 1 }

```

```

--
-- Conformance group for common Signalling.
--
pktcEnSigGroup OBJECT-GROUP
    OBJECTS {
        pktcEnNcsMinimumDtmfPlayOut
    }
    STATUS current
    DESCRIPTION
        "Enhanced group of objects for the common portion of the
        PacketCable Signalling MIB."
    ::= { pktcEnSigGroups 1 }

--
-- Conformance group for NCS Signalling.
--
pktcEnNcsGroup OBJECT-GROUP
    OBJECTS {
        pktcEnNcsEndPntQuarantineState,
        pktcEnNcsEndPntHookState,
        pktcEnNcsEndPntFaxDetection,
        pktcEnNcsEndPntStatusReportCtrl,
        pktcEnEndPntFgnPotSupport,
        pktcEnEndPntFgnPotDescr,
        pktcEnEndPntClrFgnPotTsts,
        pktcEnEndPntRunFgnPotTsts,
        pktcEnEndPntFgnTestValidity,
        pktcEnEndPntFgnTestResults
    }
    STATUS current
    DESCRIPTION
        "Enhanced group of objects for the NCS portion of the
        PacketCable Signalling MIB. This is mandatory for
        NCS signalling support."
    ::= { pktcEnSigGroups 2 }
END

```


SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems