

SCTE | **STANDARDS**

Data Standards Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 165-21 2016 (R2021)

IPCablecom 1.5 Part 21: Signaling Extension MIB

NOTICE

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

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

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

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

All Rights Reserved
© 2021 Society of Cable Telecommunications Engineers, Inc.
140 Philips Road
Exton, PA 19341

Note: DOCSIS® and PacketCable™ are registered trademarks of Cable Television Laboratories, Inc., and are used in this document with permission.

Document Types and Tags

Document Type: Specification

Document Tags:

- | | | |
|---|------------------------------------|--|
| <input type="checkbox"/> Test or Measurement | <input type="checkbox"/> Checklist | <input type="checkbox"/> Facility |
| <input type="checkbox"/> Architecture or Framework | <input type="checkbox"/> Metric | X Access Network |
| <input type="checkbox"/> Procedure, Process or Method | <input type="checkbox"/> Cloud | <input type="checkbox"/> Customer Premises |

Document Release History

| Release | Date |
|------------------|-------------------|
| SCTE 165-21 2009 | <i>10/5/2009</i> |
| SCTE 165-21 2016 | <i>10/31/2016</i> |

Note: This document is a reaffirmation of SCTE 165-21 2016. No substantive changes have been made to this document. Information components may have been updated such as the title page, NOTICE text, headers, and footers.

Contents

| | | |
|----------|---|-----------|
| 1 | SCOPE | 1 |
| 1.1 | PURPOSE OF THE DOCUMENT..... | 1 |
| 1.2 | REQUIREMENTS | 1 |
| 2 | REFERENCES | 2 |
| 2.1 | NORMATIVE REFERENCES | 2 |
| 2.2 | INFORMATIVE REFERENCES..... | 2 |
| 2.3 | REFERENCE ACQUISITION..... | 3 |
| 3 | ABBREVIATIONS | 3 |
| 4 | REQUIREMENTS | 4 |
| | APPENDIX A. LOOP VOLTAGE MANAGEMENT POLICIES | 18 |

This page intentionally left blank.

1 SCOPE

1.1 Purpose of the Document

New objects that are being introduced beyond IPCablecom 1.0 for Signaling MIBS are being grouped in this document so that the additional changes made can be tracked easily.

1.2 Requirements

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

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

2 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this standard. At the time of Subcommittee approval, the editions indicated were valid. All documents are subject to revision, and while parties to agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below, they are reminded that newer editions of those documents might not be compatible with the referenced version.

2.1 Normative References

- [1] ANSI/SCTE 165-05 2016, IPCablecom 1.5 Part 5: MTA Device Provisioning.
- [2] IETF STD 62, Simple Network Management Protocol Version 3 (SNMPv3), December 2002.
- [3] IETF RFC 2669, Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems.
- [4] ANSI/SCTE 23-03 2010, DOCSIS 1.1 Part 3: Operations Support System Interface
- [5] IETF STD 5, Internet Protocol, September 1981.
- [6] IETF RFC 2011, SNMPv2 Management Information Base for the Internet Protocol using SMIPv2, November 1996.
- [7] IETF RFC 2863, The Interfaces Group MIB, June 2000.
- [8] ANSI/SCTE 107 2009, Embedded Cable Modem Devices
- [9] CableLabs Definition MIB Specification, CL-SP-MIB-CLABDEF-I111-160317, March 17, 2016, Cable Television Laboratories, Inc.
- [10] ANSI/SCTE 79-02 2016, DOCSIS 2.0, Operations Support System Interface.
- [11] ANSI/SCTE 167-07 2016, IPCablecom 1.5 Part 7: MTA MIB.
- [12] ANSI/SCTE 165-08 2016, IPCablecom 1.5 Part 8: Signaling MIB.
- [13] ANSI/SCTE 165-06 2016, IPCablecom 1.5 Part 6: MIBs Framework Specification.
- [14] IETF RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Signals, May 2000.
- [15] ANSI/SCTE 165-02 2016, IPCablecom 1.5 Part 2: Audio/Video Codecs.
- [16] ANSI/SCTE 165-03 2016, IPCablecom 1.5 Part 3: Network-Based Call Signaling Protocol.

2.2 Informative References

In order to claim compliance with this standard, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this standard. Intellectual property rights may be required to implement these references.

- [17] Data-Over-Cable Service Interface Specifications, Cable Modem to Customer Premise Equipment Interface Specification, CMCI, CM-SP-CMCI-C01-081104, November 4, 2008, Cable Television Laboratories, Inc.
- [18] IETF RFC 3417, Transport Mappings for the Simple Network Management Protocol (SNMP), December 2002.
- [19] IETF RFC 2579, Textual Conventions for SMIPv2, April 1999.
- [20] IETF RFC 3410, Introduction and Applicability Statements for Internet-Standard Management Framework, December 2002.

- [21] IETF RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.
- [22] IETF RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), December 2002.
- [23] IETF RFC 2821, Simple Mail Transfer Protocol, April 2001.
- [24] SCTE 165-04 2016, IPCablecom 1.5 Part 4: Dynamic Quality-of-Service.
- [25] IETF RFC 3594, PacketCable Security Ticket Control Sub-Option for the DHCP CableLabs Client Configuration (CCC) Option, September 2003.
- [26] IETF RFC 2782, A DNS RR for specifying the location of services (DNS SRV), February 2000.
- [27] IETF RFC 3584, Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, August 2003.
- [28] ANSI/SCTE 165-15 2016, IPCablecom 1.5 Part 15, Management Event MIB Specification.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; Internet: <http://www.cablelabs.com/>.
- Internet Engineering Task Force (IETF) Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, Phone +1-703-620-8990, Fax +1-703-620-9071, Internet: www.ietf.org/

3 ABBREVIATIONS

There are no abbreviations used in this document.

4 REQUIREMENTS

The IPCablecom Extension Signaling MIB MUST be implemented as defined below.

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

```
IMPORTS
```

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

```
pktcEnSigMib MODULE-IDENTITY
```

```
    LAST-UPDATED "201208130000Z" -- August 13, 2012
    ORGANIZATION "Cable Television Laboratories, Inc "
```

```
    CONTACT-INFO
```

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

```
DESCRIPTION
```

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

```
Acknowledgements:
```

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

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

```
REVISION "201208130000Z" -- August 13, 2009
```

```
DESCRIPTION
```

```
    "Revised Version includes ECNs
    MIB-EXSIG1.5-N-12.0689-2-M
    and published as part of PKT-SP-MIB-EXSIG1.5-I05-121030"
```

REVISION "200906150000Z" -- June 15, 2009

DESCRIPTION

"Revised Version includes ECNs
MIB-EXSIG1.5-N-09.0569-2
and published as part of PKT-SP-MIB-EXSIG1.5-I04-090624"

REVISION "200704120000Z"

DESCRIPTION

"This revision is being published as part of the PacketCable
Signaling MIBs I03 enhancements for PacketCable 1.5."

REVISION "200508120000Z"

DESCRIPTION

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

::= { pktcEnhancements 2 }

--

-- Enhanced MIB Objects and Divisions.

--

```

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

```

--

-- Enhanced Notification groups.

--

```

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

```

pktcEnNcsMinimumDtmfPloyout OBJECT-TYPE

SYNTAX Unsigned32 (0 | 40..100)

UNITS "milliseconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object defines the minimum playout time for
the DTMF digit when IETF RFC 2833 DTMF Relay is used

for the egress gateway.
 If the value set via this `pktcEnNcsMinimumDtmfPloyout` object is different from that specified in RFC2833 packet, then the MTA MUST use the maximum of the two values.
 For example:
 If the RFC 2833 packet specifies 23ms and if the object `pktcEnNcsMinimumDtmfPloyout` is set to 40ms then the egress gateway must use a value of 40ms.
 Similarly if the RFC 2833 packet specifies 60 ms and if the object `pktcEnNcsMinimumDtmfPloyout` is set to 40ms then the egress gateway must use a value of 60 ms."

REFERENCE

"PacketCable(tm) Codec Specification"

DEFVAL {0}

::= { `pktcEnSigDevConfigObjects` 1 }

--

-- The following table enhances the NCS End Point Config Table
 -- (`pktcNcsEndPntConfigTable`) defined in `pktSigMib`.

--

--

```
pktcEnNcsEndPntConfigTable  OBJECT-TYPE
    SYNTAX          SEQUENCE OF PktcEnNcsEndPntConfigEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table augments pktcNcsEndPntConfigTable."
    ::= { pktcEnNcsEndPntConfigObjects 1 }
```

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

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

```
pktcEnNcsEndPntQuarantineState  OBJECT-TYPE
    SYNTAX  INTEGER {
        normal          (1),
        notification    (2),
        lockstep        (3),
        extendedlockstep (4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects the state of the Endpoint."
```

When the endpoint is in notification, lockstep or 'extended lockstep' states, the E-MTA MUST report the values of notification(2), lockstep(3) or extendedlockstep(4), respectively. Else, the endpoint MUST report a value of normal(1).

'Extended Lockstep' is defined as the state when the E-MTA is in the lockstep state for longer than 2 minutes. For more description about the states refer to the PacketCable Network Based Call signaling specification."

REFERENCE

"PacketCable(tm) Network-Based Call Signaling Protocol Specification,"

```
::= { pktcEnNcsEndPntConfigEntry 1 }
```

```
pktcEnNcsEndPntHookState OBJECT-TYPE
```

```
SYNTAX INTEGER {
    onHook (1),
    onHookPlusNCSActivity (2),
    offHook (3)
}
```

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

```
STATUS current
```

DESCRIPTION

"This object reflects the 'hook state' and 'NCS Activity' of an endpoint.

'NCS Activity', by definition includes: an active timeout signal, active brief signal or existence of an NCS connection.

The onHook(1) state indicates that the endpoint is 'on hook' and the absence of 'NCS Activity' on that endpoint.

The onHookPlusNCSActivity(2) indicates that the endpoint is 'on hook' and the presence of 'NCS Activity' on that endpoint.

The offHook(3) state indicates that the endpoint is 'off hook'."

REFERENCE

"PacketCable(tm) Network-Based Call Signaling Protocol Specification"

```
::= { pktcEnNcsEndPntConfigEntry 2 }
```

```
pktcEnNcsEndPntFaxDetection OBJECT-TYPE
```

```
SYNTAX TruthValue
```

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

```
STATUS current
```

DESCRIPTION

"This MIB object is used to configure the distinctive fax calling tone (CNG) detection feature on an MTA endpoint with reference to the analog interface. When set to true, the MTA MUST enable the detection of CNG tones on the specific endpoint.

When set to false, the MTA MUST disable the detection of CNG tones on the specific endpoint.

If a connection already exists on the endpoint when this MIB Object is modified, then the setting needs to take effect on the next connection."

```
DEFVAL {false}
```

```
::= { pktcEnNcsEndPntConfigEntry 3 }
```

```

pktcEnNcsEndPntStatusReportCtrl      OBJECT-TYPE
    SYNTAX INTEGER {
        unsupported          (1),
        reportActualStatus   (2),
        reportEndPointAsActive (3)
    }
    MAX-ACCESS read-create
    STATUS      deprecated
    DESCRIPTION
        "This MIB object is be used to control
        the Endpoint Status Reporting, if the feature is
        supported by the MTA and is configurable.
        The term 'Endpoint Status Reporting' refers to any
        information that the MTA may provide to External
        Systems for use in a particular reporting mechanism
        (Ex: Home Alarm Systems). The definition of the External
        Systems and reporting mechanism are beyond the scope
        of this definition (In the example of Home Alarm Systems,
        this MIB Object will allow Management Stations to
        temporarily disable outage reporting on an EndPoint
        during planned downtime).
        If supported, the MTA MUST:
        - reflect the actual Endpoint status when the value
          is set to 'reportActualStatus(2)'
        - reflect the EndPoint status as being active when the
          value is set to 'reportEndPointAsActive(3)',
          irrespective of the actual status.
        If unsupported, the MTA MUST set this value to
        'unsupported(1)' and reject any attempt to set
        this MIB object using SNMP SET to any other value."
 ::= { pktcEnNcsEndPntConfigEntry 4 }

pktcEnEndPntInfoTable      OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcEnEndPntInfoTableEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table includes any additional information
        associated with PacketCable EndPoints.
        The number of entries in this table represents the
        number of available PacketCable EndPoints."
 ::= { pktcEnNcsEndPntConfigObjects 2 }

pktcEnEndPntInfoTableEntry  OBJECT-TYPE
    SYNTAX      PktcEnEndPntInfoTableEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An entry in this table MUST be created for each
        PacketCable EndPoint.
        The index needs to be the corresponding index in the ifTable
        for the associated PacketCable EndPoint."
    INDEX { ifIndex }
 ::= { pktcEnEndPntInfoTable 1 }

PktcEnEndPntInfoTableEntry ::=
    SEQUENCE {
        pktcEnEndPntFgnPotSupport  BITS,
        pktcEnEndPntFgnPotDescr    SnmpAdminString,
        pktcEnEndPntClrFgnPotTsts  BITS,

```

```

        pktcEnEndPntRunFgnPotTsts    BITS,
        pktcEnEndPntFgnTestValidity  BITS,
        pktcEnEndPntFgnTestResults  BITS
    }

pktcEnEndPntFgnPotSupport  OBJECT-TYPE
    SYNTAX BITS {
        fgnPotDetection (0),
        hazardousFgnPotDetection (1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This MIB object indicates the capabilities of the MTA to
        detect various conditions related to the presence of
        foreign potential on an endpoint.
        The MTA MUST set a value of '1' for each bit corresponding
        to a supported functionality and a value of '0' for each
        bit corresponding to an unsupported functionality."
    ::= { pktcEnEndPntInfoTableEntry 1 }

pktcEnEndPntFgnPotDescr  OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This MIB object provides information related to the
        various tests for each detection mechanism supported by
        the MTA. While the actual contents are vendor-specific,
        the recommended format is:
        [<Capability>:<Test References>:<Other Info>]...
        Example:
        <fgnPotDetection>:<test XYZ, Reference 'Document'>:<NA>;
        <hazardousFngPotDetection>:<Test ABC, References>:<NA>
        "
    ::= { pktcEnEndPntInfoTableEntry 2 }

pktcEnEndPntClrFgnPotTsts  OBJECT-TYPE
    SYNTAX BITS {
        clrFgnPotentialResults (0),
        clrHazardousPotResults (1)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This MIB object is used to clear the current test
        results of supported conditions indicated by
        'pktcEnEndPntFgnPotSupport'.
        Setting a bit to a value of '1' clears the corresponding
        results in the MIB Object 'pktcEnEndPntFgnTestResults' and
        the validity as indicated by the MIB object
        'pktcEnEndPntFgnTestValidity' for the supported
        conditions only (i.e., the MTA MUST set the corresponding
        bits to a value of '0' in the indicated tables).
        If an SNMP SET attempts to set a bit corresponding to an
        unsupported condition to a value of '1', then the MTA MUST
        reject the entire SNMP SET and report an 'inconsistent
        value' error.
        For all unsupported scenarios, the corresponding bits MUST
        be set to a value of '0'."

```

Whenever one or more tests are enabled by the MIB Object 'pktcEnEndPntRunFgnPotTests', the MTA MUST also reset the corresponding bits in this MIB Object to a value of '0'."

```
::= { pktcEnEndPntInfoTableEntry 3}
```

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

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

```
pktcEnEndPntFgnTestValidity    OBJECT-TYPE
    SYNTAX BITS {
        fgnPotTstValidity (0),
        hazardousPotTstValidity (1)
    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This MIB object is used to indicate the validity of the
        corresponding test cases that were initiated using the MIB
        Object 'pktcEnEndPntRunFgnPotTests'.
        An MTA MUST:
        - return a value of '1' if the tests were run successfully
          and the results are valid.
        - return a value of '0' if a particular test was not
          initiated or if the tests could not be run successfully
          and hence the results are invalid.
        Note: The MTA MUST set all the BITS to '0' as soon as one
          or more test cases are initiated."
    ::= { pktcEnEndPntInfoTableEntry 5}
```

```

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

```

```

pktcEnNcsEndPntLVMgmtTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PktcEnNcsEndPntLVMgmtTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This MIB table contains the MIB Objects used for
        managing loop voltage on an MTA. An MTA MUST
        implement the defined MIB Objects and the associated
        functionality."
    ::= { pktcEnNcsEndPntConfigObjects 3 }

```

```

pktcEnNcsEndPntLVMgmtTableEntry OBJECT-TYPE
    SYNTAX PktcEnNcsEndPntLVMgmtTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry in this MIB table consists of the
        loop voltage management policy for the specified
        index. The MTA MUST use the ifIndex with a value of '1'
        to represent the E-MTA and the remaining indices
        (if used) to represent the endpoints (as specified
        in [1]).

        When the ifIndex is set to a value of '1', it represents
        a policy that MUST be applied to all the lines on an MTA.

        A MTA MUST support access to this MIB Object via the
        ifIndex set to a value of '1' (i.e., per-device policy). "
    REFERENCE "PacketCable Device Provisioning specification [1]"
    INDEX { ifIndex }
    ::= { pktcEnNcsEndPntLVMgmtTable 1 }

```

```

PktcEnNcsEndPntLVMgmtTableEntry ::=
    SEQUENCE {
        pktcEnNcsEndPntLVMgmtPolicy INTEGER,
        pktcEnNcsEndPntLVMgmtResetTimer Unsigned32,
        pktcEnNcsEndPntLVMgmtMaintTimer Unsigned32
    }

```

```

pktcEnNcsEndPntLVMgmtPolicy OBJECT-TYPE

```



```
SYNTAX INTEGER {  
    voltageAtAllTimes(1),  
    voltageUnlessRFQAMAbsent(2),  
    voltageBasedOnServiceOrTimers(3),  
    voltageBasedOnService(4)  
}  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION
```

"This MIB Object allows the Service Provider to choose a suitable policy for Loop Voltage behavior on MTAs.

Unless overridden by the operator, the MTA MUST use the default value specified in the definition of this MIB Object.

The MTA MUST adhere to PacketCable signaling requirements, such as the NCS open loop voltage requirement, irrespective of any chosen policy.

For MTAs that need to remove loop voltage during the MTA initialization phase, in contradiction to a chosen policy, such a loop voltage removal period MUST NOT exceed 1000ms.

The MTA MUST retain the value of this MIB Object across hard reboots or soft resets (for a definition of the terms, please refer to [1]).

This MIB object specifies four policies. An informative illustration of the E-MTA behavior with different policies is presented in Appendix A.

Provisioned line is an MTA Endpoint that has been provided with valid per-line configuration data either via the configuration file (during provisioning) or the SNMP management interface.

When the MIB Object is set to a value of 'voltageAtAllTimes', - indicating Policy 1 - the MTA MUST maintain the loop idle voltage on all lines, irrespective of the line status.

When this MIB Object is set to a value of 'voltageUnlessRFQAMAbsent' - indicating Policy 2 - the MTA MUST apply loop idle voltage at all times except when it confirms the absence of any RF QAM carrier, following a complete scan of the spectrum (i.e., loop voltage is maintained during the scan). When the MTA detects the presence of any RF QAM carrier, it MUST apply the loop idle voltage. Additionally, the following conditions apply:

- during a hard reboot, this policy applies at all lines until the MTA is successfully provisioned (i.e., pktcMtaDevProvisioningState has a value other than 'InProgress')
- once the MTA is successfully provisioned (i.e., pktcMtaDevProvisioningState has a value of 'pass(1)', 'passWithWarnings(4)' or 'passWithIncompleteParsing(5)') then the policy

- applies to all provisioned lines
- upon the onset of a re-initialization due to a soft reset (via SNMP or Rf conditions) the MTA MUST continue to maintain the existing policy and state on previously provisioned lines, unless overridden by a policy or the provisioning process specifies otherwise

This policy is similar to Policy 1, except for the ability to recognize events like cable cuts (due to malicious activities, or otherwise).

The following requirements apply to policies 3 and 4:

- upon the onset of a re-initialization due to a soft reset the MTA MUST continue to maintain the existing policy and state on provisioned lines, unless overridden by a policy or the provisioning process specifies otherwise
- once the provisioning process is completed with the value of the MIB Object `pktcMtaDevProvisioningState` set to a value of 'pass(1)', 'passWithWarnings(4)' or 'passWithIncompleteParsing(5)', the MTA MUST apply the chosen policy to all the provisioned lines

When this MIB Object is set to a value of 'voltageBasedOnServiceOrTimers' - indicating Policy 3 - the MTA MUST adhere to the requirements that follow at any given point in time:

- when the timer defined by `pktcEnNcsEndPntLVMgmtResetTimer` has a non-zero value, the MTA MUST apply loop idle voltage under all circumstances (similar to policy 1). Refer to the definition of the MIB Object for the persistence and timer requirements.
- when the timer defined by `pktcEnNcsEndPntLVMgmtMaintTimer` has a non-zero value, the MTA MUST maintain a line's loop idle voltage state that was in effect prior to the timer being set to a non-zero value. Refer to the definition of the MIB Object defining the timer for the persistence and timer requirements.
- When the timers defined by `pktcEnNcsEndPntLVMgmtMaintTimer` and `pktcEnNcsEndPntLVMgmtResetTimer` have expired (both have a value of zero), then:
 - = the MTA MUST apply loop voltage if the provisioning process is completed with the value of `pktcMtaDevProvisioningState` set to a value of than 'pass(1)', 'passWithWarnings(4)' or 'passWithIncompleteParsing(5)'
 - = During a T4 timeout (note: the timers have expired), the E-MTA MUST remove loop idle voltage on all lines
- When both the timers are active (i.e., they both have non-zero values), then the timer defined by the MIB Object `pktcEnNcsEndPntLVMgmtMaintTimer` takes precedence
- if none of the above cases apply, the MTA MUST remove loop idle voltage on all lines

When the MIB Object is set to a value of 'voltageBasedOnService' - indicating Policy 4 - the following conditions apply at any given point in time:

- the MTA MUST apply loop idle voltage to all the provisioned lines if the value of the MIB Object 'pktcMtaDevProvisioningState' is set to a value of 'pass(1)', 'passWithWarnings(4)' or 'passWithIncompleteParsing(5)'

- in all other cases, the MTA MUST remove loop idle voltage on all lines."

REFERENCE "PacketCable Device Provisioning specification [1]"
DEFVAL { voltageBasedOnService }
::= { pktcEnNcsEndPntLVMgmtTableEntry 1}

pktcEnNcsEndPntLVMgmtResetTimer OBJECT-TYPE
SYNTAX Unsigned32 (0..1440)
UNITS "minutes"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This MIB Object specifies the time duration allowed for an MTA to successfully provision and is only applicable when the MIB Object pktcEnNcsEndPntLVMgmtPolicy is set to a value of 'voltageBasedOnServiceOrTimers'. In all other cases, the MTA MUST:
- return a value of '0' upon any retrieval requests
- return an error of 'inconsistentValue' upon any modification requests

The value contained by this MIB Object is a countdown timer and the MTA MUST start counting down the configured value only upon a hard reboot, a soft reset or a T4 timeout. Once this timer has reached a value of zero, the MTA MUST retain the value (of zero) until successfully configured otherwise. The MTA MUST use a change in the value of this MIB Object only on the next hard reboot, soft reset or T4 timeout.

The MTA MUST persist the last configured value (i.e., not the countdown value) of this MIB Object across hard reboots and soft resets.

Refer to the MIB Object pktcEnNcsEndPntLVMgmtPolicy for usage within 'voltageBasedOnServiceOrTimers'."

DEFVAL { 5 }
::= { pktcEnNcsEndPntLVMgmtTableEntry 2}

pktcEnNcsEndPntLVMgmtMaintTimer OBJECT-TYPE
SYNTAX Unsigned32 (0..1440)
UNITS "minutes"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This MIB Object allows the operator to specify the time that loop voltage condition will be maintained, irrespective of the changes to the MTA. It is only applicable when the MIB Object pktcEnNcsEndPntLVMgmtPolicy is set to a value of 'voltageBasedOnServiceOrTimers'. In all

other cases, the MTA MUST:

- return a value of '0' upon any retrieval requests
- return an error of 'inconsistentValue' upon any modification requests

The value contained in this MIB Object is a countdown timer and the MTA MUST start counting down the value immediately after a successful configuration to a non-zero value. Once this timer has reached a value of zero, the MTA MUST retain the value (of zero) until successfully configured otherwise.

The MTA MUST retain the value of this MIB Object (i.e., the countdown value) across soft resets. The MTA MUST reset the value of this MIB Object (to its default value) during a hard reboot of the MTA.

Refer to the MIB Object `pktcEnNcsEndPntLVMgmtPolicy` for information about applicability and usage."

```
DEFVAL { 0 }
 ::= { pktcEnNcsEndPntLVMgmtTableEntry 3 }
```

```
--
```

```
-- The Provisioned Loss Plan Table
```

```
--
```

```
pktcEnNcsEndPntLossTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF PktcEnNcsEndPntLossEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

```
    " The E-MTA supports two provisioned loss parameters, one for
      the D/A direction (towards the subscriber) and one for A/D
      direction (from the subscriber) direction. This data table
      represents the loss for each line provided by the E-MTA."
```

```
 ::= { pktcEnNcsEndPntConfigObjects 4 }
```

```
pktcEnNcsEndPntLossEntry OBJECT-TYPE
```

```
SYNTAX      PktcEnNcsEndPntLossEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

```
    " The E-MTA supports two provisioned loss parameters, one for
      the D/A direction (towards the subscriber) and one for A/D
      direction (from the subscriber) direction. This data table
      represents the loss for each line provided by the E-MTA."
```

```
    Each entry in this data table describes the loss
    for the associated line."
```

```
INDEX { ifIndex }
 ::= { pktcEnNcsEndPntLossTable 1 }
```

```
PktcEnNcsEndPntLossEntry ::=
```

```
    SEQUENCE {
        pktcEnNcsEndPntLossDA      Integer32,
        pktcEnNcsEndPntLossAD      Integer32
    }
```

```
pktcEnNcsEndPntLossDA OBJECT-TYPE
```

```
SYNTAX      Integer32 (0..12)
UNITS       "dB"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

```
" This object specifies the provisioned loss parameter
for the D/A direction (towards the subscriber) in dB.

The E-MTA MUST implement this element per the PacketCable Embedded
MTA Analog Interface and Powering Specification."
REFERENCE "PacketCable AIP Specification"
DEFVAL {9}
::= { pktcEnNcsEndPntLossEntry 1 }

pktcEnNcsEndPntLossAD OBJECT-TYPE
SYNTAX Integer32 (0..6)
UNITS "dB"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
" This object specifies the provisioned loss parameter
for the A/D direction (from the subscriber)in dB.

The E-fMTA MUST implement this element per the PacketCable Embedded
MTA Analog Interface and Powering Specification."
REFERENCE "PacketCable AIP Specification"
DEFVAL {3}
::= { pktcEnNcsEndPntLossEntry 2 }

--
-- Compliance statements
--
pktcEnSigBasicCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The compliance statement for devices that implement
PacketCable defined Signaling on an MTA."

MODULE PKTC-EN-SIG-MIB

--
-- Mandatory groups
--
MANDATORY-GROUPS {
    pktcEnSigGroup
}
GROUP pktcEnNcsGroup
DESCRIPTION
"This group is mandatory for any MTA implementing
PacketCable signaling."

GROUP pktcEnNcsLVMgmtGroup
DESCRIPTION
"This group is mandatory for any MTA implementing
PacketCable signaling."

::={ pktcEnSigCompliances 1 }

--
-- Conformance group for common Signaling.
--
pktcEnSigGroup OBJECT-GROUP
OBJECTS {
```

```
        pktcEnNcsMinimumDtmfPayout
    }
    STATUS current
    DESCRIPTION
        "Enhanced group of objects for the common portion of the
        PacketCable Signaling MIB."
    ::= { pktcEnSigGroups 1 }

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

--
-- Conformance group for Loop Voltage Management
--
pktcEnNcsLVMgmtGroup OBJECT-GROUP
    OBJECTS {
        pktcEnNcsEndPntLVMgmtPolicy,
        pktcEnNcsEndPntLVMgmtResetTimer,
        pktcEnNcsEndPntLVMgmtMaintTimer
    }
    STATUS current
    DESCRIPTION
        "Enhanced group of objects for the loop voltage
        Management of PacketCable MTAs based on Signaling
        and configured policies."
    ::= { pktcEnSigGroups 3 }

pktcEnNcsDeprecatedGroup OBJECT-GROUP
    OBJECTS {
        pktcEnNcsEndPntStatusReportCtrl
    }
    STATUS deprecated
    DESCRIPTION
        "This contains a list of deprecated Extension
        Signaling MIB Objects."
    ::= { pktcEnSigGroups 4 }

END
```

Appendix A. Loop Voltage Management Policies

This appendix provides an illustration of the loop voltage management policies defined by the MIB Object `pktcEnNcsEndPntLVMgmtPolicy`. Please note that this is an informative illustration and the MIB Object definition is normative under all circumstances.

| Stage | Policy 1 | Policy 2 | Policy 3 | Policy4 (default) |
|---|-------------------------------|--|---|-------------------------------|
| Boot loader initialization following a hard reboot | unspecified | unspecified | unspecified | unspecified |
| Boot loader initialization following a soft reset | Same status as prior to reset | Same status as prior to reset | Same status as prior to reset | Same status as prior to reset |
| DOCSIS RF Scan | APPLY voltage on all lines | APPLY for first full pass through spectrum, REMOVE if no QAM lock occurred; REAPPLY if QAM returns | APPLY until Maintenance Timer and Voltage Reset Timeout expire; REMOVE thereafter | REMOVE voltage |
| eCM Ranging/Registering | APPLY | APPLY | Same status as final state from scanning; REMOVE if timers expire; | REMOVE |
| eCM provisioning process | APPLY | Same status as final state from scanning | Same status as final state from scanning; REMOVE if timers expire | REMOVE |
| Loss of Rf signal prior to T4 | APPLY | APPLY | APPLY | APPLY |
| DOCSIS Rf scanning after T4 | APPLY | APPLY for first full pass through spectrum, REMOVE if no QAM lock occurred; REAPPLY if QAM returns | APPLY until timers expire, REMOVE thereafter | REMOVE |
| eMTA provisioning (MTA1 onwards, while <code>pktcMtaDevProvisioningState</code> is set to a value of 'InProgress') | APPLY | Same status as final state from Rf scanning | APPLY until timers expire, REMOVE thereafter | REMOVE |
| E-MTA is operational and the eMTA MIB Object <code>pktcMtaDevProvisioningState</code> is set to a value of 'pass', 'passWithWarnings' or 'passWithIncompleteParsing' | APPLY | APPLY | APPLY | APPLY |
| E-MTA is operational, the eMTA provisioning is completed, but the MTA's MIB Object <code>pktcMtaDevProvisioningState</code> is set to a value *other than * 'pass', 'passWithWarnings' or 'passWithIncompleteParsing' | APPLY | APPLY | APPLY until timers expire, REMOVE thereafter | REMOVE |