

Network Working Group  
Request for Comments: 5098  
Category: Standards Track

G. Beacham  
Motorola, Inc.  
S. Kumar  
Texas Instruments  
S. Channabasappa  
CableLabs  
February 2008

Signaling MIB for PacketCable and IPCablecom  
Multimedia Terminal Adapters (MTAs)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a basic set of managed objects for Simple Network Management Protocol (SNMP)-based management of PacketCable- and IPCablecom-compliant Multimedia Terminal Adapter devices.

## Table of Contents

1. The Internet-Standard Management Framework .....	2
2. Introduction .....	2
3. Terminology .....	3
3.1. MTA .....	3
3.2. Endpoint .....	3
3.3. L Line Package .....	4
3.4. E Line Package .....	4
4. Overview .....	4
4.1. Structure of the MIB .....	5
4.2. pktcSigMibObjects .....	5
4.3. pktcSigConformance .....	6
5. Definitions .....	6
6. Examples .....	69
7. Acknowledgments .....	72
8. Security Considerations .....	73
9. IANA Considerations .....	75
10. References .....	75
10.1. Normative References .....	75
10.2. Informative References .....	76

## 1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

## 2. Introduction

A multimedia terminal adapter (MTA) is used to deliver broadband Internet, data, and/or voice access jointly with telephony service to a subscriber's or customer's premises using a cable network infrastructure. An MTA is normally installed at the customer's or subscriber's premises, and it is coupled to a multiple system operator (MSO) using a hybrid fiber coax (HFC) access network.

An MTA is provisioned by the MSO for broadband Internet, data, and/or voice service. For more information on MTA provisioning, refer to

the PacketCable Provisioning Specification [PKT-SP-PROV] and [RFC4682]. MTA devices include one or more endpoints (e.g., telephone ports), which receive call signaling information to establish ring cadence, and codecs used for providing telephony service. For more information on call signaling, refer to the PacketCable Signaling Specification [PKT-SP-MGCP] and [RFC3435]. For more information on codecs refer to the PacketCable Audio/Video Codecs Specification [PKT-SP-CODEC].

Telephone systems are typically very complex and often have a wide distribution. It is therefore important for management systems to support MTAs from multiple vendors at the same time, including those from multiple countries. This MIB module provides objects suitable for managing signaling for MTA devices in the widest possible range of markets.

### 3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The terms "MIB module" and "information module" are used interchangeably in this memo. As used here, both terms refer to any of the three types of information modules defined in Section 3 of RFC 2578 [RFC2578].

#### 3.1. MTA

An MTA is a PacketCable or IPCablecom compliant device providing telephony services over a cable or hybrid system used to deliver video signals to a community. It contains an interface to endpoints, a network interface, codecs, and all signaling and encapsulation functions required for Voice-over IP transport, call signaling, and Quality of Service signaling. An MTA can be an embedded or standalone device. An Embedded MTA (E-MTA) is an MTA device containing an embedded Data Over Cable Service Interface Specifications (DOCSIS) Cable Modem. A Standalone MTA (S-MTA) is an MTA device separated from the DOCSIS Cable Modem by non-DOCSIS Media Access Control (MAC) interface (e.g., Ethernet, USB).

#### 3.2. Endpoint

An endpoint or MTA endpoint is a standard telephony physical port located on the MTA and used for attaching the telephone device to the MTA.

### 3.3. L Line Package

The L line package refers to the Media Gateway Control Protocol (MGCP) package for the core signaling functionality, as defined by PacketCable and IPCablecom. An MTA provides all L package elements: however, the operator determines their application.

### 3.4. E Line Package

The E line package refers to the MGCP package extensions, over and above the core L package, defined in support of international requirements. E line package elements are optional, vary from country to country, and are set by operator or regulatory requirements.

## 4. Overview

This MIB module provides a set of objects required for Multimedia Terminal Adapter (MTA) devices compliant with the PacketCable and IPCablecom signaling specifications published by CableLabs, the European Telecommunications Standards Institute (ETSI), and the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) IPCablecom compliant Multimedia Terminal Adapter (MTA) devices. The Signaling MIB module (PKTC-IETF-SIG-MIB) is intended to update various Signaling MIB modules from which it is partly derived:

- the PacketCable 1.0 Signaling MIB Specification [PKT-SP-MIB-SIG-1.0],
- the PacketCable 1.5 Signaling MIB Specification [PKT-SP-MIB-SIG-1.5],
- the ITU-T IPCablecom Signaling MIB requirements [ITU-T-J169],
- the ETSI Signaling MIB [ETSI-TS-101-909-9]. The ETSI Signaling MIB requirements also refer to various signal characteristics defined in [ETSI-TS-101-909-4], [ETSI-EN-300-001], [ETSI-EN-300-659-1], [ETSI-EN-300-324-1] and [ETSI-TR-101-183].

Several normative and informative references are used to help define Signaling MIB objects. As a convention, wherever PacketCable and IPCablecom requirements are equivalent, the PacketCable reference is used in the object REFERENCE clause. IPCablecom compliant MTA devices MUST use the equivalent IPCablecom references.

This MIB module describes the various Signaling MIB objects that are directly related to the PacketCable MTA and the endpoints supported on the MTA, each of which provides services independently. The recognition and distinction of the endpoints are made by utilizing the ifTable (IF-MIB [RFC2863]), where each index (ifIndex) value refers to a unique endpoint. This MIB module also utilizes the syntax definition of the Differentiated Services Code Point (DSCP) from DIFFSERV-DSCP-TC [RFC3289] for defining MIB objects that allow for differentiation between various types of traffic in the service provider network.

#### 4.1. Structure of the MIB

This MIB module is identified by pktcIetfSigMib and is structured into two major parts:

- Signaling information that controls device and endpoint configuration (pktcSigMibObjects)
- Module Conformance information (pktcSigConformance)

The following sections explain each part in further detail. It is to be noted that future enhancements to specify Notification Objects are also allowed (pktcSigNotification).

#### 4.2. pktcSigMibObjects

This is further divided into device-specific elements (pktcSigDevObjects) and endpoint-specific elements (pktcSigEndPntConfigObjects).

Some highlights of the device-specific elements are as follows:

pktcSigDevCodecTable - this object identifies the codec types available on the device.

pktcSigDevEchoCancellation - this object identifies the capability of echo cancellation on the device.

pktcSigDevSilenceSuppression - this object specifies if the device is capable of silence suppression (Voice Activity Detection).

pktcSigPulseSignalTable - this table selects the various signals used in the application of the metering pulse signal to the twisted pair line.

pktcSigDevToneTable - this table specifies a flexible structure within which to specify all of the tones used in the MTA.

pktcSigDevMultiFreqToneTable - this table defines the characteristics of tones with multiple frequencies. Each entry in this table represents the frequency reference of a multi-frequency tone.

The endpoint-specific elements are mostly confined to the Endpoint configuration MIB table (pktcSigEndPntConfigTable). This table describes the MTA endPoint configuration. The number of entries in this table represents the number of provisioned endpoints.

#### 4.3. pktcSigConformance

pktcSigDeviceGroup - this group contains all the MIB objects that apply on a per-device basis and need to be implemented by an MTA to claim compliance with the specified MIB module.

pktcSigEndpointGroup - this group contains all the MIB objects that apply on a per-endpoint basis and need to be implemented by an MTA to claim compliance with the specified MIB module.

pktcLLinePackageGroup - this group contains the MIB objects that need to be implemented to support the L line package.

pktcELinePackageGroup - this group contains the MIB objects that need to be implemented to support the E line package.

pktcInternationalGroup - this group contains optional MIB objects designed to support operations over the widest possible range of markets.

#### 5. Definitions

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

##### IMPORTS

```
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Integer32,
    Unsigned32,
    mib-2
        FROM SNMPv2-SMI                -- [RFC2578]
    InetAddressType,
    InetAddress,
    InetPortNumber
        FROM INET-ADDRESS-MIB          -- [RFC4001]
    TEXTUAL-CONVENTION,
    RowStatus,
    TruthValue
        FROM SNMPv2-TC                -- [RFC2579]
```

OBJECT-GROUP,  
MODULE-COMPLIANCE  
    FROM SNMPv2-CONF                    -- [RFC2580]  
SnmpAdminString  
    FROM SNMP-FRAMEWORK-MIB              -- [RFC3411]  
ifIndex  
    FROM IF-MIB                          -- [RFC2863]  
Dscp  
    FROM DIFFSERV-DSCP-TC;              -- [RFC3289]

pktcIetfSigMib MODULE-IDENTITY

LAST-UPDATED      "200712180000Z" -- December 18, 2007

ORGANIZATION      "IETF IPCDN Working Group"

CONTACT-INFO

"Sumanth Channabasappa  
Cable Television Laboratories, Inc.  
858 Coal Creek Circle,  
Louisville, CO 80027, USA  
Phone: +1 303-661-3307  
Email: Sumanth@cablelabs.com

Gordon Beacham  
Motorola, Inc.  
6450 Sequence Drive, Bldg. 1  
San Diego, CA 92121, USA  
Phone: +1 858-404-2334  
Email: gordon.beacham@motorola.com

Satish Kumar Mudugere Eswaraiah  
Texas Instruments India (P) Ltd.,  
Golf view, Wind Tunnel Road  
Muruges Palya  
Bangalore 560 017, INDIA  
Phone: +91 80 5269451  
Email: satish.kumar@ti.com

IETF IPCDN Working Group

General Discussion: [ipcdn@ietf.org](mailto:ipcdn@ietf.org)

Subscribe: <http://www.ietf.org/mailman/listinfo/ipcdn>

Archive: <ftp://ftp.ietf.org/ietf-mail-archive/ipcdn>

Co-Chair: Jean-Francois Mule, [jf.mule@cablelabs.com](mailto:jf.mule@cablelabs.com)

Co-Chair: Richard Woundy, [Richard\\_Woundy@cable.comcast.com](mailto:Richard_Woundy@cable.comcast.com)

DESCRIPTION

"This MIB module supplies the basic management objects for the PacketCable and IPCablecom Signaling protocols. This version of the MIB includes common signaling and Network Call Signaling

(NCS)-related signaling objects.

Copyright (C) The IETF Trust (2008). This version of this MIB module is part of RFC 5098; see the RFC itself for full legal notices."

REVISION "200712180000Z"

DESCRIPTION

"Initial version, published as RFC 5098."

::= { mib-2 169 }

-- Textual Conventions

TenthdBm ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d-1"

STATUS current

DESCRIPTION

"This TEXTUAL-CONVENTION represents power levels that are normally expressed in dBm. Units are in tenths of a dBm; for example, -13.5 dBm will be represented as -135."

SYNTAX Integer32

PktnCodecType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" This TEXTUAL-CONVENTION defines various types of codecs that MAY be supported. The description for each enumeration is listed below:

Enumeration	Description
other	a defined codec not in the enumeration
unknown	a codec not defined by the PacketCable Codec Specification
g729	ITU-T Recommendation G.729
reserved	for future use
g729E	ITU-T Recommendation G.729E
pcmu	Pulse Code Modulation u-law (PCMU)
g726at32	ITU-T Recommendation G.726-32 (32 kbit/s)
g728	ITU-T Recommendation G.728
pcma	Pulse Code Modulation a-law (PCMA)
g726at16	ITU-T Recommendation G.726-16 (16 kbit/s)
g726at24	ITU-T Recommendation G.726-24 (24 kbit/s)
g726at40	ITU-T Recommendation G.726-40 (40 kbit/s)
ilbc	IETF Internet low-bit rate codec
bv16	Broadcom BroadVoice16

The list of codecs is consistent with the IETF Real-Time Transport Protocol (RTP) Profile registry and



the RTP Map Parameters Table in PacketCable Audio/Video Codecs Specification [PKT-SP-CODEC]. The literal codec name for each codec is listed below:

Codec	Literal Codec Name
g729	G729
g729E	G729E
pcmu	PCMU
g726at32	G726-32
g728	G728
pcma	PCMA
g726at16	G726-16
g726at24	G726-24
g726at40	G726-40
ilbc	iLBC
bv16	BV16

The literal codec name is the second column of the table with codec RTP Map Parameters. The Literal Codec Name Column contains the codec name used in the local connection options (LCO) of the NCS messages create connection (CRCX)/modify connection (MDCX) and is also used to identify the codec in the Call Management System (CMS) Provisioning Specification. The RTP Map Parameter column of the Table contains the string used in the media attribute line (a=) of the session description protocol (SDP) parameters in NCS messages."

```
SYNTAX INTEGER {
    other          (1),
    unknown       (2),
    g729          (3),
    reserved      (4),
    g729E         (5),
    pcmu          (6),
    g726at32      (7),
    g728          (8),
    pcma          (9),
    g726at16      (10),
    g726at24      (11),
    g726at40      (12),
    ilbc          (13),
    bv16          (14)
}
```

```
PktcRingCadence ::= TEXTUAL-CONVENTION
    STATUS         current
    DESCRIPTION
```

"This object provides an encoding scheme for ring

cadences, including repeatability characteristics. All fields in this object MUST be encoded in network-byte order.

The first three higher-order octets are reserved. The octets that follow are used to encode a 'bit-string', with each bit corresponding to 50 milliseconds. A bit value of '1' indicates the presence of a ring-tone, and a bit value of '0' indicates the absence of a ring-tone, for that duration (50 ms) (Note: A minimum number of octets required to encode the bit-string MUST be used).

The first two of the reserved octets MUST indicate the length of the encoded cadence (in bits) and MUST range between 1 and 264. (Note: The length in bits MUST also be consistent with the number of octets that encode the cadence). The MTA MUST ignore any unused bits in the last octet, but MUST reflect the value as provided on subsequent SNMP GETs.

The third of the reserved octets indicates 'repeatability' and MUST be either 0x80 or 0x00 -- the former value indicating 'non-repeatability', and the latter indicating 'repeatability'.

The MTA MUST reject attempts to set a value that violates any of the above requirements."

SYNTAX OCTET STRING (SIZE(4..36))

PktcSigType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" This object lists the various types of signaling that may be supported:

other(1) - set when signaling other than NCS is used  
ncs(2) - Network Call Signaling is a derivation of MGCP (Media Gateway Control Protocol) defined for IPCablecom/PacketCable MTAs."

SYNTAX INTEGER {  
    other(1),  
    ncs(2)  
}

DtmfCode ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This TEXTUAL-CONVENTION represents the Dual-Tone Multi-Frequency (DTMF) Character used to indicate the start or end of the digit transition sequence used for caller id or Visual Message Waiting Indicator (VMWI).

Note: The DTMF code '\*' is indicated using 'dtmfcodeStar', and the DTMF code '#' is indicated using ' dtmfcodeHash'."

SYNTAX INTEGER {  
    dtmfcode0(0),  
    dtmfcode1(1),  
    dtmfcode2(2),  
    dtmfcode3(3),  
    dtmfcode4(4),  
    dtmfcode5(5),  
    dtmfcode6(6),  
    dtmfcode7(7),  
    dtmfcode8(8),  
    dtmfcode9(9),  
    dtmfcodeStar(10),  
    dtmfcodeHash(11),  
    dtmfcodeA(12),  
    dtmfcodeB(13),  
    dtmfcodeC(14),  
    dtmfcodeD(15)

}

PktcSubscriberSideSigProtocol ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This TEXTUAL-CONVENTION represents the Signaling protocol being used for purposes such as caller id or VMWI.

A value of fsk(1) indicates Frequency Shift Keying (FSK).

A value of dtmf(2) indicates Dual-Tone Multi-Frequency (DTMF)."

SYNTAX INTEGER {  
    fsk(1),  
    dtmf(2)  
}

pktcSigMibObjects OBJECT IDENTIFIER ::= { pktcIetfSigMib 1 }  
pktcSigDevObjects OBJECT IDENTIFIER ::=

```

                                { pktcSigMibObjects 1 }
pktcSigEndPntConfigObjects OBJECT IDENTIFIER ::=
                                { pktcSigMibObjects 2 }
--
-- The codec table (pktcSigDevCodecTable) defines all combinations
-- of codecs supported by the Multimedia Terminal Adapter (MTA).
--
pktcSigDevCodecTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PktcSigDevCodecEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the MTA-supported codec types.  An MTA
        MUST populate this table with all possible combinations of
        codecs it supports for simultaneous operation.  For example,
        an MTA with two endpoints may be designed with a particular
        Digital Signal Processing (DSP) and memory architecture that
        allows it to support the following fixed combinations of
        codecs for simultaneous operation:

        Codec Type          Maximum Number of Simultaneous Codecs

        PCMA                  3
        PCMA                  2
        PCMU                  1
        PCMA                  1
        PCMU                  2
        PCMU                  3
        PCMA                  1
        G729                  1
        G729                  2
        PCMU                  1
        G729                  1
    
```

Based on this example, the entries in the codec table would be:

pktcSigDev CodecComboIndex	pktcSigDev CodecType	pktcSigDev CodecMax
1	pcma	3
2	pcma	2
2	pcmu	1

3	pcma	1
3	pcmu	2
4	pcmu	3
5	pcma	1
5	g729	1
6	g729	2
7	pcmu	1
7	g729	1

An operator querying this table is able to determine all possible codec combinations the MTA is capable of simultaneously supporting.

This table MUST NOT include non-voice codecs."

```
::= { pktcSigDevObjects 1 }
```

pktcSigDevCodecEntry OBJECT-TYPE

SYNTAX PktcSigDevCodecEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry represents the maximum number of active connections with a particular codec the MTA is capable of supporting. Each row is indexed by a composite key consisting of a number enumerating the particular codec combination and the codec type."

INDEX { pktcSigDevCodecComboIndex, pktcSigDevCodecType }

```
::= { pktcSigDevCodecTable 1 }
```

```
PktcSigDevCodecEntry ::= SEQUENCE {
    pktcSigDevCodecComboIndex    Unsigned32,
    pktcSigDevCodecType          PktcCodecType,
    pktcSigDevCodecMax           Unsigned32
}
```

pktcSigDevCodecComboIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..255)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" The index value that enumerates a particular codec combination in the pktcSigDevCodecTable."

```
::= { pktcSigDevCodecEntry 1 }
```

pktcSigDevCodecType OBJECT-TYPE

SYNTAX PktcCodecType

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

" A codec type supported by this MTA."

::= { pktcSigDevCodecEntry 2 }

## pktcSigDevCodecMax OBJECT-TYPE

SYNTAX Unsigned32(1..255)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

" The maximum number of simultaneous sessions of a particular codec that the MTA can support."

::= { pktcSigDevCodecEntry 3 }

--

-- These are the common signaling-related definitions that affect  
-- the entire MTA device.

--

## pktcSigDevEchoCancellation OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

" This object specifies if the device is capable of echo cancellation. The MTA MUST set this MIB object to a value of true(1) if it is capable of echo cancellation, and a value of false(2) if not."

::= { pktcSigDevObjects 2 }

## pktcSigDevSilenceSuppression OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

" This object specifies if the device is capable of silence suppression (as a result of Voice Activity Detection). The MTA MUST set this MIB object to a value of true(1) if it is capable of silence suppression, and a value of false(2) if not."

::= { pktcSigDevObjects 3 }

## pktcSigDevCidSigProtocol OBJECT-TYPE

SYNTAX PktcSubscriberSideSigProtocol

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This object is used to configure the subscriber-line protocol used for signaling on-hook caller id information."

Different countries define different caller id signaling protocols to support caller identification.

Setting this object at a value `fsk(1)` sets the subscriber line protocol to be Frequency Shift Keying (FSK).

Setting this object at a value `dtmf(2)` sets the subscriber line protocol to be Dual-Tone Multi-Frequency (DTMF).

The value of this MIB object MUST NOT persist across MTA reboots."

REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { `fsk` }

::= { `pktcSigDevObjects` 4 }

`pktcSigDevR0Cadence` OBJECT-TYPE

SYNTAX `PktcRingCadence`

MAX-ACCESS `read-write`

STATUS `current`

DESCRIPTION

" This object specifies ring cadence 0 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { `pktcSigDevObjects` 5 }

`pktcSigDevR1Cadence` OBJECT-TYPE

SYNTAX `PktcRingCadence`

MAX-ACCESS `read-write`

STATUS `current`

DESCRIPTION

" This object specifies ring cadence 1 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { `pktcSigDevObjects` 6 }

`pktcSigDevR2Cadence` OBJECT-TYPE

SYNTAX `PktcRingCadence`

MAX-ACCESS `read-write`

STATUS `current`

DESCRIPTION

" This object specifies ring cadence 2 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 7 }

pktcSigDevR3Cadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence 3 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 8 }

pktcSigDevR4Cadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence 4 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 9 }

pktcSigDevR5Cadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence 5 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 10 }

pktcSigDevR6Cadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence 6 (a user-defined field).



The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 11 }

pktcSigDevR7Cadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence 7 (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 12 }

pktcSigDevRgCadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence rg (a user-defined field).

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 13 }

pktcSigDevRsCadence OBJECT-TYPE

SYNTAX PktcRingCadence

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies ring cadence rs (a user-defined field). The MTA MUST reject any attempt to make this object repeatable.

The value of this MIB object MUST NOT persist across MTA reboots."

::= { pktcSigDevObjects 14 }

pktcSigDefCallSigDscp OBJECT-TYPE

SYNTAX Dscp -- RFC 3289: DIFFSERV-DSCP-TC

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" The default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for call

signaling.

The value of this MIB object MUST NOT persist across MTA reboots."

DEFVAL { 0 }

::= { pktcSigDevObjects 15 }

pktcSigDefMediaStreamDscp OBJECT-TYPE

SYNTAX Dscp -- RFC 3289: DIFFSERV-DSCP-TC

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object contains the default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for media stream packets. The MTA MUST NOT update this object with the value supplied by the CMS in the NCS messages (if present). Any currently active connections are not affected by updates to this object. When the value of this object is updated by SNMP, the MTA MUST use the new value as a default starting only from new connections.

The value of this MIB object MUST NOT persist across MTA reboots."

DEFVAL { 0 }

::= { pktcSigDevObjects 16 }

--

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

--

pktcSigCapabilityTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcSigCapabilityEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" This table describes the signaling types supported by this MTA."

::= { pktcSigDevObjects 17 }

pktcSigCapabilityEntry OBJECT-TYPE

SYNTAX PktcSigCapabilityEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" Entries in pktcMtaDevSigCapabilityTable - list of supported signaling types, versions, and vendor extensions

for this MTA. Each entry in the list provides for one signaling type and version combination. If the device supports multiple versions of the same signaling type, it will require multiple entries."

```
INDEX { pktcSigCapabilityIndex }
 ::= { pktcSigCapabilityTable 1 }
```

```
PktcSigCapabilityEntry ::= SEQUENCE {
    pktcSigCapabilityIndex      Unsigned32,
    pktcSigCapabilityType       PktcSigType,
    pktcSigCapabilityVersion     SnmpAdminString,
    pktcSigCapabilityVendorExt   SnmpAdminString
}
```

```
pktcSigCapabilityIndex      OBJECT-TYPE
    SYNTAX      Unsigned32 (1..255)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        " The index value that uniquely identifies an entry in the
          pktcSigCapabilityTable."
    ::= { pktcSigCapabilityEntry 1 }
```

```
pktcSigCapabilityType       OBJECT-TYPE
    SYNTAX      PktcSigType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " This object identifies the type of signaling used. This
          value has to be associated with a single signaling
          version."
    ::= { pktcSigCapabilityEntry 2 }
```

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

```
pktcSigCapabilityVendorExt   OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " The vendor extension allows vendors to provide a list of
```

additional capabilities.

The syntax for this MIB object in ABNF ([RFC5234]) is specified to be zero or more occurrences of vendor extensions, as follows:

```
pktcSigCapabilityVendorExt = *(vendor-extension)
vendor-extension = (ext symbol alphanum) DQUOTE ; DQUOTE
ext              = DQUOTE %x58 DQUOTE
symbol           = (DQUOTE %x2D DQUOTE)/(DQUOTE %x2D DQUOTE)
alphanum         = 1*6(ALPHA/DIGIT)
```

```
"
 ::= { pktcSigCapabilityEntry 4 }
```

pktcSigDefNcsReceiveUdpPort OBJECT-TYPE

SYNTAX InetPortNumber (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 signaling. This object should only be changed by the configuration file.

Unless changed via configuration, this MIB object MUST reflect a value of '2427'."

REFERENCE

"PacketCable NCS Specification"

```
::= { pktcSigDevObjects 18 }
```

pktcSigPowerRingFrequency OBJECT-TYPE

```
SYNTAX INTEGER {
    f20Hz(1),
    f25Hz(2),
    f33Point33Hz(3),
    f50Hz(4),
    f15Hz(5),
    f16Hz(6),
    f22Hz(7),
    f23Hz(8),
    f45Hz(9)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This object must only be provided via the configuration file during the provisioning process. The power ring

frequency is the frequency at which the sinusoidal voltage must travel down the twisted pair to make terminal equipment ring. Different countries define different electrical characteristics to make terminal equipment ring.

The f20Hz setting corresponds to a power ring frequency of 20 Hertz. The f25Hz setting corresponds to a power ring frequency of 25 Hertz. The f33Point33Hz setting corresponds to a power ring frequency of 33.33 Hertz. The f50Hz setting corresponds to a power ring frequency of 50 Hertz. The f15Hz setting corresponds to a power ring frequency of 15 Hertz. The f16Hz setting corresponds to a power ring frequency of 16 Hertz. The f22Hz setting corresponds to a power ring frequency of 22 Hertz. The f23Hz setting corresponds to a power ring frequency of 23 Hertz. The f45Hz setting corresponds to a power ring frequency of 45 Hertz."

## REFERENCE

"ETSI-EN-300-001"

::= { pktcSigDevObjects 19 }

pktcSigPulseSignalTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcSigPulseSignalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

" The Pulse signal table defines the pulse signal operation. There are nine types of international pulse signals, with each signal having a set of provisionable parameters. The values of the MIB objects in this table take effect only if these parameters are not defined via signaling, in which case, the latter determines the values of the parameters. The MIB objects in this table do not persist across MTA reboots."

## REFERENCE

"ETSI-TS-101-909-4 Specification"

::= { pktcSigDevObjects 20 }

pktcSigPulseSignalEntry OBJECT-TYPE

SYNTAX PktcSigPulseSignalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

" This object defines the set of parameters associated with each particular value of pktcSigPulseSignalType. Each entry in the pktcSigPulseSignalTable is indexed by the pktcSigPulseSignalType object."

The conceptual rows MUST NOT persist across MTA reboots."

```
INDEX { pktcSigPulseSignalType }
 ::= { pktcSigPulseSignalTable 1 }
```

```
PktcSigPulseSignalEntry ::= SEQUENCE {
    pktcSigPulseSignalType          INTEGER,
    pktcSigPulseSignalFrequency     INTEGER,
    pktcSigPulseSignalDbLevel       TenthdBm,
    pktcSigPulseSignalDuration      Unsigned32,
    pktcSigPulseSignalPulseInterval Unsigned32,
    pktcSigPulseSignalRepeatCount   Unsigned32
}
```

```
pktcSigPulseSignalType OBJECT-TYPE
    SYNTAX      INTEGER
                {
                    initialRing(1),
                    pulseLoopClose(2),
                    pulseLoopOpen(3),
                    enableMeterPulse(4),
                    meterPulseBurst(5),
                    pulseNoBattery(6),
                    pulseNormalPolarity(7),
                    pulseReducedBattery(8),
                    pulseReversePolarity(9)
                }
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"There are nine types of international pulse signals. These signals are defined as follows:

```
initial ring
pulse loop close
pulse loop open
enable meter pulse
meter pulse burst
pulse no battery
pulse normal polarity
pulse reduced battery
pulse reverse polarity"
```

```
REFERENCE
```

"ETSI-EN-300-324-1 Specification"

```
::= { pktcSigPulseSignalEntry 1 }
```

```
pktcSigPulseSignalFrequency OBJECT-TYPE
    SYNTAX      INTEGER {
                    twentyfive(1),
```

```

        twelvethousand(2),
        sixteenthousand(3)
    }
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    " This object is only applicable to the initialRing,
      enableMeterPulse, and meterPulseBurst signal types. This
      object identifies the frequency of the generated signal.
      The following table defines the default values for this
      object depending on signal type:

```

pktcSigPulseSignalType	Default
initialRing	25
enableMeterPulse	16000
meterPulseBurst	16000

The value of twentyfive MUST only be used for the initialRing signal type. The values of twelvethousand and sixteenthousand MUST only be used for enableMeterPulse and meterPulseBurst signal types. An attempt to set this object while the value of pktcSigPulseSignalType is not initialRing, enableMeterPulse, or meterPulseBurst will result in an 'inconsistentValue' error."

## REFERENCE

"ETSI-EN-300-001 Specification"  
 ::= { pktcSigPulseSignalEntry 2 }

```

pktcSigPulseSignalDbLevel      OBJECT-TYPE
SYNTAX      TenthdBm (-350..0)
UNITS       "1/10 of a dBm"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION

```

" This object is only applicable to the enableMeterPulse and meterPulseBurst signal types. This is the decibel level for each frequency at which tones could be generated at the a and b terminals (TE connection point). An attempt to set this object while the value of pktcSigPulseSignalType is not enableMeterPulse or meterPulseBurst will result in an 'inconsistentValue' error."

## REFERENCE

"ETSI-EN-300-001 Specification"

```

DEFVAL { -135 }
::={pktcSigPulseSignalEntry 3 }

```

```

pktcSigPulseSignalDuration      OBJECT-TYPE

```

SYNTAX Unsigned32 (0..5000)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" This object specifies the pulse duration for each signal type. In addition, the MTA must accept the values in the incremental steps specific for each signal type. The following table defines the default values and the incremental steps for this object depending on the signal type:

pktcSigPulseSignaltype	Default (ms)	Increment (ms)
initialRing	200	50
pulseLoopClose	200	10
pulseLoopOpen	200	10
enableMeterPulse	150	10
meterPulseBurst	150	10
pulseNoBattery	200	10
pulseNormalPolarity	200	10
pulseReducedBattery	200	10
pulseReversePolarity	200	10

An attempt to set this object to a value that does not fall on one of the increment boundaries, or on the wrong increment boundary for the specific signal type, will result in an 'inconsistentValue' error."

#### REFERENCE

"ETSI-EN-300-324-1 Specification"  
 ::= {pktcSigPulseSignalEntry 4 }

pktcSigPulseSignalPulseInterval OBJECT-TYPE

SYNTAX Unsigned32 (0..5000)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" This object specifies the repeat interval, or the period, for each signal type. In addition, the MTA must accept the values in the incremental steps specific for each signal type. The following table defines the default values and the incremental steps for this object, depending on the signal type:

pktcSigPulseSignaltype	Default (ms)	Increment (ms)
initialRing	200	50
pulseLoopClose	1000	10
pulseLoopOpen	1000	10



enableMeterPulse	1000	10
meterPulseBurst	1000	10
pulseNoBattery	1000	10
pulseNormalPolarity	1000	10
pulseReducedBattery	1000	10
pulseReversePolarity	1000	10

An attempt to set this object to a value that does not fall on one of the increment boundaries, or on the wrong increment boundary for the specific signal type, will result in an 'inconsistentValue' error."

## REFERENCE

"ETSI-EN-300-324-1 Specification"  
 ::= { pktcSigPulseSignalEntry 5 }

pktcSigPulseSignalRepeatCount OBJECT-TYPE

SYNTAX Unsigned32 (1..50)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

" This object specifies how many times to repeat a pulse. This object is not used by the enableMeterPulse signal type, and in that case, the value is irrelevant. The following table defines the default values and the valid ranges for this object, depending on the signal type:

pktcSigPulseSignaltype	Default	Range
initialRing	1	1-5
pulseLoopClose	1	1-50
pulseLoopOpen	1	1-50
enableMeterPulse	(any value)	(but not used)
meterPulseBurst	1	1-50
pulseNoBattery	1	1-50
pulseNormalPolarity	1	1-50
pulseReducedBattery	1	1-50
pulseReversePolarity	1	1-50

An attempt to set this object to a value that does not fall within the range for the specific signal type will result in an 'inconsistentValue' error."

::={ pktcSigPulseSignalEntry 6 }

pktcSigDevCidMode OBJECT-TYPE

SYNTAX INTEGER {  
 duringRingingETS(1),  
 dtAsETS(2),  
 rpAsETS(3),

```

        lrAsETS(4),
        lrETS(5)
    }

```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" For on-hook caller id, pktcSigDevCidMode selects the method for representing and signaling caller identification. For the duringRingingETS method, the Frequency Shift Keying (FSK) or the Dual-Tone Multi-Frequency (DTMF) containing the caller identification information is sent between the first and second ring pattern.

For the dtAsETS, rpAsETS, lrAsETS and lrETS methods, the FSK or DTMF containing the caller id information is sent before the first ring pattern.

For the dtAsETS method, the FSK or DTMF is sent after the Dual Tone Alert Signal. For the rpAsETS method, the FSK or DTMF is sent after a Ring Pulse.

For the lrAsETS method, the Line Reversal occurs first, then the Dual Tone Alert Signal, and, finally, the FSK or DTMF is sent.

For the lrETS method, the Line Reversal occurs first, then the FSK or DTMF is sent.

The value of this MIB object MUST NOT persist across MTA reboots."

```

DEFVAL { rpAsETS }
 ::= {pktcSigDevObjects 21 }

```

```

pktcSigDevCidAfterRing      OBJECT-TYPE
    SYNTAX      Unsigned32 (0|50..2000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

" This object specifies the delay between the end of first ringing pattern and the start of the transmission of the FSK or DTMF containing the caller id information. It is only used when pktcSigDevCidMode is set to a value of 'duringRingingETS'.

The following table defines the default values for this MIB object, depending on the signal type

(pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	550 ms
dtAsETS	any value (not used)
rpAsETS	any value (not used)
lrAsETS	any value (not used)
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevCidMode is not duringringingETS will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 550 }

::= {pktcSigDevObjects 22 }

pktcSigDevCidAfterDTAS OBJECT-TYPE

SYNTAX Unsigned32 (0|45..500)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" This object specifies the delay between the end of the Dual Tone Alert Signal (DT-AS) and the start of the transmission of the FSK or DTMF containing the caller id information. This object is only used when pktcSigDevCidMode is set to a value of 'dtAsETS' or 'lrAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	50 ms
rpAsETS	any value (not used)
lrAsETS	50 ms
lrETS	any value (not used)

An attempt to set this object while the value of

pktcSigDevCidMode is not 'dtAsETS' or 'lrAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

## REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 50 }

::= {pktcSigDevObjects 23 }

pktcSigDevCidAfterRPAS      OBJECT-TYPE  
SYNTAX            Unsigned32 (0|500..800)  
UNITS             "Milliseconds"  
MAX-ACCESS       read-write  
STATUS            current  
DESCRIPTION

" This object specifies the delay between the end of the Ring Pulse Alert Signal (RP-AS) and the start of the transmission of the FSK or DTMF containing the caller id information. This MIB object is only used when pktcSigDevCidMode is set to a value of 'rpAsETS'. The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	any value (not used)
rpAsETS	650 ms
lrAsETS	any value (not used)
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevCidMode is not 'rpAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

## REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 650 }

::= {pktcSigDevObjects 24 }

pktcSigDevRingAfterCID      OBJECT-TYPE  
SYNTAX            Unsigned32 (0|50..500)  
UNITS             "Milliseconds"  
MAX-ACCESS       read-write

STATUS current

#### DESCRIPTION

" This object specifies the delay between the end of the complete transmission of the FSK or DTMF containing the caller id information and the start of the first ring pattern. It is only used when pktcSigDevCidMode is set to a value of 'dtAsETS', 'rpAsETS', 'lrAsETS' or 'lrETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	250 ms
rpAsETS	250 ms
lrAsETS	250 ms
lrETS	250 ms

An attempt to set this object while the value of pktcSigDevCidMode is not 'dtAsETS', 'rpAsETS', 'lrAsETS', or 'lrETS' will result in an 'inconsistent value' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 250 }

::= {pktcSigDevObjects 25 }

pktcSigDevCidDTASAfterLR OBJECT-TYPE

SYNTAX Unsigned32 (50..655)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" This object specifies the delay between the end of the Line Reversal and the start of the Dual Tone Alert Signal (DT-AS). This object is only used when pktcSigDevCidMode is set to a value of 'lrAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	any value (not used)
rpAsETS	any value (not used)
lrAsETS	250 ms
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevCidMode is not lrAsETS will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 250 }

::= {pktcSigDevObjects 26 }

pktcSigDevVmwMode OBJECT-TYPE

```
SYNTAX      INTEGER {
                dtAsETS(1),
                rpAsETS(2),
                lrAsETS(3),
                osi(4),
                lrETS(5)
            }
```

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" For visual message waiting indicator (VMWI), pktcSigDevVmwMode selects the alerting signal method. For the dtAsETS, rpAsETS, lrAsETS, osi, and lrETS methods, the FSK containing the VMWI information is sent after an alerting signal.

For the dtAsETS method, the FSK, or DTMF is sent after the Dual Tone Alert Signal. For the rpAsETS method, the FSK or DTMF is sent after a Ring Pulse.

For the lrAsETS method, the Line Reversal occurs first, then the Dual Tone Alert Signal, and, finally, the FSK or DTMF is sent.

For the OSI method, the FSK or DTMF is sent after the Open Switching Interval.

For the lrETS method, the Line Reversal occurs first, then the FSK or DTMF is sent.

The value of this MIB object MUST NOT persist across MTA reboots."

```
DEFVAL { rpAsETS }
 ::= {pktcSigDevObjects 27 }
```

```
pktcSigDevVmwAfterDTAS      OBJECT-TYPE
    SYNTAX      Unsigned32 (0|45..500)
    UNITS       "Milliseconds"
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
```

" This object specifies the delay between the end of the Dual Tone Alert Signal (DT-AS) and the start of the transmission of the FSK or DTMF containing the VMWI information.

This object is only used when pktcSigDevVmwMode is set to a value of 'dtAsETS' or 'lrAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevVmwMode), and MUST be followed:

Value of pktcSigDevVmwMode	Default value
dtAsETS	50 ms
rpAsETS	any value (not used)
lrAsETS	50 ms
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevVmwMode is not 'dtAsETS' or 'lrAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

```
DEFVAL { 50 }
 ::= {pktcSigDevObjects 28 }
```

```
pktcSigDevVmwAfterRPAS      OBJECT-TYPE
    SYNTAX      Unsigned32 (0|500..800)
```

UNITS "Milliseconds"  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

" This object specifies the delay between the end of the Ring Pulse Alert Signal (RP-AS) and the start of the transmission of the FSK or DTMF containing the VMWI information.

This object is only used when pktcSigDevVmwiMode is set to a value of 'rpAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevVmwiMode), and MUST be followed:

Value of pktcSigDevVmwiMode	Default value
dtAsETS	any value (not used)
rpAsETS	650 ms
lrAsETS	any value (not used)
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevVmwiMode is not 'rpAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 650 }

::= {pktcSigDevObjects 29 }

pktcSigDevVmwiDTASafterLR OBJECT-TYPE

SYNTAX Unsigned32 (0|50..655)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

" This object specifies the delay between the end of the Line Reversal and the start of the Dual Tone Alert Signal (DT-AS) for VMWI information. This object is only used when pktcSigDevVmwiMode is set to a value of 'lrAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevVmwiMode), and MUST be followed:



Value of pktcSigDevVmwiMode	Default value
dtAsETS	any value (not used)
rpAsETS	any value (not used)
lrAsETS	250 ms
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevVmwiMode is not 'lrAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 Specification"

DEFVAL { 250 }

::= {pktcSigDevObjects 30 }

#### pktcSigDevRingCadenceTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcSigDevRingCadenceEntry

MAX-ACCESS not-accessible

STATUS current

#### DESCRIPTION

"Cadence rings are defined by the telco governing body for each country. The MTA must be able to support various ranges of cadence patterns and cadence periods. The MTA will be able to support country-specific provisioning of the cadence and idle period. Each cadence pattern will be assigned a unique value ranging from 0-127 (inclusive) corresponding to the value of x, where x is the value sent in the cadence ringing (cr) signal cr(x), requested per the appropriate NCS message, and defined in the E package. The MTA will derive the cadence periods from the ring cadence table entry, as provisioned by the customer. The MTA is allowed to provide appropriate default values for each of the ring cadences. This table only needs to be supported when the MTA implements the E package."

#### REFERENCE

"ETSI-TS-101-909-4 Specification"

::= { pktcSigDevObjects 31 }

#### pktcSigDevRingCadenceEntry OBJECT-TYPE

SYNTAX PktcSigDevRingCadenceEntry

MAX-ACCESS not-accessible

STATUS current

#### DESCRIPTION

" Each entry in this row corresponds to a ring cadence that is being supported by the device. The conceptual rows MUST NOT persist across MTA reboots."

```
INDEX { pktcSigDevRingCadenceIndex }
 ::= { pktcSigDevRingCadenceTable 1 }
```

```
PktcSigDevRingCadenceEntry ::= SEQUENCE {
    pktcSigDevRingCadenceIndex    Unsigned32,
    pktcSigDevRingCadence         PktcRingCadence
}
```

```
pktcSigDevRingCadenceIndex    OBJECT-TYPE
```

```
SYNTAX      Unsigned32 (0..127)
```

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

```
STATUS      current
```

```
DESCRIPTION
```

" A unique value ranging from 0 to 127 that corresponds to the value sent by the LE based on country-specific cadences, one row per cadence cycle. In any given system implementation for a particular country, it is anticipated that a small number of ring cadences will be in use. Thus, this table most likely will not be populated to its full size."

```
::= { pktcSigDevRingCadenceEntry 1 }
```

```
pktcSigDevRingCadence        OBJECT-TYPE
```

```
SYNTAX      PktcRingCadence
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"This is the Ring Cadence."

```
::= { pktcSigDevRingCadenceEntry 2 }
```

```
pktcSigDevToneTable          OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF PktcSigDevToneEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

" The Tone Table defines the composition of tones and various tone operations.

The definition of the tones callWaiting1 through callWaiting4 in this table MUST only contain the audible tone itself; the delay between tones or the value of the tone repeat count are not applicable for the call waiting tones.

The delay between tones or the repeat count is controlled by the objects `pktcSigEndPntConfigCallWaitingDelay` and `pktcSigEndPntConfigCallWaitingMaxRep`. If the `pktcSigDevToneType` is set to either of the values `callWaiting1`, `callWaiting2`, `callWaiting3`, or `callWaiting4`, then the value of the `pktcSigDevToneWholeToneRepeatCount` object indicates that the particular frequency group is applicable, as a repeatable part of the tone, based on the value of the MIB object `pktcSigDevToneWholeToneRepeatCount`.

The MTA MUST make sure that, after the provisioning cycle, the table is fully populated (i.e., for each possible index, an entry MUST be defined) using reasonable defaults for each row that was not defined by the provisioning information delivered via MTA Configuration.

The frequency composition of each tone is defined by the `pktcSigDevMultiFreqToneTable`. For each tone type defined in `pktcSigDevToneTable`, the MTA MUST populate at least one entry in the `pktcSigDevMultiFreqToneTable`.

For each particular value of `pktcSigDevToneType`, the `pktcSigDevToneTable` table can define non-repeating and repeating groups of the frequencies defined by the `pktcSigDevMultiFreqToneTable`, such that each group is represented by the set of the consecutive rows (frequency group) in the `pktcSigDevMultiFreqToneTable`.

Objects in this table do not persist across MTA reboots. For tones with multiple frequencies refer to the MIB table `pktcSigDevMultiFreqToneTable`."

#### REFERENCE

"PacketCable NCS Specification, ETSI-TS-101-909-4 Specification."

::= { pktcSigDevObjects 32 }

```
pktcSigDevToneEntry      OBJECT-TYPE
    SYNTAX                PkctlSigDevToneEntry
    MAX-ACCESS             not-accessible
    STATUS                 current
```

#### DESCRIPTION

" The different tone types that can be provisioned based on country-specific needs.

Each entry contains the tone generation parameters for a specific frequency group of the specific Tone Type.

The different parameters can be provisioned via MTA configuration based on country specific needs.  
 An MTA MUST populate all entries of this table for each tone type."

```
INDEX { pktcSigDevToneType, pktcSigDevToneFreqGroup }
 ::= { pktcSigDevToneTable 1 }
```

```
PktcSigDevToneEntry ::= SEQUENCE {
    pktcSigDevToneType          INTEGER,
    pktcSigDevToneFreqGroup     Unsigned32,
    pktcSigDevToneFreqCounter   Unsigned32,
    pktcSigDevToneWholeToneRepeatCount Unsigned32,
    pktcSigDevToneSteady        TruthValue
}
```

```
pktcSigDevToneType OBJECT-TYPE
```

```
SYNTAX      INTEGER {
    busy(1),
    confirmation(2),
    dial(3),
    messageWaiting(4),
    offHookWarning(5),
    ringBack(6),
    reOrder(7),
    stutterdial(8),
    callWaiting1(9),
    callWaiting2(10),
    callWaiting3(11),
    callWaiting4(12),
    alertingSignal(13),
    specialDial(14),
    specialInfo(15),
    release(16),
    congestion(17),
    userDefined1(18),
    userDefined2(19),
    userDefined3(20),
    userDefined4(21)
}
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"A unique value that will correspond to the different tone types. These tones can be provisioned based on country-specific needs. This object defines the type of tone being accessed.

The alertingSignal, specialDial, specialInfo, release,

```

        congestion, userDefined1, userDefined2, userDefined3,
        and userDefined4 tone types are used in
        the E line package."
 ::= { pktcSigDevToneEntry 1 }

```

```

pktcSigDevToneFreqGroup OBJECT-TYPE
    SYNTAX      Unsigned32(1..4)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This MIB object represents the Tone Sequence reference
        of a multi-sequence tone."
    ::= { pktcSigDevToneEntry 2}

```

```

pktcSigDevToneFreqCounter OBJECT-TYPE
    SYNTAX      Unsigned32(1..8)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This MIB object represents the number of consecutive
        multi-frequency tones for the particular tone type in
        the multi-frequency table (pktcSigDevMultiFreqToneTable).

        Such a sequence of the consecutive multi-frequency tones
        forms the tone group for the particular tone type in the
        pktcSigDevToneTable."
    ::= { pktcSigDevToneEntry 3}

```

```

pktcSigDevToneWholeToneRepeatCount      OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This is the repeat count, which signifies how many times
        to repeat the entire on-off cadence sequence. Setting this
        object may result in a cadence duration longer or shorter
        than the overall signal duration specified by the time out
        (TO) object for a particular signal. If the repeat count
        results in a longer tone duration than the signal duration
        specified by the TO, the tone duration defined by the
        TO object for a particular signal always represents
        the overall signal duration for a tone. In this case, the
        tone duration repeat count will not be fully exercised, and
        the desired tone duration will be truncated per the TO
        setting. If the repeat count results in a shorter tone
        duration than the signal duration specified by the TO, the
        tone duration defined by the repeat count takes precedence
        over the TO and will end the signal event. In this case,

```

the TO represents a time not to be exceeded for the signal.  
It is recommended to ensure proper telephony signaling so that  
the TO duration setting should always be longer than the  
desired repeat count-time duration."

```
::={ pktcSigDevToneEntry 4 }
```

pktcSigDevToneSteady OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object represents the steady tone status. A value  
of 'true(1)' indicates that the steady tone is applied, and  
a value of 'false(2)' indicates otherwise.

Devices must play out the on-off cadence sequence for  
the number of times indicated by the MIB object  
'pktcSigDevToneWholeToneRepeatCount' prior to applying the  
last tone steadily, indefinitely. If the MIB table  
'pktcSigDevToneTable' contains multiple rows with this  
Object set to a value of 'true(1)', the steady tone is  
applied to the last repeating frequency group of the tone.

Setting this MIB object may result in a tone duration that is  
longer or shorter than the overall signal duration  
specified by the time out (TO) MIB object for a particular  
signal. If the repeat count results in a longer tone  
duration than the signal duration specified by the TO, the  
tone duration defined by the TO object for a particular  
signal always represents the overall signal duration for a  
tone. In this case, the tone duration repeat count will  
not be fully exercised, and the desired tone duration will  
be truncated per the TO setting. If the repeat count  
results in a shorter tone duration than the signal duration  
specified by the TO, the tone duration defined by the  
repeat count takes precedence over the TO and will end the  
signal event. In this case, the TO represents a time not to  
be exceeded for the signal.

It is recommended to ensure proper telephony signaling that  
The TO duration setting should always be longer than the  
desired repeat count-time duration, plus the desired maximum  
steady tone period."

```
::={ pktcSigDevToneEntry 5 }
```

pktcSigDevMultiFreqToneTable OBJECT-TYPE

SYNTAX SEQUENCE OF PkctlSigDevMultiFreqToneEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

" This MIB table defines the characteristics of tones with multiple frequencies. The constraints imposed on the tones by the MIB table pktcSigDevToneTable need to be considered for MIB objects in this table as well.

The MTA MUST populate the corresponding row(s) of the pktcSigDevMultiFreqToneTable for each tone defined in the pktcSigDevToneTable.

The contents of the table may be provisioned via MTA configuration."

## REFERENCE

"PacketCable NCS Specification, ETSI-TS-101-909-4 Specification."

::= { pktcSigDevObjects 33 }

pktcSigDevMultiFreqToneEntry OBJECT-TYPE

SYNTAX PktcSigDevMultiFreqToneEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

" The different tone types with multiple frequencies that can be provisioned based on country-specific needs."

INDEX {pktcSigDevToneType, pktcSigDevToneNumber}

::= { pktcSigDevMultiFreqToneTable 1 }

PktcSigDevMultiFreqToneEntry ::= SEQUENCE {

pktcSigDevToneNumber	Unsigned32,
pktcSigDevToneFirstFreqValue	Unsigned32,
pktcSigDevToneSecondFreqValue	Unsigned32,
pktcSigDevToneThirdFreqValue	Unsigned32,
pktcSigDevToneFourthFreqValue	Unsigned32,
pktcSigDevToneFreqMode	INTEGER,
pktcSigDevToneFreqAmpModePrtg	Unsigned32,
pktcSigDevToneDbLevel	TenthdBm,
pktcSigDevToneFreqOnDuration	Unsigned32,
pktcSigDevToneFreqOffDuration	Unsigned32,
pktcSigDevToneFreqRepeatCount	Unsigned32

}

pktcSigDevToneNumber OBJECT-TYPE

SYNTAX Unsigned32(1..8)

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This MIB object represents the frequency reference  
of a multi-frequency tone."  
::={ pktcSigDevMultiFreqToneEntry 1}

pktcSigDevToneFirstFreqValue      OBJECT-TYPE  
SYNTAX            Unsigned32(0..4000)  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This MIB object represents the value of the first  
    frequency of a tone type. A value of zero implies  
    absence of the referenced frequency."  
::={ pktcSigDevMultiFreqToneEntry 2}

pktcSigDevToneSecondFreqValue      OBJECT-TYPE  
SYNTAX            Unsigned32(0..4000)  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This MIB object represents the value of the second  
    frequency of a tone type. A value of zero implies  
    absence of the referenced frequency."  
::={ pktcSigDevMultiFreqToneEntry 3}

pktcSigDevToneThirdFreqValue      OBJECT-TYPE  
SYNTAX            Unsigned32(0..4000)  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This MIB object represents the value of the third  
    frequency of a tone type. A value of zero implies  
    absence of the referenced frequency."  
::={ pktcSigDevMultiFreqToneEntry 4}

pktcSigDevToneFourthFreqValue      OBJECT-TYPE  
SYNTAX            Unsigned32(0..4000)  
MAX-ACCESS       read-only  
STATUS            current  
DESCRIPTION  
    "This MIB object represents the value of the fourth  
    frequency of a tone type. A value of zero implies  
    absence of the referenced frequency."  
::={ pktcSigDevMultiFreqToneEntry 5}

pktcSigDevToneFreqMode OBJECT-TYPE  
SYNTAX            INTEGER {  
                  firstModulatedBySecond(1),  
                  summation(2)



```

    }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION

```

"This MIB object provides directive on the modulation or summation of the frequencies involved in the tone.

It is to be noted that while summation can be done without any constraint on the number of frequencies, the modulation (amplitude) holds good only when there are two frequencies (first and second).

Thus:

- If the mode is set to a value of 'firstModulatedBySecond(1)', the first frequency MUST be modulated by the second, and the remaining frequencies (third and fourth) ignored. The percentage of amplitude modulation to be applied is defined by the MIB object pktcSigDevToneFreqAmpModePrtg.
- If the mode is set to a value of 'summation(2)', all the frequencies MUST be summed without any modulation.

```

"
 ::= { pktcSigDevMultiFreqToneEntry 6}

```

pktcSigDevToneFreqAmpModePrtg OBJECT-TYPE

```

SYNTAX          Unsigned32(0..100)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION

```

"This MIB object represents the percentage of amplitude modulation applied to the second frequency when the MIB object pktcSigDevToneFreqMode is set to a value of 'firstModulatedBySecond (1)'.

If the MIB object pktcSigDevToneFreqMode is set to value of 'summation (2)', then this MIB object MUST be ignored."

```

 ::= { pktcSigDevMultiFreqToneEntry 7}

```

pktcSigDevToneDbLevel OBJECT-TYPE

```

SYNTAX          TenthdBm (-250..-110)
UNITS           "1/10 of a dBm"
MAX-ACCESS      read-only

```

STATUS current

DESCRIPTION

"This MIB object contains the decibel level for each analog signal (tone) that is locally generated (versus in-band supervisory tones) and sourced to the a-b terminals (TE connection point). Each tone in itself may consist of multiple frequencies, as defined by the MIB table pktcSigDevMultiFreqToneTable.

This MIB object reflects the desired level at the Telco (POTS) a-b (T/R) terminals, including the effect of any MTA receiver gain (loss). This is required so that locally generated tones are consistent with remotely generated in-band tones at the a-b terminals, consistent with user expectations.

This MIB object must be set for each tone. When tones are formed by combining multi-frequencies, the level of each frequency shall be set so as to result in the tone level specified in this object at the a-b (T/R) terminals.

The wide range of levels for this Object is required to provide signal-generator levels across the wide range of gains (losses) -- but does not imply the entire range is to be achievable given the range of gains (losses) in the MTA."

DEFVAL { -120 }

::={ pktcSigDevMultiFreqToneEntry 8}

pktcSigDevToneFreqOnDuration OBJECT-TYPE

SYNTAX Unsigned32(0..5000)

UNITS "milliseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object represents the duration for which the frequency reference corresponding to the tone type is turned on."

::={ pktcSigDevMultiFreqToneEntry 9}

pktcSigDevToneFreqOffDuration OBJECT-TYPE

SYNTAX Unsigned32(0..5000)

UNITS "milliseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This MIB object represents the duration for which the

frequency reference corresponding to the tone type  
is turned off."  
::={ pktcSigDevMultiFreqToneEntry 10}

#### pktcSigDevToneFreqRepeatCount OBJECT-TYPE

SYNTAX Unsigned32(0..5000)

MAX-ACCESS read-only

STATUS current

#### DESCRIPTION

"This MIB object indicates the number of times to repeat the cadence cycle represented by the on/off durations (refer to the MIB objects pktcSigDevToneFreqOnDuration and pktcSigDevToneFreqOffDuration).

Setting this object may result in a tone duration that is longer or shorter than the overall signal duration specified by the time out (TO) object for the corresponding tone type. If the value of this MIB Object indicates a longer duration than that specified by the TO, the latter overrules the former, and the desired tone duration will be truncated according to the TO.

However, if the repeat count results in a shorter tone duration than the signal duration specified by the TO, the tone duration defined by the repeat count takes precedence over the TO and will end the signal event. In this case, the TO represents a time not to be exceeded for the signal. It is recommended, to ensure proper telephony signaling, that the TO duration setting should always be longer than the desired repeat count-time duration. A value of zero means the tone sequence is to be played once but not repeated."

::={ pktcSigDevMultiFreqToneEntry 11}

#### pktcSigDevCidDelayAfterLR OBJECT-TYPE

SYNTAX Unsigned32 (300..800)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

"This object specifies the delay between the end of the Line Reversal and the start of the FSK or DTMF signal. This MIB object is used only when pktcSigDevCidMode is set to a value of 'lrETS'. This timing has a range of 300 to 800 ms.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	any value (not used)
rpAsETS	any value (not used)
lrAsETS	any value (not used)
lrETS	400

An attempt to set this object while the value of pktcSigDevCidMode is not set to a value of 'lrETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

```
DEFVAL { 400 }
::= {pktcSigDevObjects 34 }
```

#### pktcSigDevCidDtmfStartCode OBJECT-TYPE

```
SYNTAX      DtmfCode
MAX-ACCESS  read-write
STATUS      current
```

#### DESCRIPTION

"This object identifies optional start codes used when the MIB object pktcSigDevCidSigProtocol is set to a value of 'dtmf(2)'.

Different countries define different caller id signaling codes to support caller identification. When Dual-Tone Multi-Frequency (DTMF) is used, the caller id digits are preceded by a 'start code' digit, followed by the digit transmission sequence <S1>...<Sn> (where Sx represents the digits 0-9), and terminated by the 'end code' digit.

For example,

```
<A><S1>...<Sn> <D><S1>...<Sn> <B><S1>...<Sn> <C>.
```

The start code for calling number delivery may be DTMF 'A' or 'D'. The start code for redirecting a number may be DTMF 'D'. The DTMF code 'B' may be sent by the network as a start code for the transfer of information values, through which special events can be indicated to the user. In some countries, the '\*' or '#' may be used instead of 'A', 'B', 'C', or 'D'.

The value of this MIB object MUST NOT persist across MTA

```

        reboots."
REFERENCE
    "ETSI-EN-300-659-1 specification"
DEFVAL {dtmfcodeA}
 ::= { pktcSigDevObjects 35 }

pktcSigDevCidDtmfEndCode OBJECT-TYPE
    SYNTAX      DtmfCode
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This object identifies optional end codes used when the
        pktcSigDevCidSigProtocol is set to a value of
        'dtmf(2)'."

        Different countries define different caller id signaling
        protocols to support caller identification.  When
        Dual-Tone Multi-Frequency (DTMF) is used, the caller id
        digits are preceded by a 'start code' digit, followed by
        the digit transmission sequence <S1>...<Sn> (where Sx
        represents the digits 0-9), and terminated by the 'end
        code' digit.

        For example,
            <A><S1>...<Sn> <D><S1>...<Sn> <B><S1>...<Sn> <C>.

        The DTMF code 'C' may be sent by the network as an
        end code for the transfer of information values, through
        which special events can be indicated to the user.  In
        some countries, the '*' or '#' may be used instead of
        'A', 'B', 'C', or 'D'.

        The value of this MIB object MUST NOT persist across MTA
        reboots."
REFERENCE
    "ETSI-EN-300-659-1 specification"
DEFVAL {dtmfcodeC}
 ::= { pktcSigDevObjects 36 }

pktcSigDevVmwISigProtocol OBJECT-TYPE
    SYNTAX      PkctSubscriberSideSigProtocol
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "This object identifies the subscriber line protocol used
        for signaling the information on Visual Message Waiting
        Indicator (VMWI).  Different countries define different
        VMWI signaling protocols to support VMWI service."

```

Frequency shift keying (FSK) is most commonly used.  
DTMF is an alternative.

The value of this MIB object MUST NOT persist across MTA reboots."

```
DEFVAL { fsk }
 ::= { pktcSigDevObjects 37 }
```

```
pktcSigDevVmwiDelayAfterLR      OBJECT-TYPE
    SYNTAX      Unsigned32 (0|300..800)
    UNITS       "Milliseconds"
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
```

"This object specifies the delay between the end of the Line Reversal and the start of the FSK or DTMF signal. This object is only used when pktcSigDevVmwiMode is set to a value of 'lrETS'. This timing has a range of 300 to 800 ms.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevVmwiMode), and MUST be followed:

Value of pktcSigDevVmwiMode	Default value
duringringingETS	any value (not used)
dtAsETS	any value (not used)
rpAsETS	any value (not used)
lrAsETS	any value (not used)
lrETS	400

An attempt to set this object while the value of pktcSigDevVmwiMode is not 'lrETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

```
DEFVAL {400}
 ::= {pktcSigDevObjects 38 }
```

```
pktcSigDevVmwiDtmfStartCode OBJECT-TYPE
    SYNTAX      DtmfCode
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
```

"This object identifies optional start codes used when

the pktcSigDevVmwiSigProtocol is set to a value of 'dtmf(2)'. Different countries define different On Hook Data Transmission Protocol signaling codes to support VMWI.

When Dual-Tone Multi-Frequency (DTMF) is used, the VMWI digits are preceded by a 'start code' digit, followed by the digit transmission sequence <S1>...<Sn> (where Sx represents the digits 0-9), and terminated by the 'end code' digit.

For example,

<A><S1>...<Sn> <D><S1>...<Sn> <B><S1>...<Sn> <C>.

The start code for redirecting VMWI may be DTMF 'D'. The DTMF code 'B' may be sent by the network as a start code for the transfer of information values, through which special events can be indicated to the user. In some countries, the '\*' or '#' may be used instead of 'A', 'B', 'C', or 'D'.

The value of this MIB object MUST NOT persist across MTA reboots."

#### REFERENCE

"ETSI-EN-300-659-1 specification"

DEFVAL {dtmfcodeA}

::= { pktcSigDevObjects 39 }

#### pktcSigDevVmwiDtmfEndCode OBJECT-TYPE

SYNTAX DtmfCode

MAX-ACCESS read-write

STATUS current

#### DESCRIPTION

"This object identifies an optional end code used when the pktcSigDevVmwiSigProtocol is set to a value of 'dtmf(2)'. Different countries define different on-hook Data Transmission Protocol signaling codes to support VMWI.

When Dual-Tone Multi-Frequency (DTMF) is used, the VMWI digits are preceded by a 'start code' digit, followed by the digit transmission sequence <S1>...<Sn> (where Sx represents the digits 0-9), and terminated by the 'end code' digit.

For example,

<A><S1>...<Sn> <D><S1>...<Sn> <B><S1>...<Sn> <C>.

The DTMF code 'C' may be sent by the network as an end code for the transfer of information values, through which special events can be indicated to the user. In some countries, the '\*' or '#' may be used instead of 'A', 'B', 'C', or 'D'.

The value of this MIB object MUST NOT persist across MTA reboots."

## REFERENCE

"ETSI-EN-300-659-1 specification"

DEFVAL {dtmfcodeC}

::= { pktcSigDevObjects 40 }

pktcSigDevrpAsDtsDuration      OBJECT-TYPE  
SYNTAX            Unsigned32 (0|200..500)  
UNITS             "Milliseconds"  
MAX-ACCESS       read-write  
STATUS            current  
DESCRIPTION

" This object specifies the duration of the rpASDTS ring pulse prior to the start of the transmission of the FSK or DTMF containing the caller id information. It is only used when pktcSigDevCidMode is set to a value of 'rpAsETS'.

The following table defines the default values for this MIB object, depending on the signal type (pktcSigDevCidMode), and MUST be followed:

Value of pktcSigDevCidMode	Default value
duringringingETS	any value (not used)
dtAsETS	any value (not used)
rpAsETS	250
lrAsETS	any value (not used)
lrETS	any value (not used)

An attempt to set this object while the value of pktcSigDevCidMode is not 'rpAsETS' will result in an 'inconsistentValue' error.

The value of this MIB object MUST NOT persist across MTA reboots."

## REFERENCE

"ETSI-EN-300-659-1 Specification and Belgacom  
BGC\_D\_48\_9811\_30\_09\_EDOC version 3.3"

DEFVAL { 250 }

::= { pktcSigDevObjects 41 }



```
--
-- The Endpoint Config Table is used to define attributes that
-- are specific to connection EndPoints.
--

pktcSigEndPntConfigTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PktcSigEndPntConfigEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        " This table describes the information pertaining to each
        endpoint of the MTA. All entries in this table represent
        the provisioned endpoints provisioned with the information
        required by the MTA to maintain the NCS protocol
        communication with the CMS. Each endpoint can be assigned
        to its own CMS. If the specific endpoint does not have
        the corresponding CMS information in this table, the
        endpoint is considered as not provisioned with voice
        services. Objects in this table do not persist across
        MTA reboots."
    ::= { pktcSigEndPntConfigObjects 1 }

pktcSigEndPntConfigEntry OBJECT-TYPE
    SYNTAX          PktcSigEndPntConfigEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Each entry in the pktcSigEndPntConfigTable represents
        required signaling parameters for the specific endpoint
        provisioned with voice services. The conceptual rows MUST
        NOT persist across MTA reboots."
    INDEX { ifIndex }
    ::= { pktcSigEndPntConfigTable 1 }

PktcSigEndPntConfigEntry ::= SEQUENCE {
    pktcSigEndPntConfigCallAgentId          SnmpAdminString,
    pktcSigEndPntConfigCallAgentUdpPort    InetPortNumber,
    pktcSigEndPntConfigPartialDialTO       Unsigned32,
    pktcSigEndPntConfigCriticalDialTO      Unsigned32,
    pktcSigEndPntConfigBusyToneTO          Unsigned32,
    pktcSigEndPntConfigDialToneTO          Unsigned32,
    pktcSigEndPntConfigMessageWaitingTO    Unsigned32,
    pktcSigEndPntConfigOffHookWarnToneTO   Unsigned32,
    pktcSigEndPntConfigRingingTO           Unsigned32,
    pktcSigEndPntConfigRingBackTO          Unsigned32,
    pktcSigEndPntConfigReorderToneTO       Unsigned32,
    pktcSigEndPntConfigStutterDialToneTO   Unsigned32,
```

```

pktcSigEndPntConfigTSMMax      Unsigned32,
pktcSigEndPntConfigMax1        Unsigned32,
pktcSigEndPntConfigMax2        Unsigned32,
pktcSigEndPntConfigMax1QEnable TruthValue,
pktcSigEndPntConfigMax2QEnable TruthValue,
pktcSigEndPntConfigMWD         Unsigned32,
pktcSigEndPntConfigTdinit      Unsigned32,
pktcSigEndPntConfigTdmin       Unsigned32,
pktcSigEndPntConfigTdmax       Unsigned32,
pktcSigEndPntConfigRtoMax      Unsigned32,
pktcSigEndPntConfigRtoInit     Unsigned32,
pktcSigEndPntConfigLongDurationKeepAlive Unsigned32,
pktcSigEndPntConfigThist       Unsigned32,
pktcSigEndPntConfigStatus      RowStatus,
pktcSigEndPntConfigCallWaitingMaxRep Unsigned32,
pktcSigEndPntConfigCallWaitingDelay Unsigned32,
pktcSigEndPntStatusCallIpAddressType InetAddressType,
pktcSigEndPntStatusCallIpAddress InetAddress,
pktcSigEndPntStatusError       INTEGER,
pktcSigEndPntConfigMinHookFlash Unsigned32,
pktcSigEndPntConfigMaxHookFlash Unsigned32,
pktcSigEndPntConfigPulseDialInterdigitTime Unsigned32,
pktcSigEndPntConfigPulseDialMinMakeTime Unsigned32,
pktcSigEndPntConfigPulseDialMaxMakeTime Unsigned32,
pktcSigEndPntConfigPulseDialMinBreakTime Unsigned32,
pktcSigEndPntConfigPulseDialMaxBreakTime Unsigned32
}

```

pktcSigEndPntConfigCallAgentId      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@example.com). The call agent name, after the character '@', MUST be a fully qualified domain name (FQDN) and MUST have a corresponding pktcMtaDevCmsFqdn entry in the pktcMtaDevCmsTable. The object pktcMtaDevCmsFqdn is defined in the PacketCable MIBMTA Specification. For each particular endpoint, 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. 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 using SNMP during normal operation."

```
::= { pktcSigEndPntConfigEntry 1 }
```

```
pktcSigEndPntConfigCallAgentUdpPort      OBJECT-TYPE
```

```
SYNTAX      InetPortNumber (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 from the endpoint. For each particular endpoint, 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 the 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 using SNMP during normal operation."

```
REFERENCE
```

"PacketCable NCS Specification"

```
DEFVAL      { 2727 }
```

```
::= { pktcSigEndPntConfigEntry 2 }
```

```
pktcSigEndPntConfigPartialDialTO         OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS       "seconds"
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This object contains the value of the partial dial time out. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

```
REFERENCE
```

"PacketCable NCS Specification"

```
DEFVAL { 16 }
```

```
::= { pktcSigEndPntConfigEntry 3 }
```

```
pktcSigEndPntConfigCriticalDialTO        OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS       "seconds"
```

MAX-ACCESS read-create  
 STATUS current  
 DESCRIPTION

"This object contains the value of the critical dial time out. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

REFERENCE  
 "PacketCable NCS Specification"  
 DEFVAL { 4 }  
 ::= { pktcSigEndPntConfigEntry 4 }

pktcSigEndPntConfigBusyToneTO OBJECT-TYPE

SYNTAX Unsigned32  
 UNITS "seconds"  
 MAX-ACCESS read-create  
 STATUS current  
 DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

REFERENCE  
 "PacketCable NCS Specification"  
 DEFVAL { 30 }  
 ::= { pktcSigEndPntConfigEntry 5 }

pktcSigEndPntConfigDialToneTO OBJECT-TYPE

SYNTAX Unsigned32  
 UNITS "seconds"  
 MAX-ACCESS read-create  
 STATUS current  
 DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcSigEndPntConfigEntry 6 }

pktcSigEndPntConfigMessageWaitingTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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 Manager application, the MTA MUST use the new value as a default only for a new signal requested by the NCS message.

The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcSigEndPntConfigEntry 7 }

pktcSigEndPntConfigOffHookWarnToneTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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 Manager

application, the MTA MUST use the new value as a default only for a new signal requested by the NCS message. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 0 }

::= { pktcSigEndPntConfigEntry 8 }

pktcSigEndPntConfigRingingTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 180 }

::= { pktcSigEndPntConfigEntry 9 }

pktcSigEndPntConfigRingBackTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time

a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 180 }

::= { pktcSigEndPntConfigEntry 10 }

pktcSigEndPntConfigReorderToneTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 30 }

::= { pktcSigEndPntConfigEntry 11 }

pktcSigEndPntConfigStutterDialToneTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

" This object contains the default time out 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. The time out (TO) elements are intended to limit the time a tone or frequency is generated. When this MIB object is set to a value of '0', the MTA MUST NOT generate the

corresponding frequency or tone, regardless of the definitions pertaining to frequency, tone duration, or cadence."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcSigEndPntConfigEntry 12 }

pktcSigEndPntConfigTSMMax OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This MIB object is used as part of an NCS retransmission algorithm. Prior to any retransmission, the MTA must check to make sure that the time elapsed since the sending of the initial datagram does not exceed the value specified by this MIB object. If more than Tsmx time has elapsed, then the retransmissions MUST cease.

Refer to the MIB object pktcSigEndPntConfigThist for information on when the endpoint becomes disconnected."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 20 }

::= { pktcSigEndPntConfigEntry 13 }

pktcSigEndPntConfigMax1 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object contains the suspicious error threshold for signaling messages. The pktcSigEndPntConfigMax1 object indicates the retransmission threshold at which the MTA MAY actively query the domain name server (DNS) in order to detect the possible change of call agent interfaces."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 5 }

::= { pktcSigEndPntConfigEntry 14 }

pktcSigEndPntConfigMax2 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

## DESCRIPTION



"This object contains the disconnect error threshold for signaling messages. The pktcSigEndPntConfigMax2 object indicates the retransmission threshold at which the MTA SHOULD contact the DNS one more time to see if any other interfaces to the call agent have become available."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 7 }

::= { pktcSigEndPntConfigEntry 15 }

pktcSigEndPntConfigMax1QEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object enables/disables the Max1 domain name server (DNS) query operation when the pktcSigEndPntConfigMax1 threshold has been reached.

A value of true(1) indicates enabling, and a value of false(2) indicates disabling."

DEFVAL { true }

::= { pktcSigEndPntConfigEntry 16 }

pktcSigEndPntConfigMax2QEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"This object enables/disables the Max2 domain name server (DNS) query operation when the pktcSigEndPntConfigMax2 threshold has been reached.

A value of true(1) indicates enabling, and a value of false(2) indicates disabling."

DEFVAL { true }

::= { pktcSigEndPntConfigEntry 17 }

pktcSigEndPntConfigMWD OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"Maximum Waiting Delay (MWD) contains the maximum number of seconds an MTA waits, after powering on, before initiating the restart procedure with the call agent."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 600 }

```
::= { pktcSigEndPntConfigEntry 18 }
```

```
pktcSigEndPntConfigTdinit      OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS       "seconds"
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This MIB object represents the 'disconnected' initial waiting delay within the context of an MTA's 'disconnected procedure'. The 'disconnected procedure' is initiated when an endpoint becomes 'disconnected' while attempting to communicate with a call agent.

The 'disconnected timer' associated with the 'disconnected Procedure' is initialized to a random value, uniformly distributed between zero and the value contained in this MIB object.

For more information on the usage of this timer, please refer to the PacketCable NCS Specification."

```
REFERENCE
```

"PacketCable NCS Specification"

```
DEFVAL { 15 }
```

```
::= { pktcSigEndPntConfigEntry 19 }
```

```
pktcSigEndPntConfigTdmin      OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
UNITS       "seconds"
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This MIB object represents the 'disconnected' minimum waiting delay within the context of an MTA's 'disconnected procedure', specifically when local user activity is detected.

The 'disconnected procedure' is initiated when an endpoint becomes 'disconnected' while attempting to communicate with a call agent.

For more information on the usage of this timer, please refer to the PacketCable NCS Specification."

```
REFERENCE
```

"PacketCable NCS Specification"

```
DEFVAL { 15 }
```

```
::= { pktcSigEndPntConfigEntry 20 }
```

```
pktcSigEndPntConfigTdmax      OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" This object contains the maximum number of seconds the MTA waits, after a disconnect, before initiating the disconnected procedure with the call agent.

"

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 600 }

::= { pktcSigEndPntConfigEntry 21 }

pktcSigEndPntConfigRtoMax OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object specifies the maximum number of seconds the MTA waits for a response to an NCS message before initiating a retransmission."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 4 }

::= { pktcSigEndPntConfigEntry 22 }

pktcSigEndPntConfigRtoInit OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" This object contains the initial number of seconds for the retransmission timer."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 200 }

::= { pktcSigEndPntConfigEntry 23 }

pktcSigEndPntConfigLongDurationKeepAlive OBJECT-TYPE

SYNTAX Unsigned32

UNITS "minutes"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" Specifies a time out value, in minutes, for sending long duration call notification messages."

## REFERENCE

"PacketCable NCS Specification"

DEFVAL { 60 }

::= { pktcSigEndPntConfigEntry 24 }

pktcSigEndPntConfigThist OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" Time out period, in seconds, before no response is declared."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 30 }

::= { pktcSigEndPntConfigEntry 25 }

pktcSigEndPntConfigStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" This object contains the Row Status associated with the pktcSigEndPntConfigTable. There are no restrictions or dependencies amidst the columnar objects before this row can be activated or for modifications of the columnar objects when this object is set to a value of 'active(1)."

::= { pktcSigEndPntConfigEntry 26 }

pktcSigEndPntConfigCallWaitingMaxRep OBJECT-TYPE

SYNTAX Unsigned32 (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 Manager application, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

DEFVAL { 1 }

::= { pktcSigEndPntConfigEntry 27 }

pktcSigEndPntConfigCallWaitingDelay OBJECT-TYPE

SYNTAX Unsigned32 (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 }
 ::= { pktcSigEndPntConfigEntry 28 }

```

```

pktcSigEndPntStatusCallIpAddressType  OBJECT-TYPE
SYNTAX          InetAddressType
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION
    " This object contains the type of Internet address contained
      in the MIB object 'pktcSigEndPntStatusCallIpAddress'.

      Since pktcSigEndPntStatusCallIpAddress is expected to
      contain an IP address, a value of dns(16) is disallowed."

 ::= { pktcSigEndPntConfigEntry 29 }

```

```

pktcSigEndPntStatusCallIpAddress  OBJECT-TYPE
SYNTAX          InetAddress
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION
    " This MIB object contains the chosen IP address of the CMS
      currently being used for the corresponding endpoint.

      The device determines the IP address by using DNS to
      resolve the IP address of the CMS from the FQDN stored in
      the MIB object 'pktcSigEndPntConfigCallAgentId'. The
      processes are outlined in the PacketCable NCS and Security
      specifications, and MUST be followed by the MTA.

      The IP address type contained in this MIB object is
      indicated by pktcSigEndPntStatusCallIpAddressType."
REFERENCE
    "PacketCable NCS Specification;
      PacketCable Security specification, [PKT-SP-SEC]."
 ::= { pktcSigEndPntConfigEntry 30 }

```

```

pktcSigEndPntStatusError  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 status indicates that all operations
      necessary to put the line in service have occurred, and the
      CMS has acknowledged the Restart In Progress (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.  If pktcMtaDevCmsIpsecCtrl is
      disabled for the associated call agent, the
      noSecurityAssociation status is not applicable and should
      not be used by the MTA.  The disconnected status indicates
      one of the following two:
      If pktcMtaDevCmsIpsecCtrl is disabled, then no security
      association is involved with this endpoint.  The NCS
      signaling software is in process of establishing the NCS
      signaling link via an RSIP exchange.
      Otherwise, when pktcMtaDevCmsIpsecCtrl is enabled,
      security Association has been established, and the NCS
      signaling software is in process of establishing the NCS
      signaling link via an RSIP exchange."
 ::= { pktcSigEndPntConfigEntry 31 }

pktcSigEndPntConfigMinHookFlash    OBJECT-TYPE
SYNTAX        Unsigned32 (20..1550)
UNITS         "Milliseconds"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    " This is the minimum time a line needs to be on-hook for a
      valid hook flash.  The value of this object MUST be
      greater than the value of
      pktcSigEndPntConfigPulseDialMaxBreakTime.  The value of
      pktcSigEndPntConfigMinHookFlash MUST be less than
      pktcSigEndPntConfigMaxHookFlash.  This object MUST only be
      set via the MTA configuration during the provisioning
      process.
      Furthermore, given the possibility for the 'pulse dial'
      and 'hook flash' to overlap, the value of this object
      MUST be greater than the value contained by the MIB
      Object 'pktcSigEndPntConfigPulseDialMaxMakeTime'."
DEFVAL { 300 }
 ::= { pktcSigEndPntConfigEntry 32 }

```

```

pktcSigEndPntConfigMaxHookFlash      OBJECT-TYPE
    SYNTAX      Unsigned32 (20..1550)
    UNITS        "Milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " This is the maximum time a line needs to be on-hook for a
          valid hook flash.  The value of
          pktcSigEndPntConfigMaxHookFlash MUST be greater than
          pktcSigEndPntConfigMinHookFlash.  This object MUST only be
          set via the MTA configuration during the provisioning
          process."
    DEFVAL { 800 }
    ::= { pktcSigEndPntConfigEntry 33 }

pktcSigEndPntConfigPulseDialInterdigitTime      OBJECT-TYPE
    SYNTAX      Unsigned32 (100..1500)
    UNITS        "Milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " This is the pulse dial inter-digit time out.  This object
          MUST only be set via the MTA configuration during the
          provisioning process."
    DEFVAL { 100 }
    ::= { pktcSigEndPntConfigEntry 34 }

pktcSigEndPntConfigPulseDialMinMakeTime      OBJECT-TYPE
    SYNTAX      Unsigned32 (20..200)
    UNITS        "Milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " This is the minimum make pulse width for the dial pulse.
          The value of pktcSigEndPntConfigPulseDialMinMakeTime MUST
          be less than pktcSigEndPntConfigPulseDialMaxMakeTime.  This
          object MUST only be set via the MTA configuration during
          the provisioning process."
    DEFVAL { 25 }
    ::= { pktcSigEndPntConfigEntry 35 }

pktcSigEndPntConfigPulseDialMaxMakeTime      OBJECT-TYPE
    SYNTAX      Unsigned32 (20..200)
    UNITS        "Milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        " This is the maximum make pulse width for the dial pulse.

```

The value of pktcSigEndPntConfigPulseDialMaxMakeTime MUST be greater than pktcSigEndPntConfigPulseDialMinMakeTime. This object MUST only be provided via the configuration file during the provisioning process. Furthermore, given the possibility for the 'pulse dial' and 'hook flash' to overlap, the value of this object MUST be less than the value contained by the MIB object pktcSigEndPntConfigMinHookFlash."

```
DEFVAL { 55 }
 ::= { pktcSigEndPntConfigEntry 36 }
```

pktcSigEndPntConfigPulseDialMinBreakTime OBJECT-TYPE

SYNTAX Unsigned32 (20..200)

UNITS "Milliseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This is the minimum break pulse width for the dial pulse. The value of pktcSigEndPntConfigPulseDialMinBreakTime MUST be less than pktcSigEndPntConfigPulseDialMaxBreakTime. This object must only be provided via the configuration file during the provisioning process."

```
DEFVAL { 45 }
 ::= { pktcSigEndPntConfigEntry 37 }
```

pktcSigEndPntConfigPulseDialMaxBreakTime OBJECT-TYPE

SYNTAX Unsigned32 (20..200)

UNITS "Milliseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This is the maximum break pulse width for the dial pulse. The value of pktcSigEndPntConfigPulseDialMaxBreakTime MUST be greater than pktcSigEndPntConfigPulseDialMinBreakTime. This object MUST only be provided via the configuration file during the provisioning process."

```
DEFVAL { 75 }
 ::= { pktcSigEndPntConfigEntry 38 }
```

--

-- notification group is for future extension.

--

```
pktcSigNotification OBJECT IDENTIFIER ::= { pktcIetfSigMib 0 }
pktcSigConformance OBJECT IDENTIFIER ::= { pktcIetfSigMib 2 }
pktcSigCompliances OBJECT IDENTIFIER ::= { pktcSigConformance 1 }
pktcSigGroups OBJECT IDENTIFIER ::= { pktcSigConformance 2 }
```

--



```
-- compliance statements
--

pktcSigBasicCompliance  MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        " The compliance statement for MTAs that implement
          NCS signaling."

MODULE  -- pktcIetfSigMib

---
-- Unconditionally mandatory groups for all MTAs
---

MANDATORY-GROUPS {
    pktcSigDeviceGroup,
    pktcSigEndpointGroup
}
---
-- Conditionally mandatory groups for MTAs
---

GROUP pktcInternationalGroup
    DESCRIPTION
        " This group is mandatory only for MTAs implementing
          international telephony features."

GROUP pktcLLinePackageGroup
    DESCRIPTION
        " This group is mandatory only for MTAs implementing the L
          line package."

GROUP pktcELinePackageGroup
    DESCRIPTION
        " This group is mandatory only for MTAs implementing the E
          Line Package."
    ::= { pktcSigCompliances 1 }

pktcSigDeviceGroup  OBJECT-GROUP
    OBJECTS {
        pktcSigDevCodecMax,
        pktcSigDevEchoCancellation,
        pktcSigDevSilenceSuppression,
        pktcSigDevR0Cadence,
        pktcSigDevR1Cadence,
        pktcSigDevR2Cadence,
        pktcSigDevR3Cadence,
```

```

pktcSigDevR4Cadence,
pktcSigDevR5Cadence,
pktcSigDevR6Cadence,
pktcSigDevR7Cadence,
pktcSigDevRgCadence,
pktcSigDevRsCadence,
pktcSigDefCallSigDscp,
pktcSigDefMediaStreamDscp,
pktcSigDevVmwMode,
pktcSigCapabilityType,
pktcSigCapabilityVersion,
pktcSigCapabilityVendorExt,
pktcSigDefNcsReceiveUdpPort
}

```

STATUS current

DESCRIPTION

"Group of MIB objects containing signaling configuration information that is applicable per-device."

```
 ::= { pktcSigGroups 1 }
```

pktcSigEndpointGroup OBJECT-GROUP

```

OBJECTS {
pktcSigEndPntConfigCallAgentId,
pktcSigEndPntConfigCallAgentUdpPort,
pktcSigEndPntConfigPartialDialTO,
pktcSigEndPntConfigCriticalDialTO,
pktcSigEndPntConfigBusyToneTO,
pktcSigEndPntConfigDialToneTO,
pktcSigEndPntConfigMessageWaitingTO,
pktcSigEndPntConfigOffHookWarnToneTO,
pktcSigEndPntConfigRingingTO,
pktcSigEndPntConfigRingBackTO,
pktcSigEndPntConfigReorderToneTO,
pktcSigEndPntConfigStutterDialToneTO,
pktcSigEndPntConfigTSMAX,
pktcSigEndPntConfigMax1,
pktcSigEndPntConfigMax2,
pktcSigEndPntConfigMax1QEnable,
pktcSigEndPntConfigMax2QEnable,
pktcSigEndPntConfigMWD,
pktcSigEndPntConfigTdinit,
pktcSigEndPntConfigTdmin,
pktcSigEndPntConfigTdmax,
pktcSigEndPntConfigRtoMax,
pktcSigEndPntConfigRtoInit,
pktcSigEndPntConfigLongDurationKeepAlive,
pktcSigEndPntConfigThist,
pktcSigEndPntConfigStatus,

```

```

pktcSigEndPntConfigCallWaitingMaxRep,
pktcSigEndPntConfigCallWaitingDelay,
pktcSigEndPntStatusCallIpAddressType,
pktcSigEndPntStatusCallIpAddress,
pktcSigEndPntStatusError
}
STATUS current
DESCRIPTION
    "Group of MIB objects containing signaling configuration
    information that is applicable per-endpoint."
 ::= { pktcSigGroups 2 }

```

```

pktcInternationalGroup      OBJECT-GROUP
OBJECTS {
    pktcSigEndPntConfigMinHookFlash,
    pktcSigEndPntConfigMaxHookFlash,
    pktcSigEndPntConfigPulseDialInterdigitTime,
    pktcSigEndPntConfigPulseDialMinMakeTime,
    pktcSigEndPntConfigPulseDialMaxMakeTime,
    pktcSigEndPntConfigPulseDialMinBreakTime,
    pktcSigEndPntConfigPulseDialMaxBreakTime,
    pktcSigDevRingCadence,
    pktcSigDevCidSigProtocol,
    pktcSigDevCidDelayAfterLR,
    pktcSigDevCidDtmfStartCode,
    pktcSigDevCidDtmfEndCode,
    pktcSigDevVmwISigProtocol,
    pktcSigDevVmwIDelayAfterLR,
    pktcSigDevVmwIDtmfStartCode,
    pktcSigDevVmwIDtmfEndCode,
    pktcSigDevvpAsDtsDuration,
    pktcSigDevCidMode,
    pktcSigDevCidAfterRing,
    pktcSigDevCidAfterDTAS,
    pktcSigDevCidAfterRPAS,
    pktcSigDevRingAfterCID,
    pktcSigDevCidDTASAfterLR,
    pktcSigDevVmwIMode,
    pktcSigDevVmwIAfterDTAS,
    pktcSigDevVmwIAfterRPAS,
    pktcSigDevVmwIDTASAfterLR,
    pktcSigPowerRingFrequency,
    pktcSigPulseSignalFrequency,
    pktcSigPulseSignalDbLevel,
    pktcSigPulseSignalDuration,
    pktcSigPulseSignalPulseInterval,
    pktcSigPulseSignalRepeatCount,
    pktcSigDevToneDbLevel,

```

```

pktcSigDevToneFreqCounter,
pktcSigDevToneWholeToneRepeatCount,
pktcSigDevToneSteady,
pktcSigDevToneFirstFreqValue,
pktcSigDevToneSecondFreqValue,
pktcSigDevToneThirdFreqValue,
pktcSigDevToneFourthFreqValue,
pktcSigDevToneFreqMode,
pktcSigDevToneFreqAmpModePrtg,
pktcSigDevToneFreqOnDuration,
pktcSigDevToneFreqOffDuration,
pktcSigDevToneFreqRepeatCount
}

```

STATUS current

DESCRIPTION

" Group of objects that extend the behavior of existing objects to support operations in the widest possible set of international marketplaces. Note that many of these objects represent a superset of behaviors described in other objects within this MIB module."

```
 ::= { pktcSigGroups 3 }
```

pktcLLinePackageGroup OBJECT-GROUP

```

OBJECTS {
  pktcSigDevR0Cadence,
  pktcSigDevR1Cadence,
  pktcSigDevR2Cadence,
  pktcSigDevR3Cadence,
  pktcSigDevR4Cadence,
  pktcSigDevR5Cadence,
  pktcSigDevR6Cadence,
  pktcSigDevR7Cadence,
  pktcSigDevRgCadence,
  pktcSigDevRsCadence
}

```

STATUS current

DESCRIPTION

"Group of Objects to support the L line package."

```
 ::= { pktcSigGroups 4 }
```

pktcELinePackageGroup OBJECT-GROUP

```

OBJECTS {
  pktcSigDevR0Cadence,
  pktcSigDevR1Cadence,
  pktcSigDevR2Cadence,
  pktcSigDevR3Cadence,
  pktcSigDevR4Cadence,
  pktcSigDevR5Cadence,

```

```
pktcSigDevR6Cadence,  
pktcSigDevR7Cadence,  
pktcSigDevRgCadence,  
pktcSigDevRsCadence,  
pktcSigPulseSignalFrequency,  
pktcSigPulseSignalDbLevel,  
pktcSigPulseSignalDuration,  
pktcSigPulseSignalPulseInterval,  
pktcSigPulseSignalRepeatCount,  
pktcSigDevRingCadence  
}  
STATUS current  
DESCRIPTION  
    "Group of Objects to support the E line package."  
 ::= { pktcSigGroups 5 }
```

END

## 6. Examples

This section provides a couple of examples, specifically related to the MIB tables `pktcSigDevToneTable` and `pktcSigDevMultiFreqToneTable`.

Example A: Call Waiting Tone Defined per [ITU-T E.180]:

- 1) 400 Hz AM modulated by 16 Hz, on for 500ms at -4 dBm
- 2) 400 Hz AM modulated by 16 Hz, off for 400ms
- 3) 400 Hz not AM modulated, on for 50 ms at -4 dBm
- 4) 400 Hz not AM modulated, off for 450 ms
- 5) 400 Hz not AM modulated, on for 50 ms at -4 dBm
- 6) 400 Hz not AM modulated, off for 3450 ms
- 7) 400 Hz not AM modulated, on for 50 ms at -4 dBm
- 8) 400 Hz not AM modulated, off for 450 ms
- 9) 400 Hz not AM modulated, on for 50 ms at -4 dBm
- 10) 400 Hz not AM modulated, off for 3450 ms
- 11) not repeated, not continuous

Assume userDefined1(18) is assigned to this tone:

pktcSigDevMultiFreqToneTable:

ToneType	F-1	F-2	F-3	F-4	F-Mode	ModePrtg	DbL	OnDur	OffDur	Rep-Count
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
18	400	16	0	0	1	90	-40	500	400	0
18	400	0	0	0	2	0	-40	50	450	0
18	400	0	0	0	2	0	-40	50	3450	0
18	400	0	0	0	2	0	-40	50	450	0
18	400	0	0	0	2	0	-40	50	3450	0

pktcSigDevToneTable:

ToneType	ToneFreqGroup	ToneFreqCounter	ToneRep-Count	Steady
=====	=====	=====	=====	=====
18	1	5	0	false(2)

The single row of the pktcSigDevToneTable defines one multi-frequency group of five rows (ToneFreqCounter) defined in the pktcSigDevMultiFreqToneTable and instructs the MTA to play this group only once (non-repeatable as ToneRep-Count equals 0).

Example B - Congestion Tone - congestion(17):

Note: This example of an embedded cadence is based on an operator variation.

1) 400Hz on for 400ms -10 dBm

2) 400Hz off for 350ms

3) 400Hz on for 225ms -4 dBm

4) 400Hz off for 525ms

5) repeat (1) through (4) 5000 times or T0 time out (whichever is the shortest period)

pktcSigDevMultiFreqToneTable:

ToneType	F-1	F-2	F-3	F-4	F-Mode	ModePrtg	DbL	OnDur	OffDur	Rep-Count
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
17	400	0	0	0	2	0	-100	400	350	0
17	400	0	0	0	2	0	-40	225	525	0

pktcSigDevToneTable:

ToneType	ToneFreqGroup	ToneFreqCounter	ToneRep-Count	Steady
=====	=====	=====	=====	=====
17	1	2	5000	false(2)

Example C - Call Waiting Tone - callWaiting1(9):

- 1) 16 Hz is modulated to carry the 400 Hz signal, ModulationRate within 85%, on for 500msec, at -25 dBm or more but less than -14 dBm
- 2) 16 Hz is modulated to carry the 400 Hz signal, off for 0 ~ 4 secs
- 3) 400 Hz not modulated, on for 50 ms at -25 dBm or more but less than -14 dBm
- 4) 400 Hz not modulated, off for 450ms
- 5) 400 Hz not modulated, on for 50 ms at -25 dBm or more but less than -14 dBm
- 6) 400 Hz not modulated, off for 3450ms ([4000 - (50+450+50)])
- 7) Steps 3 thru 6 are repeated

pktcSigDevMultiFreqToneTable:

ToneType	F-1	F-2	F-3	F-4	F-Mode	ModePrtg	DbL	OnDur	OffDur	Rep-Count	
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	
9	1	400	16	0	0	1	85	-25	500	1000	0
9	2	400	0	0	0	2	0	-25	50	450	0
9	3	400	0	0	0	2	0	-25	50	3450	0

pktcSigDevToneTable:

ToneType	ToneFreqGroup	ToneFreqCounter	ToneRep-Count	Steady
9	1	1	0	false(2)
9	2	2	1	false(2)

The first row of the pktcSigDevToneTable table instructs the MTA to play one row (ToneFreqCounter) of the pktcSigDevMultiFreqToneTable table only once (non-repeatable as ToneRep-Count equals 0). The second row of the pktcSigDevToneTable table instructs the MTA to play the next two rows (ToneFreqCounter) of the pktcSigDevMultiFreqToneTable table and make this frequency group repeatable (ToneRep-Count is not 0).

## 7. Acknowledgments

The authors would like to thank the members of the IETF IPCDN working group and the CableLabs PacketCable Provisioning focus team for their contributions, comments, and suggestions.

Specifically, the following individuals are recognized:

Angela Lyda	Arris Interactive
Romascanu, Dan	Avaya
Chad Griffiths	Broadcom Corp.
Eugene Nechamkin	Broadcom Corp.
Jean-Francois Mule	CableLabs
Matt A. Osman	CableLabs
Klaus Hermanns	Cisco Systems, Inc.
Rich Woundy	Comcast Corp.
Bert Wijnen	Alcatel-Lucent
Randy Presuhn	Mindspring
Phillip Freyman	Motorola, Inc.
Rick Vetter	Motorola, Inc.
Sasha Medvinsky	Motorola, Inc.
Wim De Ketelaere	tComLabs
David De Reu	tComLabs
Kristof Sercu	tComLabs
Roy Spitzer	Telogy Networks, Inc.
Itay Sherman	Texas Instruments, Inc.
Mauricio Sanchez	Texas Instruments, Inc.
Shivakumar Thangapandi	Texas Instruments, Inc.
Mike Heard	Consultant

The current editor (Sumanth Channabasappa) would like to recognize Phillip Freyman and Eugene Nechamkin for their contributions towards the international objects, and Stephane Bortzmeyer for assistance with the ABNF.

The editor also extends appreciation to the IPCDN co-chairs (Jean-Francois Mule, Rich Woundy) and Dan Romascanu for the numerous reviews and valuable comments. Special appreciation is extended to Bert Wijnen, as the MIB doctor, for his ever-useful and constructive comments.



## 8. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The following Differentiated Services Code Point (DSCP) and mask objects are used to differentiate between various types of traffic in the service provider network:

```
pktcSigDefCallSigDscp
pktcSigDefMediaStreamDscp
```

These objects may contain information that may be sensitive from a business perspective. For example, they may represent a customer's service contract that a service provider chooses to apply to a customer's ingress or egress traffic. If these objects are SET maliciously, it may permit unmarked or inappropriately marked signaling and media traffic to enter the service provider network, resulting in unauthorized levels of service for customers.

The following objects determine ring cadence, repeatable characteristics, signal duration, and caller id subscriber line protocol for telephony operation:

```
pktcSigDevR0Cadence
pktcSigDevR1Cadence
pktcSigDevR2Cadence
pktcSigDevR3Cadence
pktcSigDevR4Cadence
pktcSigDevR5Cadence
pktcSigDevR6Cadence
pktcSigDevR7Cadence
pktcSigDevRgCadence
pktcSigDevRsCadence
pktcSigDevCidSigProtocol
pktcSigDevVmwsiSigProtocol
pktcSigPulseSignalDuration
pktcSigPulseSignalPauseDuration
```

If these objects are SET maliciously, it may result in unwanted operation, or a failure to obtain telephony service from client (MTA) devices.

The objects in the `pkcSigEndPntConfigTable` are used for endpoint signaling. The `pkcSigEndPntConfigCallAgentId` object contains the name of the call agent, which includes the call agent Fully Qualified Domain Name (FQDN). If this object is SET maliciously, the MTA will not be able to communicate with the call agent, resulting in a disruption of telephony service. The `pkcSigEndPntConfigCallAgentUdpPort` object identifies the UDP port for NCS traffic. If this object is SET maliciously, the call agent will not receive NCS traffic from the MTA, also resulting in a disruption of telephony service.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. The most sensitive is `pkcSigEndPntStatusCallIpAddress` within `pkcSigEndPntConfigTable`. This information itself may be valuable to would-be attackers. Other MIB Objects of similar sensitivity include `pkcSigEndPntStatusError`, which can provide useful information to MTA impersonators, and `pkcSigDevCodecMax`, which can provide useful information for planning Denial of Service (DoS) attacks on MTAs.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 9. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER value recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER Value
-----	-----
pktcIetfSigMib	{ mib-2 169 }

## 10. References

### 10.1. Normative References

- [PKT-SP-MIB-SIG-1.0] PacketCable(TM) 1.0 Signaling MIB Specification, Issued, PKT-SP-MIB-SIG-I09-050812, August 2005.  
<http://www.packetcable.com/specifications/>  
<http://www.cablelabs.com/specifications/archives>
- [PKT-SP-MIB-SIG-1.5] PacketCable(TM) 1.5 Signaling MIB Specification, Issued, PKT-SP-MIB-SIG1.5-I01-050128, January 2005.  
<http://www.packetcable.com/specifications/>  
<http://www.cablelabs.com/specifications/archives>
- [PKT-SP-SEC] PacketCable Security Specification, Issued, PKT-SP-SEC-I12-050812, August 2005.  
<http://www.packetcable.com/specifications/>  
<http://www.cablelabs.com/specifications/archives>
- [ITU-T-J169] IPCablecom Network Call Signaling (NCS) MIB requirements, J.169, ITU-T, March, 2001.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

- [RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [PKT-SP-CODEC] PacketCable Audio/Video Codecs Specification PKT-SP-CODEC-IO5-040113.
- [PKT-SP-MGCP] PacketCable Network-Based Call Signaling Protocol Specification PKT-SP-EC-MGCP-I10-040402.
- [PKT-SP-PROV] PacketCable MTA Device Provisioning Specification PKT-SP-PROV-I10-040730.

## 10.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3435] Andreasen, F. and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 3435, January 2003.
- [RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC4682] Nechamkin, E. and J-F. Mule, "Multimedia Terminal Adapter (MTA) Management Information Base for PacketCable- and IPCablecom-Compliant Devices", RFC 4682, December 2006.

## [ETSI-TS-101-909-4]

ETSI TS 101 909-4: "Access and Terminals (AT); Digital Broadband Cable Access to the Public Telecommunications Network; IP Multimedia Time Critical Services; Part 4: Network Call Signaling Protocol".

## [ETSI-TS-101-909-9]

ETSI TS 101 909-9: "Access and Terminals (AT); Digital Broadband Cable Access to the Public Telecommunications Network; IP Multimedia Time Critical Services; Part 9: IPCablecom Network Call Signalling (NCS) MIB Requirements".

## [ETSI-EN-300-001]

ETSI EN 300-001 V1.5.1 (1998-10): "European Standard (Telecommunications series) Attachments to Public Switched Telephone Network (PSTN); General technical requirements for equipment connected to an analogue subscriber interface in the PSTN; Chapter 3: Ringing signal characteristics (national deviations are in Table 3.1.1)".

## [ETSI-EN-300-324-1]

ETSI EN 300 324-1 V2.1.1 (2000-04): "V Interfaces at the digital Loop Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification".

## [ETSI-EN-300-659-1]

ETSI EN 300 659-1: "Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 1: On hook data transmission".

## [ITU-T-E.180] ITU-T E.180: "Various Tones Used in National Networks, Supplement 2 to Recommendation E.180".

## [ETSI-TR-101-183]

ETSI TR-101-183: "Public Switched Telephone Network (PSTN) Analogue Ringing Signals".

## Authors' Addresses

Gordon Beacham  
Motorola, Inc.  
6450 Sequence Drive, Bldg. 1  
San Diego, CA 92121, USA

Phone: +1 858-404-2334  
EMail: gordon.beacham@motorola.com

Satish Kumar Mudugere Eswaraiah  
Texas Instruments India (P) Ltd.,  
Golf view, Wind Tunnel Road  
Murugesh Palya  
Bangalore 560 017, INDIA

Phone: +91 80 5269451  
EMail: satish.kumar@ti.com

Sumanth Channabasappa  
Cable Television Laboratories, Inc.  
858 Coal Creek Circle,  
Louisville, CO 80027, USA

Phone: +1 303-661-3307  
EMail: Sumanth@cablelabs.com

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

