

INTERNET-DRAFT

draft-ietf-ipcdn-pktn-sigaling-09.txt

Expires: March 2006

Gordon Beacham
Motorola, Inc.
Satish Kumar
Texas Instruments
Sumanth Channabasappa
CableLabs(R)
September 2005

Network-Based Call Signaling (NCS) Signaling MIB for PacketCable and
IPCablecom Multimedia Terminal Adapters (MTAs)

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

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 provides a common data and format representation for PacketCable and IPCablecom compliant Multimedia Terminal Adapter devices.

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2. The set of objects are consistent with the SNMP framework and existing SNMP standards.

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 pktcSigDevConfigObjects.....5
 4.3 pktcSigNotification.....7
 4.4 pktcSigConformance.....7
 5. Definitions.....7
 6. Examples.....55
 7. Acknowledgments.....57
 8. Security Considerations.....57
 9. IANA Considerations.....59
 10. Normative References.....59
 11. Informative References.....61
 Authors' Addresses.....62
 Disclaimer of validity.....62
 Full Copyright Statement.....63

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 MIBmodule 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 [PKT-SP-PROV] and [RFCXYZ]. 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 [PKT-SP-MGCP] and [RFC3435]. For more information on codecs refer to [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", "NOT RECOMMENDED", "MAY", and "OPTIONAL", when used in the guidelines in this memo, 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 a 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 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 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 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 Signaling PacketCable, European Telecommunications Standards Institute(ETSI), and International Telecommunication Union Telecommunication Standardization Sector (ITU-T) IPCablecom compliant Multimedia Terminal Adapter (MTA) devices. The SIG MIB module is intended to supersede various SIG 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 SIG MIB requirements [ITU-T-J169],
- the ETSI SIG MIB [ETSI-TS-101-909-9]. The ETSI SIG MIB requirements also refer to various signal characteristics defined in [ETSI-EN-300-001], [ETSI-EN-300-659-1], and [ETSI-TR-101-183].

Several normative and informative references are used to help define NCS 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 is made by utilizing the ifTable (IF-MIB), where each index (ifIndex) 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 signaling MIB objects to allow for differentiation between various types of traffic in the service provider network. This MIB module also utilizes SNMP management MIB architecture from SNMP-FRAMEWORK-MIB [RFC3411].

4.1 Structure of the MIB

This MIB is structured in three groups:

- Signaling information that control device and endpoint configuration objects (pktcSigMibObjects)
- Signaling Notification object, that notifies the status (pktcSigNotification)
- Signaling Conformance has mandatory signaling objects (pktcSigConformance)

Each group of objects is explained in detail.

4.2 pktcSigDevConfigObjects

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).

pktcSigDevR0Cadence - this object specifies ring cadence 0.

pktcSigDevR1Cadence - this object specifies ring cadence 1.

pktcSigDevR2Cadence - this object specifies ring cadence 2.

pktcSigDevR3Cadence - this object specifies ring cadence 3.

pktcSigDevR4Cadence - this object specifies ring cadence 4.

pktcSigDevR5Cadence - this object specifies ring cadence 5.

pktcSigDevR6Cadence - this object specifies ring cadence 6.

pktcSigDevR7Cadence - this object specifies ring cadence 7.

pktcSigDevRgCadence - this object specifies ring cadence for ringing (rg).

pktcSigDevRsCadence - this object specifies ring cadence for ring splash (rs).

pktcSigDefCallSigDscp - this object specifies the default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for call signaling.

pkcSigDefMediaStreamDscp - this object specifies the default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for media stream packets.

pkcSigCapabilityTable - this table specifies list of supported signaling types, versions and vendor extensions for MTA.

pkcSigDefNcsReceiveUdpPort - this object contains the MTA User Datagram Protocol (UDP) receive port that is being used for NCS call signaling.

pkcSigDevCIDMode - this international object selects various modes of caller id in common use.

pkcSigDevCIDFskAfterRing - this international object sets the delay between the end of first ringing and the transmission of caller id information.

pkcSigDevCIDFskAfterDTAS - this international object sets the delay between the dual-tone alert signal and the transmission of caller id information.

pkcSigDevCIDFskAfterRPAS - this international object sets the delay between the ring pulse alert signal and the transmission of caller id information.

pkcSigDevCIDRingAfterFSK - this international object sets the delay between the transmission of caller id information and the first ringing pattern.

pkcSigDevCIDDTASAfterLR - this international object sets the delay between the end of a line reversal and the dual-tone alert signal.

pkcSigDevVmwMode - this object selects various modes of visual message waiting indicator service in common use.

pkcSigDevVmwFskAfterDTAS - this international object sets the delay between the dual-tone alert signal and the transmission of visual message waiting information.

pkcSigDevVmwFskAfterRPAS - this international object sets the delay between the ring pulse alert signal and the transmission of visual message waiting information.

pkcSigDevVmwDTASAfterLR - this international object sets the delay between the end of a line reversal and the dual-tone alert signal for visual message waiting information.

pktcSigDevRingCadenceTable - this international object provides a flexible structure within which to specify a variety of ring cadences.

pktcSigDevToneTable - this international 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.

pktcNcsEndPointConfigTable - this table describes the PacketCable NCS endPoint configuration. The number of entries in this table represents the number of provisioned endpoints.

pktcSigEndPointConfigTable - this table describes the PacketCable endPoint selected signaling type. The number of entries in this table represents the number of provisioned endpoints.

4.3 pktcSigNotification

pktcSigNotification - this object is used for signaling notification and reserved for future use.

4.4 pktcSigConformance

pktcSigCompliances - this table has one object that has compliance statements for devices that implement Signaling on the MTA.

pktcSigGroups - this table contains group of objects for the common portion of the PacketCable NCS and Signaling MIB.

pktcInternationalGroup - this table extends this MIB Module by establishing a set of 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 "200509090000Z" -- September 9, 2005
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: +1 858-404-2335
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

IETF IPCDN Working Group
General Discussion: 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
Co-Chair: Richard Woundy, Richard_Woundy@cable.comcast.com

DESCRIPTION

"This MIB module supplies the basic management object 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 Internet Society (2005). This version of this MIB module is part of RFC yyyy; see the RFC itself for full legal notices."

-- RFC Ed: replace yyyy with actual RFC number and remove this note

REVISION "200509090000Z"

DESCRIPTION

"Initial version, published as RFC yyyy."

-- RFC Ed: replace yyyy with actual RFC number and remove this note

::= { mib-2 XXX }

-- RFC Ed: replace XXX with IANA-assigned number and remove this

-- note

-- 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

PkrtcCodecType ::= 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 in PacketCable
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 [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. 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. 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 - set when signaling other than ncs is used

reserved - for future use

ncs - Network call signaling is a derivation of MGCP (Media Gateway Control Protocol) defined for IPCablecom/PacketCable MTAs."

SYNTAX INTEGER {
 other(1),
 reserved(2),

```

        ncs(3)
    }

pktcSigMibObjects OBJECT IDENTIFIER ::= { pkcIetfSigMib 1 }
pktcSigDevConfigObjects OBJECT IDENTIFIER ::=
    { pktcSigMibObjects 1 }
pktcNcsEndPntConfigObjects 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 PkctcSigDevCodecEntry
    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 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:

CodecComboIndex	pkctcSigDevCodecType	pkctcSigDevCodecMax
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."

```
::= { pktcSigDevConfigObjects 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 which 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."
 ::= { pktcSigDevConfigObjects 2 }

pktcSigDevSilenceSuppression OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object specifies if the device is capable of
          silence suppression (Voice Activity Detection)."
 ::= { pktcSigDevConfigObjects 3 }

pktcSigDevCallerIdSigProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
                    fsk (1),
                    dtmf (2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object identifies the subscriber line protocol used
         for signaling on-hook caller id information. Different
         countries define different caller id signaling protocols to
         support caller identification. Frequency shift keying (FSK)
         is most commonly used. Dual tone multi-frequency (DTMF)

```

```
        is an alternative."
REFERENCE
    "ETSI-EN-300-659-1 Specification"
DEFVAL { fsk }
 ::= { pktcSigDevConfigObjects 4 }

pktcSigDevR0Cadence      OBJECT-TYPE
SYNTAX      PktcRingCadence
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object specifies ring cadence 0 (a user defined
      field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 5 }

pktcSigDevR1Cadence      OBJECT-TYPE
SYNTAX      PktcRingCadence
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object specifies ring cadence 1 (a user defined
      field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 6 }

pktcSigDevR2Cadence      OBJECT-TYPE
SYNTAX      PktcRingCadence
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object specifies ring cadence 2 (a user defined
      field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 7 }

pktcSigDevR3Cadence      OBJECT-TYPE
SYNTAX      PktcRingCadence
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object specifies ring cadence 3 (a user defined
      field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 8 }

pktcSigDevR4Cadence      OBJECT-TYPE
SYNTAX      PktcRingCadence
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object specifies ring cadence 4 (a user defined
      field). This object is required for the L line package."
```

```
 ::= { pktcSigDevConfigObjects 9 }

pktcSigDevR5Cadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence 5 (a user defined
          field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 10 }

pktcSigDevR6Cadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence 6 (a user defined
          field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 11 }

pktcSigDevR7Cadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence 7 (a user defined
          field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 12 }

pktcSigDevRgCadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence rg (a user defined
          field). This object is required for the L line package."
 ::= { pktcSigDevConfigObjects 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. This object is required for the L line
          package."
 ::= { pktcSigDevConfigObjects 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."
    DEFVAL { 0 }
    ::= { pktcSigDevConfigObjects 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."
    DEFVAL { 0 }
    ::= { pktcSigDevConfigObjects 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."
    ::= { pktcSigDevConfigObjects 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 { pktcSignalingIndex }
 ::= { pktcSigCapabilityTable 1 }
```

```
PktcSigCapabilityEntry ::= SEQUENCE {
    pktcSignalingIndex      Unsigned32,
    pktcSignalingType       PktcSigType,
    pktcSignalingVersion    SnmpAdminString,
    pktcSignalingVendorExtension SnmpAdminString
}
```

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

```
pktcSignalingType      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 }
```

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

```
pktcSignalingVendorExtension OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " The vendor extension allows vendors to provide a list of
          additional capabilities, vendors can decide how to encode
```

these extensions, although space separated text is suggested."
 ::= { pktcSigCapabilityEntry 4 }

pktcSigDefNcsReceiveUdpPort OBJECT-TYPE

SYNTAX 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."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 2427 }

::= { pktcSigDevConfigObjects 18 }

pktcSigPowerRingFrequency OBJECT-TYPE

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

}

UNITS "Hertz"

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 contains a list of frequency ranges that are defined for each country."

DEFVAL { f20Hz }

::= { pktcSigDevConfigObjects 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. This object is required for the E line package. Signals defined in this table are triggered using the E line package. Objects in this table do not persist across MTA reboots."

REFERENCE

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

::= { pktcSigDevConfigObjects 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."

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)
    }
    UNITS                    "Hertz"
    MAX-ACCESS                read-write
    STATUS                    current
    DESCRIPTION
        " This object is only applicable to the initialRing,
        enableMeterPulse, and meterPulseBurst signal type. 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 'inconsistent value' error."

```

REFERENCE

```

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

```

```

pktcSigPulseSignalDbLevel  OBJECT-TYPE

```

```

SYNTAX      TenthdBm (-350..0)

```

```

UNITS       "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 'inconsistent value' 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 'inconsistent value' 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 'inconsistent value' error."

REFERENCE

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

pktcSigPulseSignalRepeatCount OBJECT-TYPE

SYNTAX Unsigned32
 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 as such must have a value of zero. 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	0	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 (or is not used) for the specific signal type will result in an 'inconsistent value' error."

```
::={ pktcSigPulseSignalEntry 6 }
```

```
pktcSigDevCIDMode OBJECT-TYPE
```

```
SYNTAX      INTEGER {
                duringRingingETS(1),
                dtAsETS(2),
                rpAsETS(3),
                lrAsETS(4)
            }
```

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

```
STATUS current
```

```
DESCRIPTION
```

```
" For on-hook Caller ID, pktcSigDevCIDMode selects the
method of Caller ID. For the duringRingingETS method, the
Frequency Shift Keying (FSK) containing the Caller ID
information is sent between the first and second ring
pattern. For the dtAsETS, rpAsETS, and lrAsETS methods, the
FSK containing the Caller ID information is sent before
the first ring pattern. For the dtAsETS method, the FSK is
sent after the Dual Tone Alert Signal. For the rpAsETS
method, the FSK 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 is sent."
```

```
DEFVAL { rpAsETS }
```

```
::= {pktcSigDevConfigObjects 21 }
```

```
pktcSigDevCIDFskAfterRing OBJECT-TYPE
```

```
SYNTAX      Unsigned32 (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 containing the Caller ID information. It is only used when pktcSigDevCIDMode is duringringingETS. The following table defines the default values for this object depending on signal type:

pktcSigDevCIDMode	Default
duringringingETS	550 ms
dtAsETS	not used
rpAsETS	not used
lrAsETS	not used

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

REFERENCE

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

DEFVAL { 550 }

::= {pktcSigDevConfigObjects 22 }

pktcSigDevCIDFskAfterDTAS OBJECT-TYPE

SYNTAX Unsigned32 (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 containing the Caller ID information. This object is only used when pktcSigDevCIDMode is dtAsETS or lrAsETS. The following table defines the default values for this object depending on signal type:

pktcSigDevCIDMode	Default
duringringingETS	not used
dtAsETS	50 ms
rpAsETS	not used
lrAsETS	50 ms

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

REFERENCE

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

DEFVAL { 50 }

::= {pktcSigDevConfigObjects 23 }

pktcSigDevCIDFskAfterRPAS OBJECT-TYPE

SYNTAX Unsigned32 (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 containing the Caller ID
      information. This object is only used when
      pktcSigDevCIDMode is rpAsETS. The following table defines
      the default values for this object depending on signal
      type:
      pktcSigDevCIDMode      Default
      duringringingETS      not used
      dtAsETS                not used
      rpAsETS                650 ms
      lrAsETS                not used
      An attempt to set this object while the value of
      pktcSigDevCIDMode is not rpAsETS will result in an
      'inconsistent value' error."
REFERENCE
    "ETSI-EN-300-659-1 Specification"
DEFVAL { 650 }
 ::= { pktcSigDevConfigObjects 24 }

```

```

pktcSigDevCIDRingAfterFSK      OBJECT-TYPE
SYNTAX          Unsigned32 (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 containing the Caller ID
      information and the start of the first ring pattern. It is
      only used when pktcSigDevCIDMode is dtAsETS, rpAsETS or
      lrAsETS. The following table defines the default values
      for this object depending on signal type:
      pktcSigDevCIDMode      Default
      duringringingETS      not used
      dtAsETS                250 ms
      rpAsETS                250 ms
      lrAsETS                250 ms
      An attempt to set this object while the value of
      pktcSigDevCIDMode is not dtAsETS, rpAsETS, or lrAsETS will
      result in an 'inconsistent value' error."
REFERENCE
    "ETSI-EN-300-659-1 Specification"
DEFVAL { 250 }
 ::= { pktcSigDevConfigObjects 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 lrAsETS. The following table defines the default values
  for this object depending on signal type:
  pktcSigDevCIDMode      Default
  duringringingETS      not used
  dtAsETS                not used
  rpAsETS                not used
  lrAsETS                250 ms
  An attempt to set this object while the value of
  pktcSigDevCIDMode is not lrAsETS will result in an
  'inconsistent value' error."
REFERENCE
  "ETSI-EN-300-659-1 Specification"
DEFVAL { 250 }

 ::= { pktcSigDevConfigObjects 26 }

```

```

pktcSigDevVmwMode      OBJECT-TYPE
SYNTAX                INTEGER {
                        dtAsETS(1),
                        rpAsETS(2),
                        lrAsETS(3),
                        osi(4)
                      }
MAX-ACCESS            read-write
STATUS                current
DESCRIPTION
  " For visual message waiting indicator (VMWI),
  pktcSigDevVmwMode selects the alerting signal method. For
  the dtAsETS, rpAsETS, lrAsETS, and OSI methods,
  the FSK containing the VMWI information is sent after an
  alerting signal. For the dtAsETS method, the FSK is sent
  after the Dual Tone Alert Signal. For the rpAsETS method,
  the FSK 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 is sent. For the OSI
  method, the FSK is sent after the Open Switching
  Interval."
DEFVAL { dtAsETS }
 ::= { pktcSigDevConfigObjects 27 }

```

```

pktcSigDevVmwFskAfterDTAS  OBJECT-TYPE
SYNTAX                    Unsigned32 (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 containing the VMWI information.
    This object is only used when pktcSigDevVmwiMode is
    dtAsETS or lrAsETS. The following table defines the
    default values for this object depending on signal type:
    pktcSigDevVmwiMode    Default
    dtAsETS               50 ms
    rpAsETS               not used
    lrAsETS               50 ms
    An attempt to set this object while the value of
    pktcSigDevVmwiMode is not dtAsETS or lrAsETS will result
    in an 'inconsistent value' error."
REFERENCE    "ETSI-EN-300-659-1 Specification"
DEFVAL { 50 }
 ::= { pktcSigDevConfigObjects 28 }

```

```

pktcSigDevVmwiFskAfterRPAS    OBJECT-TYPE
SYNTAX      Unsigned32 (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 containing the VMWI information.
    This object is only used when pktcSigDevVmwiMode is
    rpAsETS. The following table defines the default values
    for this object depending on signal type:
    pktcSigDevVmwiMode    Default
    dtAsETS               not used
    rpAsETS               650 ms
    lrAsETS               not used
    An attempt to set this object while the value of
    pktcSigDevVmwiMode is not rpAsETS will result in an
    'inconsistent value' error."
REFERENCE    "ETSI-EN-300-659-1 Specification"
DEFVAL { 650 }
 ::= { pktcSigDevConfigObjects 29 }

```

```

pktcSigDevVmwiDTASAfterLR    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) for VMWI information. This object is only used
 when pktcSigDevVmwiMode is lrAsETS. The following table
 defines the default values for this object depending on
 signal type:

pktcSigDevVmwiMode	Default
dtAsETS	not used
rpAsETS	not used
lrAsETS	250 ms

 An attempt to set this object while the value of
 pktcSigDevVmwiMode is not lrAsETS will result in an
 'inconsistent value' error."
 REFERENCE
 "ETSI-EN-300-659-1 Specification"
 DEFVAL { 250 }
 ::= { pktcSigDevConfigObjects 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. Objects in this table do not
 persist across MTA reboots."
 REFERENCE
 "ETSI-TS-101-909-4 Specification"
 ::= { pktcSigDevConfigObjects 31 }

pktcSigDevRingCadenceEntry OBJECT-TYPE
 SYNTAX PktcSigDevRingCadenceEntry

```

MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    " Unique value ranging from 0 to 127 that will correspond to
      the different ring cadences that are being supported by
      the device."
INDEX { pktcSigDevRingCadenceIndex }
 ::= { pktcSigDevRingCadenceTable 1 }

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

pktcSigDevRingCadenceIndex  OBJECT-TYPE
SYNTAX    Unsigned32 (0..127)
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    " 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. This object is required for the
      E line package."
 ::= { pktcSigDevRingCadenceEntry 2 }

pktcSigDevToneTable  OBJECT-TYPE
SYNTAX    SEQUENCE OF PktcSigDevToneEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    " The Tone Table defines the various tone operations. Any
      definition of the tones callWaiting1-4 in this table
      should just contain the audible tone itself and NOT
      contain the delay between tones or the tone repeat count.
      The delay between tones or the repeat count is controlled
      by the objects pktcNcsEndPntConfigCallWaitingDelay, and

```

pktcNcsEndPntConfigCallWaitingMaxRep. If the pktcSigDevToneType is set to either of the values callWaiting1, callWaiting2, callWaiting3 or callWaiting4, then the value of the pktcSigDevToneWholeToneRepeatCount object has no effect on the tone. 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 by the MTA Configuration File. The frequency composition of each tone is defined by the pktcSigDevMultiFreqToneTable. For each ToneType defined in pktcSigDevToneTable, the MTA MUST populate at least one entry 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."

::= { pktcSigDevConfigObjects 32 }

pktcSigDevToneEntry OBJECT-TYPE

SYNTAX PktcSigDevToneEntry

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

INDEX { pktcSigDevToneType }

::= { pktcSigDevToneTable 1 }

PktcSigDevToneEntry ::= SEQUENCE {

pktcSigDevToneType	INTEGER,
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

```

"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 }
```

```
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 that The TO duration setting should always be longer than the desired repeat count time duration."

```
::={ pktcSigDevToneEntry 2 }
```

```
pktcSigDevToneSteady      OBJECT-TYPE
```

```
    SYNTAX                TruthValue
```

```
    MAX-ACCESS            read-only
```

```
    STATUS                current
```

```
    DESCRIPTION
```

```
        "This is the steady tone. Device must play out the on-off cadence sequence for pktcSigDevToneWholeRepeatCount times
```

```
and
```

```
        then apply the last tone forever. Setting this object may result in a tone 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 that The TO duration setting should always be longer than the desired repeat count time duration plus the desired maximum steady tone period."
```

```
::={ pktcSigDevToneEntry 3 }
```

```
pktcSigDevMultiFreqToneTable  OBJECT-TYPE
```

```
    SYNTAX                SEQUENCE OF PktcSigDevMultiFreqToneEntry
```

```
    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 using the MTA configuration file."

REFERENCE

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

::= { pktcSigDevConfigObjects 35 }

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	Integer32,
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 Integer32(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..-30)

UNITS "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 MUST reflect the desired level at the Telco (POTS) a-b (T/R) terminals including the affect 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 (loss) - but does not imply the entire range is to be achievable given the range of gains (loss) in the MTA."

DEFVAL { -40 }

::={ pktcSigDevMultiFreqToneEntry 8}

pktcSigDevToneFreqOnDuration OBJECT-TYPE

SYNTAX Unsigned32(0..5000)

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)

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 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 the 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}
```

```
--
```

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

```
--
```

```
pktcNcsEndPntConfigTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF PktcNcsEndPntConfigEntry
```

```
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 signaling 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."
```

```
::= { pktcNcsEndPntConfigObjects 1 }
```

```
pktcNcsEndPntConfigEntry OBJECT-TYPE
```

```
SYNTAX PktcNcsEndPntConfigEntry
```

```

MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "Each entry in the pktcNcsEndPntConfigTable represents
    required signaling parameters for the specific endpoint
    provisioned with voice services."
INDEX { ifIndex }
 ::= { pktcNcsEndPntConfigTable 1 }

```

```

PktcNcsEndPntConfigEntry ::= SEQUENCE {
    pktcNcsEndPntConfigCallAgentId          SnmpAdminString,
    pktcNcsEndPntConfigCallAgentUdpPort    InetPortNumber,
    pktcNcsEndPntConfigPartialDialTO       Unsigned32,
    pktcNcsEndPntConfigCriticalDialTO      Unsigned32,
    pktcNcsEndPntConfigBusyToneTO          Unsigned32,
    pktcNcsEndPntConfigDialToneTO          Unsigned32,
    pktcNcsEndPntConfigMessageWaitingTO    Unsigned32,
    pktcNcsEndPntConfigOffHookWarnToneTO   Unsigned32,
    pktcNcsEndPntConfigRingingTO           Unsigned32,
    pktcNcsEndPntConfigRingBackTO          Unsigned32,
    pktcNcsEndPntConfigReorderToneTO       Unsigned32,
    pktcNcsEndPntConfigStutterDialToneTO   Unsigned32,
    pktcNcsEndPntConfigTSMAX                Unsigned32,
    pktcNcsEndPntConfigMax1                 Unsigned32,
    pktcNcsEndPntConfigMax2                 Unsigned32,
    pktcNcsEndPntConfigMax1QEnable          TruthValue,
    pktcNcsEndPntConfigMax2QEnable          TruthValue,
    pktcNcsEndPntConfigMWD                  Unsigned32,
    pktcNcsEndPntConfigTdinit                Unsigned32,
    pktcNcsEndPntConfigTdmin                 Unsigned32,
    pktcNcsEndPntConfigTdmax                 Unsigned32,
    pktcNcsEndPntConfigRtoMax                Unsigned32,
    pktcNcsEndPntConfigRtoInit              Unsigned32,
    pktcNcsEndPntConfigLongDurationKeepAlive Unsigned32,
    pktcNcsEndPntConfigThist                 Unsigned32,
    pktcNcsEndPntConfigStatus                RowStatus,
    pktcNcsEndPntConfigCallWaitingMaxRep     Unsigned32,
    pktcNcsEndPntConfigCallWaitingDelay      Unsigned32,
    pktcNcsEndPntStatusCallIpAddressType     InetAddressType,
    pktcNcsEndPntStatusCallIpAddress         InetAddress,
    pktcNcsEndPntStatusError                 INTEGER,
    pktcNcsEndPntConfigMinHookFlash          Unsigned32,
    pktcNcsEndPntConfigMaxHookFlash          Unsigned32,
    pktcNcsEndPntConfigPulseDialInterdigitTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMinMakeTime  Unsigned32,
    pktcNcsEndPntConfigPulseDialMaxMakeTime  Unsigned32,
    pktcNcsEndPntConfigPulseDialMinBreakTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMaxBreakTime Unsigned32
}

```

```

pktcNcsEndPntConfigCallAgentId      OBJECT-TYPE
    SYNTAX      SnmpAdminString(SIZE (3..255))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains a string indicating the call agent
          name (e.g.: ca@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."
    ::= { pktcNcsEndPntConfigEntry 1 }

pktcNcsEndPntConfigCallAgentUdpPort  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 signaling 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 }
    ::= { pktcNcsEndPntConfigEntry 2 }

pktcNcsEndPntConfigPartialDialTO     OBJECT-TYPE
    SYNTAX      Unsigned32

```



```

UNITS          "seconds"
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "This object contains the value of the partial dial
              time out."
REFERENCE     "PacketCable NCS Specification"
DEFVAL { 16 }
 ::= { pktcNcsEndPntConfigEntry 3 }

pktcNcsEndPntConfigCriticalDialTO      OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION "This object contains the value of the critical
              dial time out."
REFERENCE   "PacketCable NCS Specification"
DEFVAL { 4 }
 ::= { pktcNcsEndPntConfigEntry 4 }

pktcNcsEndPntConfigBusyToneTO          OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION " This object contains the default timeout value for busy
              tone. The MTA MUST NOT update this object with the
              value provided in the NCS message (if present). If
              the value of the object is modified by the SNMP Management
              Station, the MTA MUST use the new value as a default only
              for a new signal requested by the NCS message."
REFERENCE   "PacketCable NCS Specification"
DEFVAL      { 30 }
 ::= { pktcNcsEndPntConfigEntry 5 }

pktcNcsEndPntConfigDialToneTO          OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION " This object contains the default timeout value for dial
              tone. The MTA MUST NOT update this object with the

```

value provided in the NCS message (if present). If the value of the object is modified by the SNMP Management Station, the MTA MUST use the new value as a default only for a new signal requested by the NCS message."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcNcsEndPntConfigEntry 6 }

pktcNcsEndPntConfigMessageWaitingTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcNcsEndPntConfigEntry 7 }

pktcNcsEndPntConfigOffHookWarnToneTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

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

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 0 }

::= { pktcNcsEndPntConfigEntry 8 }

pktcNcsEndPntConfigRingingTO OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-create

```

STATUS          current
DESCRIPTION
    " This object contains the default timeout value for
      ringing. The MTA MUST NOT update this object with the
      value provided in the NCS message (if present). If
      the value of the object is modified by the SNMP Management
      Station, the MTA MUST use the new value as a default only
      for a new signal requested by the NCS message."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL          { 180 }
 ::= { pktcNcsEndPntConfigEntry 9 }

```

```

pktcNcsEndPntConfigRingBackTO      OBJECT-TYPE
SYNTAX          Unsigned32
UNITS           "seconds"
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    " This object contains the default timeout value for ring
      back. The MTA MUST NOT update this object with the
      value provided in the NCS message (if present). If
      the value of the object is modified by the SNMP Management
      Station, the MTA MUST use the new value as a default only
      for a new signal requested by the NCS message."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL          { 180 }
 ::= { pktcNcsEndPntConfigEntry 10 }

```

```

pktcNcsEndPntConfigReorderToneTO   OBJECT-TYPE
SYNTAX          Unsigned32
UNITS           "seconds"
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    " This object contains the default timeout value for reorder
      tone. The MTA MUST NOT update this object with the
      value provided in the NCS message (if present). If
      the value of the object is modified by the SNMP Management
      Station, the MTA MUST use the new value as a default only
      for a new signal requested by the NCS message."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL          { 30 }
 ::= { pktcNcsEndPntConfigEntry 11 }

```

```

pktcNcsEndPntConfigStutterDialToneTO OBJECT-TYPE
SYNTAX          Unsigned32

```

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

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 16 }

::= { pktcNcsEndPntConfigEntry 12 }

pktcNcsEndPntConfigTSMMax 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 Tsmax time has elapsed, then the retransmissions MUST cease.

Refer to the MIB Object pktcNcsEndPntConfigThist for Information on when the endpoint becomes disconnected."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 20 }

::= { pktcNcsEndPntConfigEntry 13 }

pktcNcsEndPntConfigMax1 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object contains the suspicious error threshold for signaling messages. The pktcNcsEndPntConfigMax1 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 }

```

 ::= { pktcNcsEndPntConfigEntry 14 }

pktcNcsEndPntConfigMax2      OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object contains the disconnect error threshold for
        signaling messages. The pktcNcsEndPntConfigMax2 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 }
    ::= { pktcNcsEndPntConfigEntry 15 }

pktcNcsEndPntConfigMax1QEnable  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object enables/disables the Max1 domain name server
        (DNS) query operation when the pktcNcsEndPntConfigMax1
        threshold has been reached."
    DEFVAL { true }
    ::= { pktcNcsEndPntConfigEntry 16 }

pktcNcsEndPntConfigMax2QEnable  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 pktcNcsEndPntConfigMax2
        threshold has been reached."
    DEFVAL { true }
    ::= { pktcNcsEndPntConfigEntry 17 }

pktcNcsEndPntConfigMWD      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 }
 ::= { pktcNcsEndPntConfigEntry 18 }

pktcNcsEndPntConfigTdinit 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 }
 ::= { pktcNcsEndPntConfigEntry 19 }

pktcNcsEndPntConfigTdmin 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 }
 ::= { pktcNcsEndPntConfigEntry 20 }

```

```

pktcNcsEndPntConfigTdmax      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 }
    ::= { pktcNcsEndPntConfigEntry 21 }

```

```

pktcNcsEndPntConfigRtoMax    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 }
    ::= { pktcNcsEndPntConfigEntry 22 }

```

```

pktcNcsEndPntConfigRtoInit   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 }
    ::= { pktcNcsEndPntConfigEntry 23 }

```

```

pktcNcsEndPntConfigLongDurationKeepAlive  OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "minutes"
    MAX-ACCESS  read-create

```

```

STATUS          current
DESCRIPTION
    " Specifies a timeout value in minutes for sending long
      duration call notification message."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 60 }
 ::= { pktcNcsEndPntConfigEntry 24 }

pktcNcsEndPntConfigThist OBJECT-TYPE
SYNTAX          Unsigned32
UNITS           "seconds"
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION
    " Timeout period in seconds before no response is declared."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 30 }
 ::= { pktcNcsEndPntConfigEntry 25 }

pktcNcsEndPntConfigStatus OBJECT-TYPE
SYNTAX          RowStatus
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION
    " This object contains the Row Status associated with the
      pktcNcsEndPntConfigTable. 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 active(1). "
 ::= { pktcNcsEndPntConfigEntry 26 }

pktcNcsEndPntConfigCallWaitingMaxRep 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 Management Station, the MTA MUST use
      the new value as a default only for a new signal
      requested by the NCS message."
DEFVAL          { 1 }
 ::= { pktcNcsEndPntConfigEntry 27 }

```



```

pktcNcsEndPntConfigCallWaitingDelay      OBJECT-TYPE
    SYNTAX      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 }
    ::= { pktcNcsEndPntConfigEntry 28 }

pktcNcsEndPntStatusCallIpAddressType    OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object contains the type of Internet address of the
          CMS currently being used for this endpoint."
    ::= { pktcNcsEndPntConfigEntry 29 }

pktcNcsEndPntStatusCallIpAddress      OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object contains the Internet address of the CMS
          currently being used for this endpoint. This Internet
          address is used to create the appropriate security
          association. The type of this IP address is determined by
          the value of the pktcNcsEndPntStatusCallIpAddressType
          object."
    ::= { pktcNcsEndPntConfigEntry 30 }

pktcNcsEndPntStatusError      OBJECT-TYPE
    SYNTAX INTEGER {
        operational (1),
        noSecurityAssociation (2),
        disconnected (3)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object contains the error status for this interface.
          The operational 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."

```
::= { pktcNcsEndPntConfigEntry 31 }
```

```
pktcNcsEndPntConfigMinHookFlash    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
      pktcNcsEndPntConfigPulseDialMaxBreakTime. The value of
      pktcNcsEndPntConfigMinHookFlash MUST be less than
      pktcNcsEndPntConfigMaxHookFlash. This object MUST only be
      set 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 greater than the value contained by the MIB Object
      pktcNcsEndPntConfigPulseDialMaxMakeTime."
```

```
DEFVAL { 300 }
```

```
::= { pktcNcsEndPntConfigEntry 32 }
```

```
pktcNcsEndPntConfigMaxHookFlash    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
      pktcNcsEndPntConfigMaxHookFlash MUST be greater than
      pktcNcsEndPntConfigMinHookFlash. This object MUST only be
```

```

        set via the configuration file during the provisioning
        process."
    DEFVAL { 800 }
    ::= { pktcNcsEndPntConfigEntry 33 }

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

pktcNcsEndPntConfigPulseDialMinMakeTime      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 pktcNcsEndPntConfigPulseDialMinMakeTime MUST
        be less than pktcNcsEndPntConfigPulseDialMaxMakeTime. This
        object MUST only be set via the configuration file during
        the provisioning process."
    DEFVAL { 25 }
    ::= { pktcNcsEndPntConfigEntry 35 }

pktcNcsEndPntConfigPulseDialMaxMakeTime      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 pktcNcsEndPntConfigPulseDialMaxMakeTime MUST
        be greater than pktcNcsEndPntConfigPulseDialMinMakeTime.
        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
        pktcNcsEndPntConfigMinHookFlash."
    DEFVAL { 55 }
    ::= { pktcNcsEndPntConfigEntry 36 }

```

```

pktcNcsEndPntConfigPulseDialMinBreakTime    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 pktcNcsEndPntConfigPulseDialMinBreakTime MUST
          be less than pktcNcsEndPntConfigPulseDialMaxBreakTime.
          This object must only be provided via the configuration
          file during the provisioning process."
    DEFVAL { 45 }
    ::= { pktcNcsEndPntConfigEntry 37 }

pktcNcsEndPntConfigPulseDialMaxBreakTime    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 pktcNcsEndPntConfigPulseDialMaxBreakTime MUST
          be greater than pktcNcsEndPntConfigPulseDialMinBreakTime.
          This object MUST only be provided via the configuration
          file during the provisioning process."
    DEFVAL { 75 }
    ::= { pktcNcsEndPntConfigEntry 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 devices that implement
          Signaling on the MTA."

MODULE -- pktcIetfSigMib

```

```
--
-- unconditionally mandatory groups
--

MANDATORY-GROUPS {
    pktcSigGroup
}

GROUP pktcNcsGroup
DESCRIPTION
    " This group is mandatory for any MTA implementing NCS
      signaling"

GROUP pktcInternationalGroup
DESCRIPTION
    " This group is mandatory for any MTA implementing
      international telephony features. In such cases, it is
      left to manufacturers to determine whether to support both
      PacketCable and IPCablecom objects in the same MTA."
 ::= { pktcSigCompliances 1 }

--
-- units of conformance
--

pktcSigGroup OBJECT-GROUP
OBJECTS {
    pktcSigDevCodecMax,
    pktcSigDevEchoCancellation,
    pktcSigDevSilenceSuppression,
    pktcSigDevR0Cadence,
    pktcSigDevR1Cadence,
    pktcSigDevR2Cadence,
    pktcSigDevR3Cadence,
    pktcSigDevR4Cadence,
    pktcSigDevR5Cadence,
    pktcSigDevR6Cadence,
    pktcSigDevR7Cadence,
    pktcSigDevRgCadence,
    pktcSigDevRsCadence,
    pktcSigDefCallSigDscp,
    pktcSigDefMediaStreamDscp,
    pktcSigDevVmwiMode,
    pktcSignalingType,
    pktcSignalingVersion,
    pktcSignalingVendorExtension,
    pktcSigDefNcsReceiveUdpPort
}
```

```

STATUS current
DESCRIPTION
    "Group of objects for the common portion of the
    PacketCable Signaling MIB."
 ::= { pktcSigGroups 1 }

```

```

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

pktcInternationalGroup OBJECT-GROUP
OBJECTS {
    pktcNcsEndPntConfigMinHookFlash,

```

```

pktcNcsEndPntConfigMaxHookFlash,
pktcNcsEndPntConfigPulseDialInterdigitTime,
pktcNcsEndPntConfigPulseDialMinMakeTime,
pktcNcsEndPntConfigPulseDialMaxMakeTime,
pktcNcsEndPntConfigPulseDialMinBreakTime,
pktcNcsEndPntConfigPulseDialMaxBreakTime,
pktcSigDevRingCadence,
pktcSigDevCallerIdSigProtocol,
pktcSigDevCIDMode,
pktcSigDevCIDFskAfterRing,
pktcSigDevCIDFskAfterDTAS,
pktcSigDevCIDFskAfterRPAS,
pktcSigDevCIDRingAfterFSK,
pktcSigDevCIDDTASAfterLR,
pktcSigDevVmwiFskAfterDTAS,
pktcSigDevVmwiFskAfterRPAS,
pktcSigDevVmwiDTASAfterLR,
pktcSigPowerRingFrequency,
pktcSigPulseSignalFrequency,
pktcSigPulseSignalDbLevel,
pktcSigPulseSignalDuration,
pktcSigPulseSignalPulseInterval,
pktcSigPulseSignalRepeatCount,
pktcSigDevToneDbLevel,
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 }

```

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(17) is assigned to this tone:

pktcSigDevMultiFreqToneTable:

ToneType	F-1	F-2	F-3	F-4	F-Mode	ModePrtg	DbL	OnDur	OffDur	Rep-Count
17	400	16	0	0	1	90	-40	500	400	0
17	400	0	0	0	2	0	-40	50	450	0
17	400	0	0	0	2	0	-40	50	3450	0
17	400	0	0	0	2	0	-40	50	450	0
17	400	0	0	0	2	0	-40	50	3450	0

pktcSigDevToneTable:

ToneType	ToneRep-Count	Steady
17	0	false(2)

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 timeout (which ever is 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 | ToneRep-Count | Steady
=====
17        5000           false(0)
    
```

7. Acknowledgments

This document is a production of the PacketCable Working Group.

The editors wish to express gratitude to:

- | | |
|------------------------|-------------------------|
| Angela Lyda | Arris Interactive |
| Chad Griffiths | Broadcom Corp. |
| Eugene Nechamkin | Broadcom Corp. |
| Jean-Francois Mule | CableLabs(R) |
| Matt A. Osman | CableLabs(R) |
| Klaus Hermanns | Cisco Systems, Inc. |
| Rich Woundy | Comcast Corp. |
| Bert Wijnen | Lucent Technologies |
| 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 |

8. Security Considerations

There are a number of management objects defined in this MIB that have 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
pktcSigDevCallerIdSigProtocol
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 `pktcNcsEndPntConfigTable` are used for endpoint signaling. The `pktcNcsEndPntConfigCallAgentId` 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 `pktcNcsEndPntConfigCallAgentUdpPort` object identifies the UDP port for NCS signaling traffic. If this object is SET maliciously, the

call agent will not receive NCS signaling 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 `pktcNcsEndPntStatusCallIpAddress` within `pktcNcsEndPntConfigTable`. This information itself may be valuable to would-be attackers.

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 values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER Value
-----	-----
<code>pktcIetfSigMib</code>	{ mib-2 XXX }

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for XXX under the mib-2 subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace XXX (here and in the MIB module) with the assigned value and to remove this note.

10. 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/>
- [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., Schoenwaelder, J., Case, J.,
Rose, M., and S. Waldbusser, "Structure of Management
Information Version 2 (SMIv2)", STD 58, RFC 2578, April
1999.
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J.,
Rose, M., and S. Waldbusser, "Textual Conventions for
SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J.,
Rose, M., and S. Waldbusser, "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.
- [RFC3291] Daniel, M., Haberman, B., Routhier, S., and J.
Schoenwaelder, "Textual Conventions for Internet Network
Addresses", RFC 3291, May 2002.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen "An
Architecture for Describing Simple Network Management
Protocol (SNMP) Management Frameworks", RFC 3411, December
2002.

--

-- NOTES TO RFC EDITOR (to be removed prior to publication)

--

-- The I-D <draft-ietf-ops-rfc3291bis-01.txt> (or a successor) is
-- expected to eventually replace RFC 3291. If that draft (or a

-- successor) is published as an RFC prior to, or concurrently with
-- this document, then the normative reference [RFC3291] should be
-- updated to point to the replacement RFC, and the reference tag
-- [RFC3291] should be updated to match.
--

[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.

11. 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)", RFC 3435, January 2003.

[RFCXYZ] Nechamkin, E., and Mule J., "Multimedia Terminal Adapter
(MTA) Management Information Base for PacketCable and
IPCablecom compliant devices", RFCXYZ, <Date>.

Editor's Note (to be removed prior to publication): This is the
reference to 'draft-ietf-ipcdn-pktdn-mtamib' which is expected to
accepted at the same time as this draft. When the draft is accepted,
the RFC Editor is asked to replace XYZ (and in the reference
earlier) with the assigned value and to remove this note.

[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
+1 858-404-2335
gordon.beacham@motorola.com

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

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

Disclaimer of validity

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.

Full Copyright Statement

Copyright (C) The Internet Society (2005). 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 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.

