SCTE | STANDARDS

Network Operations Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 84-1 2017 (R2022)

HMS Common Inside Plant Management Information Base (MIB) Part 1: SCTE-HMS-HE-COMMON-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.

140 Philips Road
Exton, PA 19341

DOCUMENT TYPES AND TAGS

Document Type: Specification

Document Tags:

☐ Test or Measurement
☐ Architecture or Framework
☐ Procedure, Process or Method
☐ Checklist
☐ Metric
☐ Cloud
☐ Facility
☒ Access Network
☐ Customer Premises

DOCUMENT RELEASE HISTORY

Release	Date
SCTE 84-1 2003	*05/09/2003*
SCTE 84-1 2009	*07/10/2009*
SCTE 84-1 2017	*08/28/2017*

Note: Standards that are released multiple times in the same year use: a, b, c, etc. to indicate normative balloted updates and/or r1, r2, r3, etc. to indicate editorial changes to a released document after the year.

Note: This document is a reaffirmation of SCTE 84-1 2017. 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

SCOPE

This document is identical to SCTE 84-1 2009 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.

The MIB module is for representing general information about optical equipment present in the headend (or indoor) and is supported by an SNMP agent.

COPYRIGHT

NORMATIVE REFERENCE

IETF RFC2578, Structure of Management Information Version 2 (SMIv2)

IETF RFC2579, Textual Conventions for SMIv2

IETF RFC2580, Conformance Statements for SMIv2

IETF RFC2737, Entity MIB (Version 2)

ANSI/SCTE 36 ,SCTE Root Management Information Block (MIB)

SCTE 38-11, Hybrid Management Sub-layer Management Information Base (MIB) Part 11: SCTE-HMS-HEADENDIDENT-MIB

IETF RFC2573, SNMP Applications

IETF RFC1907, Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)

ANSI/SCTE 38-1, Hybrid Management Sublayer Management Information Blocks (MIB) Part 1: Property MIB

INFORMATIVE REFERENCE

None

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-HE-COMMON-MIB. It follows the IETF Simple Network Management Protocol (SNMP) for defining managed objects.

The syntax is given below:

```
-- Module Name: HMS111R9.MIB (SCTE 84-1)
-- SCTE Status: Adopted

SCTE-HMS-HE-COMMON-MIB DEFINITIONS ::= BEGIN

IMPORTS
   Integer32, MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE
      FROM SNMPv2-SMI
   DisplayString, DateAndTime
      FROM SNMPv2-TC
   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF
   entPhysicalIndex
      FROM ENTITY-MIB
   scteHmsTree
      FROM SCTE-ROOT  -- see SCTE 36 (formerly HMS028)
   heCommon, HeTenthCentigrade
      FROM SCTE-HMS-HEADENDIDENT-MIB; -- see SCTE 38-11 (formerly HMS114)

heCommonMib MODULE-IDENTITY
   LAST-UPDATED "200302170000Z" -- February 17, 2003
   ORGANIZATION "SCTE HMS Working Group"
   CONTACT-INFO
         " SCTE HMS Subcommittee, Chairman
           mailto:standards@scte.org
      "
   DESCRIPTION
         "The MIB module is for representing general information
         about optical equipment present in the headend (or indoor)
         and is supported by an SNMP agent."

   REVISION "200302170000Z"  -- February 17, 2003
   DESCRIPTION
      "
      Incorporated RTF comments posted by January 10, 2003.
      "
```

```
    ::= { heCommon 1 }

heCommonObjects OBJECT IDENTIFIER ::= { heCommonMib 1 }

-- MIB contains 2 groups
heCommonParams  OBJECT IDENTIFIER ::= { heCommonObjects 1 }
heCommonLog     OBJECT IDENTIFIER ::= { heCommonObjects 2 }

--  The Common Table
heCommonTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF HeCommonEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A table containing information about headend (or indoor)
            equipment."
    ::= { heCommonParams 1 }

heCommonEntry OBJECT-TYPE
    SYNTAX      HeCommonEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "Information about particular headend equipment."
    INDEX       { entPhysicalIndex }
    ::= { heCommonTable 1 }

HeCommonEntry ::= SEQUENCE {
    heCommonTime                    DateAndTime,
    heCommonTemperature             HeTenthCentigrade,
    heCommonSoftwareReset           INTEGER,
    heCommonAlarmDetectionControl   INTEGER
}

heCommonTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-write
```

```
   STATUS      current
   DESCRIPTION
         "Real time clock."
   ::= { heCommonEntry 1 }

heCommonTemperature OBJECT-TYPE
   SYNTAX      HeTenthCentigrade
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "Temperature measured inside the headend equipment.

         This object must provide for the alarm management capabilities
         with a corresponding entry in the propertyTable of
         SCTE-HMS-PROPERTY-MIB (HMS026).

         An alarm shall be recorded as an entry in the currentAlarmTable
         of SCTE-HMS-PROPERTY-MIB (HMS026).

         A log record shall be added as an entry in the heCommonLogTable.

         An heCommonAlarmEvent notification shall be sent."
   ::= { heCommonEntry 2 }

heCommonSoftwareReset OBJECT-TYPE
   SYNTAX      INTEGER { reset(1) }
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "This object is used to reset software of the headend physical
         entity.

         A SET request with the value reset(1) only shall reset
         the software application. The reset implementation is
         vendor specific.

         A GET request shall always return the value reset(1) and
```



```
        shall have no effect on the entity."
    ::= { heCommonEntry 3 }

heCommonAlarmDetectionControl OBJECT-TYPE
    SYNTAX      INTEGER {
        detectionDisabled(1),
        detectionEnabled(2),
        detectionEnabledAndRegenerate(3)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object is used to control the detection of alarms in
        this headend entity.

        Each headend entity may provide for the alarm management
        capabilities. The provisions shall be done by means of
        the propertyTable and/or the discretePropertyTable of
        SCTE-HMS-PROPERTY-MIB (HMS026).

        When a threshold from either the propertyTable or the
        discretePropertyTable is crossed in a manner described by
        SCTE-HMS-PROPERTY-MIB (HMS026), then an alarm is said to have
        occurred. When the alarm is detected, then

        (1) an entry is placed in the heCommonLogTable, which serves
            as a log of the most recent alarm events;

        (2) an heCommonAlarmEvent trap is generated;

        (3) a property which is not in the nominal state will have
            an entry in the currentAlarmTable of SCTE-HMS-PROPERTY-MIB.

        The detectionDisabled(1) value prevents the threshold detection
        process associated with the property table and discrete property
        table from running. The headend entity shall not generate alarms.
        The contents of the heCommonLogTable, currentAlarmTable, each
```

```
            instance of discreteAlarmState, and each instance of
            currentAlarmState shall remain in the state prior to
            detectionDisabled(1) being applied.

            The detectionEnabled(2) value permits alarm detection to run.
            The detection process continues from the state the headend entity
            was in prior to detectionEnabled(2) being set.

            The detectionEnabledAndRegenerate(3) value clears all alarm
            information and permits alarm detection to run. All alarm
            properties, both discrete and analog, are restored to the
            nominal value before alarm detection runs. Any properties
            that are in an alarm state SHALL NOT produce a 'return to
            normal' alarm as part of the process. Setting this value
            clears the heCommonLogTable and the currentAlarmTable.

            The detectionEnabledAndRegenerate(3) value is transient, that is
            a SET request with a value detectionEnabledAndRegnerate(3) shall
            return the same value detectionEnabledAndRegnerate(3). Subsequent
            GET requests shall return a value detectionEnabled(2).

            The detectionDisabled(1) value shall affect the generation
            of heCommonAlarmEvent trap only. Traps added in
            the future are assumed to be unaffected by this object,
            unless stated in the description of that trap.

            This object has a default value of detectionEnabled(2).

            The value shall be maintained in non-volatile memory."
        ::= { heCommonEntry 4 }

-- The Common Log Group and Table
heCommonLogNumberOfEntries OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The current number of entries in the heCommonLogTable.
        Before the very first wrap-around condition occurs
        for heCommonLogIndex, heCommonLogNumberOfEntries will
        return the total number of entries logged in heCommonLogTable,
        since the unit was powered up.
        After the first wrap-around condition has occured for
        for the value of the MIB variable heCommonLogIndex,
        heCommonLogNumberOfEntries will return the maximum
        number of rows the heCommonLogTable can hold.
        "
    ::= { heCommonLog 1 }

heCommonLogLastIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Index of the most recent alarm entry logged in the
        heCommonLogTable. The value of this variable can be
        used as the value of heCommonLogIndex to retrieve the
        most recent logged entry."
    ::= { heCommonLog 2 }

heCommonLogTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF HeCommonLogEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of alarms that have been logged.
        Agent should generate the SNMP HMS notification every time a
        new alarm entry is logged.
        This table should support a minimum of 16 entries."
    ::= { heCommonLog 3 }

heCommonLogEntry OBJECT-TYPE
    SYNTAX      HeCommonLogEntry
    MAX-ACCESS  not-accessible
```

```
   STATUS      current
   DESCRIPTION
         "A set of data describing an alarm event that has
         been logged."
   INDEX       { heCommonLogIndex }
   ::= { heCommonLogTable 1 }

HeCommonLogEntry ::= SEQUENCE {
   heCommonLogIndex  Integer32,
   heCommonLogOID    OBJECT IDENTIFIER,
   heCommonLogValue  Integer32,
   heCommonLogState  INTEGER,
   heCommonLogTime   DateAndTime,
   heCommonLogText   DisplayString
}

heCommonLogIndex OBJECT-TYPE
   SYNTAX      Integer32 (1..65535)
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
         "An index that uniquely identifies an entry
         in the log table.  Indexes are assigned beginning with 1
         and increased by one with each new log entry up to 65535.
         The next entry after 65535 is one. The agent may choose to
         delete the oldest instances of heCommonLogEntry as required
         because of lack of memory.  It is an implementation-specific
         matter as to when this deletion may occur.
         Note - The wrap-around for the heCommonLogIndex variable
         MUST occur after 65535 regardless of the implementation
         specific size of the hlCommonLogTable."
   ::= { heCommonLogEntry 1 }

heCommonLogOID OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-only
   STATUS      current
```

```
   DESCRIPTION
         "This is the OID of the object that has changed alarm state."
   ::= { heCommonLogEntry 2 }

heCommonLogValue OBJECT-TYPE
   SYNTAX      Integer32 (-2147483648..2147483647)
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This is the value of the object at the time it changed alarm
         state."
   ::= { heCommonLogEntry 3 }

heCommonLogState OBJECT-TYPE
   SYNTAX      INTEGER {
      heCommonNominal(1),
      heCommonHIHI(2),
      heCommonHI(3),
      heCommonLO(4),
      heCommonLOLO(5),
      heCommonDiscreteMajor(6),
      heCommonDiscreteMinor(7)
   }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "The new alarm state of the object which caused the event
         to be recorded into the log."
   ::= { heCommonLogEntry 4 }

heCommonLogTime OBJECT-TYPE
   SYNTAX      DateAndTime
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This is the time when the alarm change for the object
         occured."
```

```
   ::= { heCommonLogEntry 5 }

heCommonLogText OBJECT-TYPE
   SYNTAX      DisplayString
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
         "This is a text field describing the alarm. This
         field could be a zero length string in certain
         agent implementations."
   ::= { heCommonLogEntry 6 }

-- Headend Common MIB Trap Definitions
heCommonTraps      OBJECT IDENTIFIER ::= { heCommonMib 2 }
heCommonTrapPrefix OBJECT IDENTIFIER ::= { scteHmsTree 0 }

heCommonAlarmEvent NOTIFICATION-TYPE
   OBJECTS {
      heCommonLogOID,
      heCommonLogValue,
      heCommonLogState,
      heCommonLogTime
   }
   STATUS  current
   DESCRIPTION
         "The SNMP trap that is generated when an alarm event is found.
         At the option of the unit, the heCommonLogText variable may be
         reported as a fifth varbind, for those instances where an
         additional text field is supported."
   ::= { heCommonTrapPrefix 5 }

-- Conformance information
heCommonConformance OBJECT IDENTIFIER ::= { heCommonMib 3 }

heCommonCompliances OBJECT IDENTIFIER ::= { heCommonConformance 1 }
heCommonGroups      OBJECT IDENTIFIER ::= { heCommonConformance 2 }
```

```
-- Compliance statements
heCommonCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
         "The compliance statement for SNMP entities which implement
         this MIB."
   MODULE  -- this module
      MANDATORY-GROUPS { heCommonLogGroup,
                         heCommonNotificationsGroup
                       }

   MODULE  ENTITY-MIB
      MANDATORY-GROUPS { entityPhysicalGroup,
                         entityPhysical2Group,
                         entityGeneralGroup,
                         entityNotificationsGroup
                       }

   MODULE  SNMP-TARGET-MIB
      MANDATORY-GROUPS { snmpTargetBasicGroup }

   MODULE  SNMP-NOTIFICATION-MIB
      MANDATORY-GROUPS { snmpNotifyGroup }

   MODULE  SNMPv2-MIB
      MANDATORY-GROUPS { systemGroup }

-- The OBJECT clauses below indicate the optional objects of
-- the systemGroup. They also imply that other objects of
-- the group must be implemented:
--   sysDescr,
--   sysObjectID,
--   sysUpTime,
--   sysContact,
--   sysName,
--   sysLocation,
--   sysServices.
```

```
        OBJECT      sysORDescr
        MIN-ACCESS  not-accessible
        DESCRIPTION
              "Implementation of this object is optional."

        OBJECT      sysORID
        MIN-ACCESS  not-accessible
        DESCRIPTION
              "Implementation of this object is optional."

        OBJECT      sysORLastChange
        MIN-ACCESS  not-accessible
        DESCRIPTION
              "Implementation of this object is optional."

        OBJECT      sysORUpTime
        MIN-ACCESS  not-accessible
        DESCRIPTION
              "Implementation of this object is optional."

   MODULE  SCTE-HMS-PROPERTY-MIB
      MANDATORY-GROUPS { analogAlarmsGroup,
                         discreteAlarmsGroup,
                         currentAlarmsGroup
                       }

   ::= { heCommonCompliances 1 }

-- MIB groupings
heCommonParamsGroup OBJECT-GROUP
   OBJECTS {
           heCommonTime,
           heCommonTemperature,
           heCommonSoftwareReset,
           heCommonAlarmDetectionControl
         }
```

```
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to represent the
            common parameters of the headend managed entities."
    ::= { heCommonGroups 1 }

heCommonLogGroup OBJECT-GROUP
    OBJECTS {
             heCommonLogNumberOfEntries,
             heCommonLogLastIndex,
             heCommonLogOID,
             heCommonLogValue,
             heCommonLogState,
             heCommonLogTime,
             heCommonLogText
            }
    STATUS  current
    DESCRIPTION
            "The collection of objects which are used to record
            an alarm event into the headend agent log."
    ::= { heCommonGroups 2 }

heCommonNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { heCommonAlarmEvent }
    STATUS  current
    DESCRIPTION
            "The collection of notifications used by the headend agent
            to report the exceptional conditions to the management
            application."
    ::= { heCommonGroups 3 }

END
```