

SCTE • ISBE[®]

S T A N D A R D S

Network Operations Subcommittee

SCTE STANDARD

SCTE 154-3 2018

Encoder MIB

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) / International Society of Broadband Experts (ISBE) 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, interchangeability, best practices and ultimately the long-term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE•ISBE 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•ISBE members.

SCTE•ISBE 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.

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE•ISBE 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•ISBE web site at <http://www.scte.org>.

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

CONTENTS

SCOPE.....	4
COPYRIGHT.....	4
NORMATIVE REFERENCE	4
INFORMATIVE REFERENCE.....	4
TERMS AND DEFINITIONS	4
REQUIREMENTS	4

SCOPE

This document is identical to SCTE 154-3 2008 except for informative components which may have been updated such as the title page, NOTICE text, headers and footers. No normative changes have been made to this document.

This document provides the branch object identifiers for each of the MIBs within the SCTE HMS HEADENDIDENT Tree.

COPYRIGHT

The MIB definition found in this document may be incorporated directly in products without further permission from the copyright owner, SCTE.

NORMATIVE REFERENCE

IETF RFC 2578 SNMPv2-SMI
IETF RFC 2579 SNMPv2-TC
IETF RFC 2863 IF-MIB
IETF RFC 4001 INET-ADDRESS-MIB
IETF RFC 4133 ENTITY-MIB
SCTE 36 2002R2007 (formerly HMS028) SCTE-ROOT
SCTE 37 2007 (formerly HMS072) SCTE-HMS-ROOTS

INFORMATIVE REFERENCE

HE MPEG MIBs Overview Document – Number TBD

TERMS AND DEFINITIONS

This document defines the following terms:

Management Information Base (MIB) – the specification of information in a manner that allows standard access through a network management protocol.

REQUIREMENTS

This section defines the mandatory syntax of the SCTE-HMS-HEADENDIDENT-MIB. It follows the IETF Simple Network Management Protocol (SNMP) for defining managed objects.

The syntax is given below.

SCTE-HMS-MPEG-ENCODER-MIB DEFINITIONS ::= BEGIN

IMPORTS

OBJECT-TYPE, MODULE-IDENTITY, TimeTicks,
 Unsigned32, Integer32, enterprises, Counter32
 FROM SNMPv2-SMI
 OBJECT-GROUP, MODULE-COMPLIANCE
 FROM SNMPv2-CONF
 DisplayString
 FROM SNMPv2-TC
 InetAddressPrefixLength, InetPortNumber, InetAddress,
 InetAddressType
 FROM INET-ADDRESS-MIB
 InterfaceIndex
 FROM IF-MIB
 entPhysicalIndex, PhysicalIndex
 FROM ENTITY-MIB
 DeviceEnableDisableValues, MpegErrorStatus,
 HePIDValue, VideoInputFrameRateType
 FROM SCTE-HMS-HEADENDIDENT-TC;

mpegEncoderMIB MODULE-IDENTITY

LAST-UPDATED "200710031700Z"

ORGANIZATION

"SCTE HMS Working Group"

CONTACT-INFO

"SCTE HMS Subcommittee, Chairman

mail to: standards@scte.org"

DESCRIPTION

"The MIB module is for representing an MPEG2 or MPEG4-AVC video
 encoder present in the headend (or indoor) and supported by an SNMP
 agent. Multiple encoders sharing a single agent are also supported,
 the encoders being identified by entPhysicalIndex and unique
 ifIndexes for their interfaces.

This MIB assumes there is one video source per encoder.
 If a physical unit encodes more than one video source at
 a time it shall be considered to be a group of encoders sharing
 a common chassis and each encoder treated as a discrete physical
 entity.

An encoder may encode more than one stream from a single source."

REVISION "200710031700Z"

DESCRIPTION

" Added SCTE-HMS-HEADENDIDENT-TC name to file"

REVISION "200709281545Z"

DESCRIPTION

" Prepare MIB for ballot"

::= { enterprises scteRoot(5591) scteHmsTree (1) insidePlantIdent (11)
 heDigital (5) heDigitalMpegGroups (5) 1}

mpegEncoderMIBObjects OBJECT IDENTIFIER ::= { mpegEncoderMIB 1 }

configurationReport OBJECT IDENTIFIER ::= { mpegEncoderMIBObjects 1 }

```

inputMonitor          OBJECT IDENTIFIER ::= { mpegEncoderMIBObjects 2 }
outputMonitor         OBJECT IDENTIFIER ::= { mpegEncoderMIBObjects 3 }

-- Conformance Information

mpegEncoderMIBConformance OBJECT IDENTIFIER ::= { mpegEncoderMIB 2 }

-- The minimum compliance statement for MPEG encoders.

mpegEncoderMIBCompliances OBJECT IDENTIFIER ::= { mpegEncoderMIBConformance 1 }
mpegEncoderMIBGroups      OBJECT IDENTIFIER ::= { mpegEncoderMIBConformance 2 }

-- *****
-- ***** CONFIGURATION REPORT *****
-- *****

-- The tables in this section define the output from the encoder. Each
-- component is referenced to the service to which it belongs. Each
-- service is referenced to the transport stream in which it is carried
-- and each transport stream is referenced to the physical and logical
-- output.
-- If a component is used in more than one service then the table row
-- should be repeated with the service cross reference pointing to the
-- alternative service. Likewise services output on more than one
-- transport streams should have a table row for each instance with
-- appropriate referencing to each transport stream.

encoderCfgVideoTable OBJECT-TYPE
    SYNTAX SEQUENCE OF EncoderCfgVideoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table shows the video configuration of this encoder."
    ::= { configurationReport 1 }

encoderCfgVideoEntry OBJECT-TYPE
    SYNTAX EncoderCfgVideoEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry containing management information applicable to a
        particular encoder's video configuration. This table is indexed by
        the entPhysicalIndex of the encoder, and encoderCfgVideoIndex
        to allow for multiple video channels. All video channels shall be
        derived from a single source and consist of a primary and auxiliary
        videos such as PIP. If more than one primary video is encoded within
        the chassis then it shall be treated as a multiple encoder chassis."
    INDEX { entPhysicalIndex,
            encoderCfgVideoIndex }
    ::= { encoderCfgVideoTable 1 }

EncoderCfgVideoEntry ::= SEQUENCE {
    encoderCfgVideoIndex
        Unsigned32,
    encoderCfgVideoType
        INTEGER,

```

```

encoderCfgVideoCompression
    INTEGER,
encoderCfgVideoPid
    HePIDValue,
encoderCfgVideoPcrPid
    HePIDValue,
encoderCfgVideoServiceIndex
    Unsigned32,
encoderCfgVideoVertResolution
    Integer32,
encoderCfgVideoHorzResolution
    Integer32,
encoderCfgVideoBitrateAvg
    Integer32,
encoderCfgVideoBitrateMax
    Integer32,
encoderCfgVideoBitrateMin
    Integer32,
encoderCfgVideoFilmMode
    DeviceEnableDisableValues ,
encoderCfgVideoRateMode
    INTEGER,
encoderCfgVideoBorderProcessing
    INTEGER,
encoderCfgVideoPesAlignment
    INTEGER,
encoderCfgVideoCodingDelay
    Integer32,
encoderCfgVideoDeblockEnable
    DeviceEnableDisableValues,
encoderCfgVideoDeblockAlpha
    Integer32,
encoderCfgVideoDeblockBeta
    Integer32,
encoderCfgVideoIdrRate
    Integer32,
encoderCfgVideoInputIf
    INTEGER,
encoderCfgVideoInputFrameRate
    VideoInputFrameRateType,
encoderCfgVideoInputScan
    INTEGER,
encoderCfgVideoInputFormat
    INTEGER,
encoderCfgVideoAspectRatio
    INTEGER
}

encoderCfgVideoIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Identifier index for each video channel (from a single source)
        provided by this encoder."
    ::= { encoderCfgVideoEntry 1 }

```

encoderCfgVideoType OBJECT-TYPE

SYNTAX INTEGER {

other (1),
primary (2),
pip (3),
dvbH (4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the video is encoded into more than one component this object defines the type of this component. If one video component encoded then this object shall always be primary.

mpeg2 - ISO/IEC 13818 part 2 video

mpeg4 - ISO/IEC 14496 part 2 video

avc - ITU-T H.264 video"

::= { encoderCfgVideoEntry 2 }

encoderCfgVideoCompression OBJECT-TYPE

SYNTAX INTEGER {

other (1),
mpeg2 (2),
mpeg4 (3),
avc (4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Video encoding standard

mpeg2 - ISO/IEC 13818 part 2 video

mpeg4 - ISO/IEC 14496 part 2 video

avc - ITU-T H.264 video"

::= { encoderCfgVideoEntry 3 }

encoderCfgVideoPid OBJECT-TYPE

SYNTAX HePIDValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The PID carrying the primary video component."

::= { encoderCfgVideoEntry 4 }

encoderCfgVideoPcrPid OBJECT-TYPE

SYNTAX HePIDValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The PID of the PCR used by this component."

::= { encoderCfgVideoEntry 5 }

encoderCfgVideoServiceIndex OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "When multiple services/transport streams are supported by this encoder, this field provides a cross reference to a specific service based on the encoderCfgServiceInstIndex in the encoderCfgServiceTable.
 When this component is output as an elementary stream without PAT/PMT reference, this field provides an indirect cross reference to the output transport stream"
 ::= { encoderCfgVideoEntry 6 }

encoderCfgVideoVertResolution OBJECT-TYPE
 SYNTAX Integer32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Number of active video lines used in encoding.
 Values include, but are not restricted to
 240, 288, 480, 512, 576, 608, 720 and 1080"
 ::= { encoderCfgVideoEntry 7 }

encoderCfgVideoHorzResolution OBJECT-TYPE
 SYNTAX Integer32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Number of pixels per line used in encoding.
 Values include, but are not restricted to
 352, 480, 528, 544, 640, 704, 720, 1280, 1440, 1920"
 ::= { encoderCfgVideoEntry 8 }

encoderCfgVideoBitrateAvg OBJECT-TYPE
 SYNTAX Integer32
 UNITS "bps"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Average or nominal bit rate for video component expressed in Transport rate. If component is output as elementary stream The encoder shall convert the actual rate to equivalent hypothetical transport rate for compatibility with down stream monitoring points "
 ::= { encoderCfgVideoEntry 9 }

encoderCfgVideoBitrateMax OBJECT-TYPE
 SYNTAX Integer32
 UNITS "bps"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Maximum allowed bit rate for video component when stat mux/VBR selected. Equal to average when CBR."

```
::= { encoderCfgVideoEntry 10 }
```

encoderCfgVideoBitrateMin OBJECT-TYPE

SYNTAX Integer32

UNITS "bps"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Minimum allowed bit rate for video component when
stat mux/VBR selected. Equal to average when CBR."

```
::= { encoderCfgVideoEntry 11 }
```

encoderCfgVideoFilmMode OBJECT-TYPE

SYNTAX DeviceEnableDisableValues

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Normal frame encoding or 3-2 pulldown allowed."

```
::= { encoderCfgVideoEntry 12 }
```

encoderCfgVideoRateMode OBJECT-TYPE

SYNTAX INTEGER {

cbr (1),

vbr (2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Fixed bit rate, stat mux or other variable rate mode selected."

```
::= { encoderCfgVideoEntry 14 }
```

encoderCfgVideoBorderProcessing OBJECT-TYPE

SYNTAX INTEGER {

disabled (1),

cropped (2),

other(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Border processing selected."

```
::= { encoderCfgVideoEntry 15 }
```

encoderCfgVideoPesAlignment OBJECT-TYPE

SYNTAX INTEGER {

other (1),

aligned (2),

nonAligned (3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```

        "PES to transport stream alignment."
        ::= { encoderCfgVideoEntry 16 }

encoderCfgVideoCodingDelay OBJECT-TYPE
    SYNTAX      Integer32
    UNITS        "milliseconds"
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Video delay due to encoding, frame capture to PTS.
        Zero indicates parameter not reported."
        ::= { encoderCfgVideoEntry 17 }

encoderCfgVideoDeblockEnable OBJECT-TYPE
    SYNTAX      DeviceEnableDisableValues
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Deblock filter enable/disable."
        ::= { encoderCfgVideoEntry 18 }

encoderCfgVideoDeblockAlpha OBJECT-TYPE
    SYNTAX      Integer32 (-6..6)
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Deblock filter strength - edge offset."
        ::= { encoderCfgVideoEntry 19 }

encoderCfgVideoDeblockBeta OBJECT-TYPE
    SYNTAX      Integer32 (-6..6)
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Deblock filter strength - edge neighbor offset."
        ::= { encoderCfgVideoEntry 20 }

encoderCfgVideoIdrRate OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Rate IDR pictures are sent, expressed in frames.Say AVC"
        ::= { encoderCfgVideoEntry 21 }

encoderCfgVideoInputIf OBJECT-TYPE
    SYNTAX      INTEGER {
        other (1),
        analog (2),
        sdi (3),
        hdsdi (4) }
    MAX-ACCESS  read-only
    STATUS       current
    DESCRIPTION
        "Video input standard selected."
        ::= { encoderCfgVideoEntry 22 }

```

```

encoderCfgVideoInputFrameRate  OBJECT-TYPE
    SYNTAX      VideoInputFrameRateType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Video input rate selected. autoSelect indicates encoder
        can accept most frame rates without specific selection.
        f29or30Hz and f59or60Hz indicate limited auto-select
        between 29.97Hz and 30Hz or 59.94Hz and 60Hz supported
        by the encoder."
    ::= { encoderCfgVideoEntry 23 }

encoderCfgVideoInputScan  OBJECT-TYPE
    SYNTAX      INTEGER {
        interlaced (1),
        progressive (2)  }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Video input scanning selected."
    ::= { encoderCfgVideoEntry 24 }

encoderCfgVideoInputFormat  OBJECT-TYPE
    SYNTAX      INTEGER {
        component (1),
        composite (2)    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Video input format selected."
    ::= { encoderCfgVideoEntry 25 }

encoderCfgVideoAspectRatio  OBJECT-TYPE
    SYNTAX      INTEGER {
        other (1),
        aspect4x3 (2),
        aspect16x9 (3)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Input video aspect ratio"
    ::= { encoderCfgVideoEntry 26 }

encoderCfgAudioTable  OBJECT-TYPE
    SYNTAX SEQUENCE OF EncoderCfgAudioEntry
    MAX-ACCESS not-accessible
    STATUS      current
    DESCRIPTION
        "This table shows the audio configuration of this encoder."
    ::= { configurationReport 2 }

encoderCfgAudioEntry  OBJECT-TYPE
    SYNTAX      EncoderCfgAudioEntry
    MAX-ACCESS  not-accessible
    STATUS      current

```

```

DESCRIPTION
    "An entry containing management information applicable to a
    particular encoder's audio components. This table is indexed
    by the entPhysicalIndex of the encoder and the
    encoderCfgAudioChannelIndex for each audio component carried by
    this encoder."
INDEX { entPhysicalIndex,
        encoderCfgAudioChannelIndex }
 ::= { encoderCfgAudioTable 1 }

EncoderCfgAudioEntry ::= SEQUENCE {
    encoderCfgAudioChannelIndex
        Unsigned32,
    encoderCfgAudioEnabled
        DeviceEnableDisableValues,
    encoderCfgAudioStandard
        INTEGER,
    encoderCfgAudioPid
        HePIDValue,
    encoderCfgAudioPcrPid
        HePIDValue,
    encoderCfgAudioServiceIndex
        Unsigned32,
    encoderCfgAudioMode
        INTEGER,
    encoderCfgAudioPassThru
        INTEGER,
    encoderCfgAudioRate
        Integer32,
    encoderCfgAudioLanguageA
        DisplayString,
    encoderCfgAudioLanguageB
        DisplayString,
    encoderCfgAudioLipSync
        INTEGER,
    encoderCfgAudioDialogNorm
        Integer32,
    encoderCfgAudioInputType
        INTEGER,
    encoderCfgAudioInputLabel
        DisplayString
}

encoderCfgAudioChannelIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Identifier index for each audio output channel provided by
        this encoder. This number shall be unique within an encoder."
    ::= { encoderCfgAudioEntry 1 }

encoderCfgAudioEnabled OBJECT-TYPE
    SYNTAX      DeviceEnableDisableValues
    MAX-ACCESS  read-only
    STATUS      current

```

```

DESCRIPTION
    "Shows if this audio channel is configured for service."
    ::= { encoderCfgAudioEntry 2 }

encoderCfgAudioStandard OBJECT-TYPE
    SYNTAX      INTEGER {
        other (1),
        ac3 (2),
        eac3 (3),
        mpeg1 (4),
        mpeg2 (5),
        aacMpeg2 (6),
        heAacMpeg2 (7),
        aacMpeg4 (8),
        heAacMpeg4 (9)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Audio encoding standard for this channel. Values identified
        against 13818-1 stream_types as follows:
        ac3          stream_type 0x81/0x06(DVB)
        eac3          stream_type 0x87
        mpeg1          stream_type 0x03
        mpeg2          stream_type 0x04
        aacMpeg2 stream_type 0x0F
        heAacMpeg2 stream_type 0x0F
        aacMpeg4 stream_type 0x11
        heAacMpeg4 stream_type 0x11"
    ::= { encoderCfgAudioEntry 3 }

encoderCfgAudioPid OBJECT-TYPE
    SYNTAX      HePIDValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Selected PID for this audio component."
    ::= { encoderCfgAudioEntry 4 }

encoderCfgAudioPcrPid OBJECT-TYPE
    SYNTAX      HePIDValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The PCR used by this component."
    ::= { encoderCfgAudioEntry 5 }

encoderCfgAudioServiceIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "When multiple services/transport streams are supported
        by this encoder, this field provides a cross reference to
        a specific service based on the encoderCfgServiceInstIndex
        in the encoderCfgServiceTable.
        When this component is output as an elementary stream

```