SCTE · ISBE
STANDARDS

Network Operations Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 83-3 2017

Hybrid Fiber/Coax Inside Plant Status Monitoring
SCTE-HMS-HMTS-MIB
Management Information Base (MIB) Definition

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.

140 Philips Road
Exton, PA 19341

CONTENTS

SCOPE

This document is identical to SCTE 83-3 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.

This document provides the MIB definitions for management of an HMTS system and defines how to address the HMS transponders connected to the HTMS system.

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 2573 SNMP-NOTIFICATION-MIB.
IETF RFC 2573 SNMP-TARGET-MIB.
IETF RFC 2578 SNMPv2-SMI.
IETF RFC 2579 SNMPv2-TC.
IETF RFC 2580 SNMPv2-CONF.
IETF RFC 2737 ENTITY-MIB.
IETF RFC 3418 SNMPv2-MIB.
SCTE 25-2 HMS MAC Specification.
SCTE 36 SCTE-ROOTS.
SCTE 37 SCTE-HMS-ROOTS.
SCTE 38-1 SCTE-HMS-PROPERTY-MIB.
SCTE 38-11 SCTE-HMS-HEADENDIDENT-MIB.
SCTE 84-1 SCTE-HMS-HE-COMMON-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.

HMTS: Hybrid Management Termination System.

HMS: Hybrid Management System.

EMS: Element Management System.

XP / Xpndr: Abbreviation for transponder.

IP: Internet Protocol.

SNMP: Simple Network Management Protocol.

MAC: Media Access Control. Basically a data packet definition that provides for end to end transmission of data between two network elements.

Broadcast Address: A MAC specific address value that all devices will recognize and process requests to, but not reply to.

Multicast Address: A MAC specific address value that can be configured into multiple devices such that a group is created. Use of a multicast address will cause all members of the group to act upon the request, but none of the members will reply. This differs from the broadcast address only in that all devices are pre-configured with the broadcast address.

Unicast Address: A MAC specific unique address that is used to identify what device is to process a message and return a reply.

Forward Path / Return Path: bi-directional communication requires a forward (from HMTS to HMS device) and return (from HMS device to HMTS) path (communication connection) regardless of the type of port in use.

REQUIREMENTS

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

The syntax is given below.

```
SCTE-HMS-HMTS-MIB DEFINITIONS ::= BEGIN

-- **********************************************************************
-- *
-- * Module Name: SCTE 83-3 (formerly HMS120)
-- *
-- * SCTE Status:
-- *
-- * See also: HMS134, The HMTS Theory of Operation document. HMS134 has
-- *           the status of recommended practice document.
-- *
-- **********************************************************************
-- *
-- **********************************************************************


IMPORTS
    OBJECT-TYPE,
    MODULE-IDENTITY,
    NOTIFICATION-TYPE,
    IpAddress,
    Unsigned32,
    Integer32
      FROM SNMPv2-SMI

    OBJECT-GROUP,
    MODULE-COMPLIANCE
      FROM SNMPv2-CONF

    TEXTUAL-CONVENTION,
    RowStatus,
    MacAddress,
    DisplayString
      FROM SNMPv2-TC

    systemGroup
      FROM SNMPv2-MIB

    entityNotificationsGroup,
    entityGeneralGroup,
    entityPhysical2Group,
    entityPhysicalGroup
      FROM ENTITY-MIB

    snmpTargetBasicGroup
      FROM SNMP-TARGET-MIB

    currentAlarmsGroup,
    discreteAlarmsGroup,
    analogAlarmsGroup
      FROM SCTE-HMS-PROPERTY-MIB     -- SCTE 38-1  (Formerly HMS026)

    heHMTS
      FROM SCTE-HMS-HEADENDIDENT-MIB -- SCTE 38-11 (Formerly HMS114)

    heCommonNotificationsGroup,
    heCommonLogGroup
       FROM SCTE-HMS-HE-COMMON-MIB;   -- SCTE 84-1  (Formerly HMS111)

heHMTSMIB MODULE-IDENTITY
    LAST-UPDATED "200405040000Z" -- May 4, 2004
    ORGANIZATION
       "SCTE HMS HMTS Working Group"
    CONTACT-INFO
       "Hung Nguyen,
        SCTE HMS Subcommittee, Chairman
        AOL Time Warner Cable
        http://www.scte.org/standards/
        mailto:standards@scte.org

        Bart Van Assche
        HMTS Working Group, Editor
        Scientific-Atlanta
        Luipaardstraat 12
        B-8500 Kortrijk
        Belgium
        mailto:Bart.VanAssche@sciatl.com
       "
   DESCRIPTION
       "This MIB contains information that must be supported by all Hybrid
        Management Termination Systems (HMTS).  An HMTS is defined as a
        gateway between a IP (Internet Protocol) network and the HMS Physical
        and MAC layers defined by SCTE 25-1 (Formerly HMS005) and SCTE 25-2
        (Formerly HMS004).

        The HMTS shall support, but is not limited to the following MIBs:
          RFC-1213 - all current groups and objects
          SNMPv2-MIB - all current groups and objects
          SCTE 38-1 (formerly HMS026) - Properties MIB
          SCTE 84-1 (formerly HMS111) - HMS HE Common
       "
::= { heHMTS 2 }

-- *
-- *   Textual Conventions in the HMTS MIB
-- *

HmtsComStatCodes ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
       "This enumerates the communication status condition codes.
        These codes indicate the state of the communication with a
        network elements transponder (device).  The conditions attempt
        to indicate where the problem with communication to the device
        is to be found.
       "
    SYNTAX INTEGER {
       noError(1),       -- No communication fault detected.
       noRevPortId(2),   -- Device entry's reverse port ID is not assigned.
       notActive(3),     -- Device entry is not active.
       notRegis(4),      -- Device is not registered.
       pendRegis(5),     -- Device registration is pending.
```

```
        registering(6),   -- Device is registering.
        transDisabled(7), -- Device's HMTS transmitter is disabled.
        rcvrDisabled(8),  -- Device's HMTS receiver is disabled.
        rtrnLvl(9),       -- Device's return level is bad.
        notResp(10),      -- Device is not responding.
        invMac(11),       -- Device has an invalid MAC address.
        fwdFreqMismatch(12), -- Forward frequencies of Transmitter/CHNLDESC PDU
                          --   do not match.
        -- The following errors (20-29) only apply to HMTS' that support
        -- IP Based Proxies.
        invIP(20),        -- Device has an invalid IP address (IP is the
                          --   Default).
        dupIP(21),        -- Device has a duplicate IP Address.
        -- The following errors (30-39) only apply to HMTS' that support
        -- Community Based Proxies.
        invComm(30),      -- Device has an invalid community string.
        dupComm(31),      -- Device has a duplicate community string.
        other(32)         -- Any status other than the above.
     }


-- *
-- *   Groups in the HMTS MIB
-- *

heHMTSObjects        OBJECT IDENTIFIER ::= { heHMTSMIB 1 }
hmtsNotifications    OBJECT IDENTIFIER ::= { heHMTSObjects 0 }
hmtsInfoGroup        OBJECT IDENTIFIER ::= { heHMTSObjects 1 }
hmtsManagementGroup  OBJECT IDENTIFIER ::= { heHMTSObjects 2 }
hmtsDeviceGroup      OBJECT IDENTIFIER ::= { heHMTSObjects 3 }
hmtsIPGroup          OBJECT IDENTIFIER ::= { heHMTSObjects 4 }
hmtsCommGroup        OBJECT IDENTIFIER ::= { heHMTSObjects 5 }


-- **********************************************************
-- *
-- *  HMTS Information Group
-- *
-- *  Description:
-- *    This group provides a set of information common to each HMTS.
-- *    This group is current for all termination systems.
-- *
-- **********************************************************

hmtsAdminState  OBJECT-TYPE
     SYNTAX INTEGER {
        active (1),
        inactive (2)
     }
     MAX-ACCESS read-write
     STATUS     current
     DESCRIPTION
        "This controls whether the HMTS monitoring service is active
         active   - The HMTS is actively looking for traps and
                    non-responding devices on the serial links.  Fully Enabled
                    functionality of the HMTS.
         inactive - The HMTS does not poll devices to search for
                    non-respondering devices or alarms (Synchronous or
                    Asynchronous), but HMTS continues all other
                    activities.  Contention mode is set to OFF while
                    inactive.  This disables registration as well as
                    polling.  No messages from a higher level manager
                    will be forwarded to the devices.  In essence all
                    directed communication to or from the device is
                    disabled.
         In both states SNMP requests to configured serial equipment is
         possible.  Sending CHNLDESC and TIME PDUs continues at their
         configured intervals.
         This object is non-volatile.
        "
     ::= { hmtsInfoGroup 1 }

hmtsOperState OBJECT-TYPE
     SYNTAX INTEGER {
        operational (1),
        swFailure (2),
        hwFailure (3)
     }
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This reports the overall status of the termination system.
         Vendors must use caution and prevent repeating alarms when
         trying to automatically recover a failure.
         operational - System is fully operational, no detected failures.
         hwFailure   - System has detected a hardware failure.
         swFailure   - System has detected a software failure, and can
                       not be recovered.

         This object has a threshold property in the discretePropertyTable as
         defined in SCTE 38-1 (formerly HMS026). Note: it will depend on the
         type or software or hardware failure whether or not it is possible to
         send out an SNMP trap for a specific type of failure. Furthermore,
         whether or not a trap will be sent is vendor specific.
        "
     ::= { hmtsInfoGroup 2 }

hmtsProxyType OBJECT-TYPE
     SYNTAX INTEGER {
        ipBased (1),
        communityBased (2),
        both (3)
     }
     MAX-ACCESS read-write
     STATUS     current
     DESCRIPTION
        "This value controls the type of proxy supported by the HMTS,
         Community, IP, or MIB based. If the termination system does not
         support setting a specific value the termination system shall
         return a bad-value response.
         The HMTS will either be:
         ipBased        - The HMTS uses an IP address to designate the
```

```
                     network element where the SNMP request is
                     destined.
       communityBased - The HMTS uses the SNMP community string to
                     designate the network element there the SNMP
                     request is destined.
       both           - The HMTS can use either proxy schemas to
                     designate the network element where the SNMP
                     request is destined.
       This object is non-volatile.
      "
    ::= { hmtsInfoGroup 3 }


hmtsFreqSwitchMethod  OBJECT-TYPE
    SYNTAX INTEGER {
      automatic (1),
      manual (2)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This controls whether the HMTS automatically changes the Forward
       ports frequency (hmtsFwdHmtsFrequency) when hmtsFwdXpndrFrequency
       is set.
       automatic - The HMTS automatically sets its transmitter to
                 the newly commanded transponder frequency.  The
                 algorithm executed for this change is vender
                 specific.
       manual    - The HMTS requires the that its forward port
                 transmitter frequency be explicitly set by a
                 higher level manager.
       The HMTS shall respond with a bad value error if an attempt
       is made to set this object to an unsupported value.
       This object is non-volatile.
      "
    DEFVAL     { manual }
    ::= { hmtsInfoGroup 4 }

hmtsModel OBJECT-TYPE
    SYNTAX     DisplayString (SIZE (0..255))
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "Vendor-specific HMTS model identification string, in human-readable format.
      "
    ::= { hmtsInfoGroup 5 }

hmtsSerialNumber OBJECT-TYPE
    SYNTAX     DisplayString (SIZE (0..255))
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "Serial number of the device, in human-readable format.
      "
    ::= { hmtsInfoGroup 6 }
```

```
hmtsSoftwareVersion OBJECT-TYPE
    SYNTAX     DisplayString (SIZE (0..255))
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "Version of the running HMTS software.
      "
    ::= { hmtsInfoGroup 7 }

hmtsTimeServerAddress OBJECT-TYPE
    SYNTAX     IpAddress
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "If not set to 0.0.0.0, IP address of the time server the HMTS should
       synchronize its internal clock with. If supported, the vendor must
       specify the protocol used. Examples:
       - daytime protocol (RFC 867), local time, string format;
         DAY, MONTH DD, YYYY HH:MM:SS.
       - NTP, UTC (RFC 958, NTP's time format is specified in the RFC).

       Note: you can query the HMTS' clock via heCommonTime.
       This object is non-volatile.
      "
    DEFVAL { '0000'h } -- '0000'h is IP address 0.0.0.0
    ::= { hmtsInfoGroup 8 }

hmtsTimeServerSyncInterval OBJECT-TYPE
    SYNTAX     Integer32 (3600..8640000)
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "Specifies the time between successive clock synchronization attempts.
       This object is non-volatile.
      "
    DEFVAL { 86400 }
    ::= { hmtsInfoGroup 9 }


-- **********************************************************************
-- *
-- *  HMTS MAC Protocol Information Group
-- *
-- *  Description:
-- *    This group provides controls of the MAC payload commands.
-- *    This group is current for all termination systems.
-- *
-- **********************************************************************

hmtsMacPduGroup           OBJECT IDENTIFIER
   ::= { hmtsManagementGroup 1 }

hmtsRegistrationGroup     OBJECT IDENTIFIER
   ::= { hmtsManagementGroup 2 }
```

```
hmtsSnmpTrapControlGroup  OBJECT IDENTIFIER
   ::= { hmtsManagementGroup 3 }

hmtsSnmpProtocolGroup OBJECT IDENTIFIER
   ::= { hmtsManagementGroup 4 }

-- ***********************************************************************
-- *
-- *  HMTS Mac PDU Group
-- *
-- *  Description:
-- *    This group defines the controls for the generic behavior of the
-- *    MAC PDUs within the HMTS.  This also includes the control of
-- *    broadcasting the TIME and CHNLDESC PDUs.
-- *    This group is mandatory for all termination systems.
-- *
-- ***********************************************************************

hmtsMacPduTimeout OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the timeout of all MAC PDUs except the TALK PDU.
        This object is non-volatile.
       "
    DEFVAL     { 15 }
    ::= { hmtsMacPduGroup 1 }

hmtsTalkPduTimeout OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the timeout of the TALK PDU.
        This object is non-volatile.
       "
    DEFVAL { 5000 }
    ::= { hmtsMacPduGroup 2 }

hmtsMacBroadcastDelay OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the delay after sending a broadcast MAC PDU.
        This object is non-volatile.
       "
    DEFVAL { 250 }
    ::= { hmtsMacPduGroup 3 }

hmtsAlarmDiscoveryMode OBJECT-TYPE
    SYNTAX INTEGER {
       polling (1),
       contention (2),
       hybrid (3),
       off (4)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the mode of operation for discovering alarms:
        Polling    - HMTS uses polling to determine if the device has
                   an alarm.
        Contention - HMTS uses broadcast contention mode to be notified
                   asynchronously about an alarm.
        Hybrid     - HMTS uses a vendor specific means for combining polling
                   and contention modes to discover alarms.
        Off        - HMTS is not looking for any alarms.

        If the termination system does not support setting a specific
        value the termination system shall return a bad-value response.
        This object is non-volatile.
       "
    ::= { hmtsMacPduGroup 4 }

hmtsChnldescPduInt OBJECT-TYPE
    SYNTAX     Integer32 (1..30)
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the delay between the broadcasting of the
        CHNLDESC PDUs.  This should continue even if the hmtsAdminState is
        inactive.  If either the hmtsFwdXpndrFrequency or the
        hmtsRevFrequency are not valid then this PDU shall not be sent.
        This object is non-volatile.
       "
    DEFVAL     { 30 }
    ::= { hmtsMacPduGroup 5 }

hmtsTimePduInt OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the delay between the broadcasting of the
        TIME PDU.  The time within the HMTS must first be synchronized
        before the broadcasting of this PDU.
        This object is non-volatile.
       "
    DEFVAL     { 3600 }
    ::= { hmtsMacPduGroup 6 }

hmtsDeviceAccessMode OBJECT-TYPE
    SYNTAX INTEGER {
       immediate (1),
       queued (2),
       interrupt (3)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This object reports and controls how the HMTS will respond to
        requests to devices during a contention period.
        immediate - means the HMTS will forward the request to the device
                  even during an ongoing contention period.  In this
                  mode it is possible for the devices response to have a
                  collision with an asynchronous response from another
                  device.
        queued    - means that the HMTS will queue this message.  When the
                  contention period has completed the HMTS will forward
                  the queued requests.
        interrupt - means that the HMTS will abort the contention period
                  and forward the request, and the resume of normal
                  operations.

        The HMTS must support at least one of these modes of operation.  When
        an attempt is made to set an unsupported value the HMTS shall return a
        bad value error.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsMacPduGroup 7 }



-- **********************************************************************
-- *
-- *  HMTS Registration Control Group
-- *
-- *  Description:
-- *    This group defines the controls for asynchronous registration
-- *    requests.  This group controls the broadcasting of the
-- *    CONTMODE REG PDU. This is part of the MAC Protocol Group.
-- *    This group is mandatory for all termination systems.
-- *
-- **********************************************************************

hmtsRegInterval OBJECT-TYPE
    SYNTAX     Integer32 (1..86400)
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the delay between the beginnings of each
        registration period.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsRegistrationGroup 1 }

hmtsRegMinDuration OBJECT-TYPE
    SYNTAX     Integer32 (1..255)
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the minimum time for the registration
        duration period.   This is the minimum amount of time the
        registration window will be open.  Attempt to set minimum
        duration greater than the maximum duration will result in
        a bad value error being returned.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsRegistrationGroup 2 }

hmtsRegMaxDuration OBJECT-TYPE
    SYNTAX     Integer32 (1..255)
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the maximum time for the registration duration
        period.  This is provided to allow a dynamic duration when a
        large number of TALKRQST PDUs are received during registration
        period.  How the HMTS increases the registration duration
        period to this value is vendor specific.  Attempt to set the
        maximum duration less than the minimum duration will result in
        a bad value error being returned.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsRegistrationGroup 3 }

hmtsRegContinuity OBJECT-TYPE
    SYNTAX INTEGER {
       immediate (1),
       queued (2),
       continuous (3)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This indicates the manner that the HMTS uses to handle
        processing of the TALKRQST PDU for auto discovery.
        immediate  - means that the HMTS immediately terminates the
                     contention period and request the discover PDU.
        queued     - means that the HMTS will request the discover PDU
                     after the contention period terminates.
        continuous - means that the HMTS will request the discovery PDU
                     despite the contention period being open.
        If the HMTS does not support setting a specific value the HMTS
        shall return a bad-value response.  The HMTS shall support at
        least one of the above values.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsRegistrationGroup 4 }
```

```
-- ********************************************************************
-- *
-- *  HMTS SNMP Trap Control Group
-- *
-- *  Description:
-- *    This group shall be supported if the termination system supports
-- *    the broadcasting of the CONTMODE ON/OFF/RES/INH PDUs.  For the
-- *    asynchronous collection of SNMP Traps
-- *    This group is optional.
-- *
-- ********************************************************************

hmtsTrapControlTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsTrapControlEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "This table contains entries to control multicast groups for
        alarm collection.  This table allows for multiple groups of
        alarms to be collected.  It is expected that a higher level
        manager will add entries to this table and the transponders
        multicast address table.  When trap contention modes are used
        polling will be disable while an active entry in this table has
        contention enabled.  This way the polling does not interfere with
        the possible asynchronous notifications.  As default this table
        shall have one entry, containing the broadcast address.  This is
        a non-volatile table.
       "
    ::= { hmtsSnmpTrapControlGroup 1 }

hmtsTrapControlEntry OBJECT-TYPE
    SYNTAX HmtsTrapControlEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "A list of information about the Trap Control Entry.
       "
    INDEX      { hmtsTControlId }
    ::= { hmtsTrapControlTable 1 }

HmtsTrapControlEntry ::= SEQUENCE {
    hmtsTControlId
        Integer32,
    hmtsTControlInterval
        Integer32,
    hmtsTControlChainId
        Integer32,
    hmtsTControlMinDuration
        Integer32,
    hmtsTControlMaxDuration
        Integer32,
    hmtsTControlContinuity
        INTEGER,
    hmtsTControlMulticastAddr
        MacAddress,
    hmtsTControlRowStatus
        RowStatus
}


hmtsTControlId OBJECT-TYPE
    SYNTAX     Integer32 ( 1..2147483647 )
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This value uniquely identifies this entry.

        DEFAULT: vendor specific.
        This object is non-volatile.
      "
    ::= { hmtsTrapControlEntry 1 }

hmtsTControlInterval OBJECT-TYPE
    SYNTAX     Integer32 ( 0..86400 )
    UNITS      "1 s"
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the time between the beginnings of two
        consecutive alarm discovery periods.  If the
        record is chained this is the interval between
        start of this entry and the start of the next
        entry with the same chain Id.
        DEFAULT: vendor specific.
        This object is non-volatile.
      "
    ::= { hmtsTrapControlEntry 2 }

hmtsTControlChainId OBJECT-TYPE
    SYNTAX     Integer32 ( 0..86400 )
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This identifies which Trap control
        records are to be run consecutively.  The value of zero
        indices that the entry is independent from all other
        entries in this table.  Likewise and entry with the
        chain id of 1 is independent of the chain id of 2.
        DEFAULT: vendor specific.
        This object is non-volatile.
      "
    ::= { hmtsTrapControlEntry 3 }


hmtsTControlMinDuration OBJECT-TYPE
    SYNTAX     Integer32 ( 1..86400 )
    UNITS      "1 s"
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the minimum time for the alarm/trap discovery
```

```
        duration period for this entry.
        Attempt to set minimum duration greater than the maximum
        duration will result in a bad value error being returned.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsTrapControlEntry 4 }

hmtsTControlMaxDuration OBJECT-TYPE
    SYNTAX     Integer32 ( 1..86400 )
    UNITS      "1 s"
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This is the maximum time for the alarm//trap discovery duration
        period for this entry.  This is provided to allow a varying
        window size when a large number of TALKRQST PDUs are received
        during the alarm discovery period.  How the HMTS increases
        the alarm discovery duration period to this value is vendor
        specific.  Attempt to set the maximum duration less than the
        minimum duration will result in a bad value error being returned.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsTrapControlEntry 5 }

hmtsTControlContinuity OBJECT-TYPE
    SYNTAX INTEGER {
        immediate (1),
        queued (2),
        continuous (3)
    }
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This indicates the manner that the HMTS uses to handle
        processing of the TALKRQST PDU for alarm discovery.
        immediate  - means that the HMTS immediately terminates the
                     contention period and request the SNMP traps.
        queued     - means that the HMTS will request the SNMP traps
                     after the contention period terminates.
        continuous - means that the HMTS will request the SNMP traps
                     despite the contention period being open.
        If the HMTS does not support setting a specific value the HMTS
        shall return a bad-value response.  The HMTS shall support at
        least one of the above values.
        DEFAULT: vendor specific.
        This object is non-volatile.
       "
    ::= { hmtsTrapControlEntry 6 }

hmtsTControlMulticastAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
```

```
        "This multicast address is used by the HMTS to address transponders
         that are to enable their contention when the row containing this
         object is processed. This address must already have been defined in
         hmtsMulticastAddrTable. If this address is not ff-ff-ff-ff-ff-ff,
         the multicast address should already be present in the multicast
         address table (commonMulticastAddressTable) of the appropriate
         transponders. There are two ways in which a multicast address can be
         programmed in that table: either directly by the EMS, or indirectly
         by filling the multicast address in in hmtsRevMulticastAddr such
         that the HMTS programs it into commonMulticastAddressTable.
         This object is non-volatile.
        "
    ::= { hmtsTrapControlEntry 7 }

hmtsTControlRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This is used to add and delete rows from this table dynamically.
        "
    ::= { hmtsTrapControlEntry 8 }


-- ********************************************************************
-- *
-- *  HMTS SNMP Protocol Group
-- *
-- *  Description:
-- *    This group provides control over the SNMP payloads on the HMS MAC
-- *    layer.
-- *    This group is mandatory for all termination systems.
-- *
-- ********************************************************************

hmtsSnmpTimeout OBJECT-TYPE
    SYNTAX     Integer32 (1..10000)
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This is how long the HMTS will wait before timing out a SNMP
         request. The HMTS shall not retry SNMP requests. This is the
         role of the external management system. Although it is required
         in the MAC layer document that each transponder responds in
         less than 5000ms, most transponders can respond faster. Please
         consult the documentation of the installed transponders.
         Warning: if the SNMP timeout is set too low, all SNMP
         communication with transponders will fail.
         This object is non-volatile.
        "
    DEFVAL     { 5000 }
    ::= { hmtsSnmpProtocolGroup 1 }

hmtsSnmpBroadcastDelay OBJECT-TYPE
    SYNTAX     Integer32 (1..10000)
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This is the amount of time that the HMTS will wait after
         multicasting or broadcasting an SNMP request. This applies both
         to mulicast/broadcast SNMP requests generated by the HMTS itself
         and to mulicast/broadcast SNMP requests generated by an external
         SNMP manager. Note: the rate at which multicast/broadcast SNMP
         requests are sent to an HMTS must be lower than the rate at which
         the HMTS sends these requests to the HFC network. Otherwise some
         of the multicast/broadcast SNMP requests may be lost in the HMTS.
         This object is non-volatile.
        "
    DEFVAL     { 5000 }
    ::= { hmtsSnmpProtocolGroup 2 }

-- **********************************************************************
-- *
-- *  HMTS Forward Path Port Table
-- *
-- **********************************************************************

hmtsFwdPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsFwdPortEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The forward port table contains information about the forward
         transmitter ports of the HMTS.  The ports may be of several
         varieties.  The majority of the control fields apply to the RF
         serial port types, which support the HMS physical and MAC
         layers.
        "
    ::= { hmtsManagementGroup 5 }

hmtsFwdPortEntry OBJECT-TYPE
    SYNTAX HmtsFwdPortEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "A list of information about each forward path port
        "
    INDEX { hmtsFwdPortId }
    ::= { hmtsFwdPortTable 1 }

HmtsFwdPortEntry ::= SEQUENCE {
    hmtsFwdPortId
        DisplayString,
    hmtsFwdPortDescr
        DisplayString,
    hmtsFwdPortType
        INTEGER,
    hmtsFwdPortAdminState
        INTEGER,
    hmtsFwdPortOperState
```

```
      INTEGER,
   hmtsFwdHmtsFrequency
      Integer32,
   hmtsFwdXpndrFrequency
      Integer32,
   hmtsFwdProvPwrLvl
      Integer32,
   hmtsFwdMaxPwrLvl
      Integer32,
   hmtsFwdPollTime
      Integer32
}

hmtsFwdPortId OBJECT-TYPE
   SYNTAX DisplayString (SIZE(1..64))
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
      "This is a unique string identifying the forward port.  This
       value is vendor specific and should physically identify the
       port.
      "
   ::= { hmtsFwdPortEntry 1 }

hmtsFwdPortDescr OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "This is a string of text describing the function of the port
       and physical location.  Such as 'HMS RF transmitter - Slot 1'.
       This value is vendor specific.
      "
   ::= { hmtsFwdPortEntry 2 }

hmtsFwdPortType OBJECT-TYPE
   SYNTAX INTEGER {
      rf    (1),
      rs485 (2),
      rs232 (3),
      other (4)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "Serial communication type:
       rf    - This HMTS port uses an RF modem for forward communication.
       rs485 - This HMTS port uses an RS-485 for forward communication.
       rs232 - This HMTS port uses an RS-232 for forward communication.
       other - This HMTS port uses a modem for forward communication that
       is neither a RF, RS-485 or RS-232 modem.
      "
   ::= { hmtsFwdPortEntry 3 }

hmtsFwdPortAdminState OBJECT-TYPE
   SYNTAX INTEGER {
      enable (1),
      disableCarrierOn (2),
      disableCarrierOff (3)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This allows the operator to disable/enable the processing of
       data sent on this port.  When disabled no forward path
       messages shall be sent out this port.  If the termination
       system does not support setting a specific value the
       termination system shall return a bad-value response.
       For RSxxx communication disableCarrierOn shall be used to
       disable the port.
       This object is non-volatile.
      "
    ::= { hmtsFwdPortEntry 4 }

hmtsFwdPortOperState OBJECT-TYPE
    SYNTAX INTEGER {
      noError (1),
      noFreqAssgn (2),
      freqUnlocked (3),
      portComFailure (4),
      otherError (5)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "This reports the operational status of the port.
       This object has a threshold property in the discretePropertyTable as
       defined in SCTE 38-1 (formerly HMS026)
      "
    ::= { hmtsFwdPortEntry 5 }

hmtsFwdHmtsFrequency OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 Hz"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This is the frequency at which an HMTS transmits its FSK signal
       on the HFC network, and only applies to RF ports. Other port
       types should return noSuchName upon SNMP get or set commands.

       When changing from one downstream frequency to another, an HMTS
       should first transmit several CHNLDESC packets with the new
       frequency as payload and modulated on the old frequency. This
       will tell transponders to switch from the old to new frequency.
       Next the HMTS must change its modulator from the old to the
       new frequency.

       In combination with the hmtsFwdXpndrFrequency variable, two
       applications are possible:
       - Tuning all transponders to a new frequency by first changing
         hmtsFwdXpndrFrequency and next setting hmtsFwdHmtsFrequency
         to the same frequency.
       - Recovering transponders that missed a frequency transition,
         e.g. because they were disconnected at the time the CHNLDESC
         PDU's were transmitted. Recovery can be performed by setting
         hmtsFwdHmtsFrequency for a short time to the frequency at
         which the non-communicating transponders are listening.

       Frequency transitions can be implemented in an HMTS e.g. in
       one of the following ways:
       1. Manual frequency transition: hmtsFwdHmtsFrequency and
          hmtsFwdXpndrFrequency can be set independently by an
          operator. Hence, both frequency transitions and recovering
          transponders is supported.
       2. Automatic frequency transition: when hmtsFwdXpndrFrequency
          is set, and after the transition CHNLDESC PDU's have been
          sent, the HMTS also sets hmtsFwdHmtsFrequency to the new
          frequency. SNMP sets on hmtsFwdHmtsFrequency will result in
          a bad value error.

       RESOLUTION: <=100 Hz
       This object is non-volatile.
      "
    ::= { hmtsFwdPortEntry 6 }

hmtsFwdXpndrFrequency OBJECT-TYPE
    SYNTAX     Integer32
    UNITS "1 Hz"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This is the frequency being sent in the payload of the CHNLDESC
       PDU. This variable only applies to RF ports, other port types
       should return noSuchName. When this variable is changed, several
       successive CHNLDESC PDU's should be broadcast by the HMTS.

       For more information, see also the description of
       hmtsFwdHmtsFrequency.

       RESOLUTION: <=100 Hz
       This object is non-volatile.
      "
    ::= { hmtsFwdPortEntry 7 }

hmtsFwdProvPwrLvl OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "0.1 dBmV"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This is the power level of the forward port transmitter modem.
       This value only applies to the RF port type; all other port
       types should return noSuchName.  Any attempt to write to this
       value for a non RF port Type will result in a Bad Value Error.
       Any attempt to set this level larger than hmtsFwdMaxPwrLvl shall
       also result in a bad value error.
       The value used internally will be rounded to the nearest supported
```

```
        value. The object reports the requested value, not the rounded value.
        RESOLUTION: vendor specific
        This object is non-volatile.
       "

    ::= { hmtsFwdPortEntry 8 }

hmtsFwdMaxPwrLvl OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "0.1 dBmV"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the maximum power level of the forward port transmitter modem.
        This value only applies to the RF port type; all other port types
        should return noSuchName.  Any attempt to write to this value for
        a non RF port Type will result in a Bad Value Error.
        The value used internally will be rounded to the nearest supported
        value. The object reports the requested value, not the rounded value.
        RESOLUTION: vender specific
        This object is non-volatile.
       "
    ::= { hmtsFwdPortEntry 9 }

hmtsFwdPollTime OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 s"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "The HMTS is required to poll the devices on the network.  This is
        the maximum amount of time the HMTS will allow to pass before
        automatically attempting communication to see if the device is
        still responding.  This is intended for use when the main alarm
        discovery mode is by contention.
        This object is non-volatile.
       "
    DEFVAL     { 360 }
    ::= { hmtsFwdPortEntry 10 }

-- *****************************************************************************
-- *
-- *  HMTS Reverse Path Port Table
-- *
-- *****************************************************************************

hmtsRevPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsRevPortEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "The Reverse Port table contains information about the receiver
        ports of the HMTS.  The reverse path may be Serial RF,
        RS-485 or another serial interface.  The serial RF port is
        usually a reverse path RF modem.  A unique identifier,
        hmtsRevPortId, indexes this table.
        "
    ::= { hmtsManagementGroup 6 }

hmtsRevPortEntry OBJECT-TYPE
    SYNTAX HmtsRevPortEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "A list of information about each reverse path port.
        "
    INDEX { hmtsRevPortId }
    ::= { hmtsRevPortTable 1 }

HmtsRevPortEntry ::= SEQUENCE {
    hmtsRevPortId
        DisplayString,
    hmtsRevFwdPortId
        DisplayString,
    hmtsRevPortDescr
        DisplayString,
    hmtsRevPortType
        INTEGER,
    hmtsRevPortAdminState
        INTEGER,
    hmtsRevPortOperState
        INTEGER,
    hmtsRevFrequency
        Integer32,
    hmtsRevMuteLvl
        Integer32,
    hmtsRevMulticastAddr
        MacAddress,
    hmtsRevReturnLvl
        Integer32,
    hmtsRevCRCErrors
        Integer32,
    hmtsRevFrameErrors
        Integer32,
    hmtsRevBackOffPeriod
        Integer32,
    hmtsRevACKTimeout
        Integer32,
    hmtsRevMaxMACRetries
        Integer32,
    hmtsRevBackOffMinExp
        Integer32,
    hmtsRevBackOffMaxExp
        Integer32
}

hmtsRevPortId OBJECT-TYPE
    SYNTAX DisplayString (SIZE(1..64))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "This is a unique string identifying the reverse path port.
```

```
         This value is vendor specific and should physically identify
         the port.
        "
    ::= { hmtsRevPortEntry 1 }

hmtsRevFwdPortId OBJECT-TYPE
    SYNTAX DisplayString (SIZE(1..64))
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This is the unique identifier of the forward port table.
         This ties this reverse port to the commanding forward port.
         This shall be a valid index in the hmtsFwdPortTable.  Setting
         this object to a value not in the hmtsFwdPortTable shall result
         in a Bad Value response.
         This object is non-volatile.
        "
    ::= { hmtsRevPortEntry 2 }

hmtsRevPortDescr  OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "This is a string of text describing the function of the port
         and physical location.  Such as 'HMS RF Receiver - Slot 3'.
         This value is vendor specific.
        "
    ::= { hmtsRevPortEntry 3 }

hmtsRevPortType OBJECT-TYPE
    SYNTAX INTEGER {
        rf    (1),
        rs485 (2),
        rs232 (3),
        other (4)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Serial communication type:
         rf    - This HMTS port uses an RF modem for reverse communication.
         rs485 - This HMTS port uses an RS-485 for reverse communication.
         rs232 - This HMTS port uses an RS-232 for reverse communication.
         other - This HMTS port uses a modem for reverse communication that
         is neither a RF, RS-485 or RS-232 modem.
        "
    ::= { hmtsRevPortEntry 4 }

hmtsRevPortAdminState OBJECT-TYPE
    SYNTAX INTEGER {
        enable (1),
        disable (2)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This allows the operator to disable/enable the processing of
         data received on this port.  When disabled no reverse path
         messages shall be received from this port.
         This object is non-volatile.
        "
    ::= { hmtsRevPortEntry 5 }

hmtsRevPortOperState OBJECT-TYPE
    SYNTAX INTEGER {
        noError (1),
        noFreqAssgn (2),
        freqUnlocked (3),
        portComFailure (4),
        otherError (5)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "This reports the operational status of the port.
         This object has a threshold property in the discreteProperty as
         describe in SCTE 38-1 (formerly HMS026).
        "
    ::= { hmtsRevPortEntry 6 }

hmtsRevFrequency OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "1 Hz"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This is the return path frequency of both the HMTS' reverse
         path modem and the network element's reverse path transmitter.
         This value only applies to the RF port type; all
         other port types should return noSuchName.  This value is
         used in the broadcasted CHNLDESC PDU.  When this value is set
         the CHNLDESC PDU shall be immediately issued.
         It is recommended that the commands to change the frequency
         be issued more than once, to insure proper setting.
         RESOLUTION: <=100 Hz
         This object is non-volatile.
        "
    ::= { hmtsRevPortEntry 7 }

hmtsRevMuteLvl OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "0.1 dBmV"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This level controls the automatic receiver Mute.  Power below
         this level will not be converted into a digital data.  This
         field only applies to the RF port type; all others port types
         should report Zero.  Attempts to set this value for non RF port
         types shall result in a bad value error.
         This object is non-volatile.
```

```
        "
    ::= { hmtsRevPortEntry 8 }

hmtsRevMulticastAddr OBJECT-TYPE
    SYNTAX     MacAddress
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "This is the Multicast address associated with all devices on
         this receiver.  When a device first registers with HMTS, this
         value is set as one of the devices multicast addresses.  This
         must be a valid entry in the hmtsMulticastAddrTable, not equal
         to ff-ff-ff-ff-ff-ff.  An attempt to set this value to
         ff-ff-ff-ff-ff-ff or to a multicast address not in the
         hmtsMulticastAddrTable will result in a bad value error.
         This object is non-volatile.

         The HMTS will automatically set one of the entries in the device's
         commonMulticastAddressTable to this address, such that all devices
         that are on this reverse path will have an entry in their
         commonMulticastAddressTable with this multicast address.
         Which entry in the commonMulticastAddressTable the HMTS sets is
         vendor specific.
        "
    ::= { hmtsRevPortEntry 9 }

hmtsRevReturnLvl OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "0.1 dBmV"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "This is the return level as recorded on the last response
         received on this receiver.  This value only applies to the RF
         port types; all other port types should return noSuchName Error.
         If the port has not received a response and therefore has not
         recorded a reverse return level then this objects shall return
         -999.  When the return level falls outside acceptable limits, as
         indicated by the properties of this value, the offending units
         hmtsDevComStat shall indicate a bad return level.  This value
         is a placeholder to globally affect the properties that really
         apply to the return level value in the hmtsDevTable.  In all cases
         the HMTS shall set hmtsDevReturnLvl for the device to this as
         received level upon receiving a valid massage.  The property
         thresholds and enable when set using this object shall be used for
         all hmtsDevReturnLvl in the device table using this receiver.
         This object never reports an alarm, hmtsDevReturnLvl does when the
         thresholds of this object are tripped.  Use hmtsDevReturnLvl
         threshold properties to affect a property change for an individual
         device.
         This object has a threshold property in the propertyTable as defined
         in SCTE 38-1 (formerly HMS026)
        "
    ::= { hmtsRevPortEntry 10 }

hmtsRevCRCErrors OBJECT-TYPE
```

```
    SYNTAX     Integer32
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This object maintains the number of packets received that have
       a bit error and do not pass CRC check.  This count is maintained
       until the HMTS is reset or this count is reset.  Writing to this
       object will reset the CRC count to Zero.
       UNITS: Packets
       This object has a threshold property in the propertyTable as defined
       in SCTE 38-1 (formerly HMS026)
      "
    ::= { hmtsRevPortEntry 11 }

hmtsRevFrameErrors OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This object maintains the number of packets received that have
       a framing error.  This count is maintained until the HMTS is
       reset or this count is reset.  Writing to this object will reset
       the frame error count to Zero.
       UNITS: Packets
       This object has a threshold property in the propertyTable as defined
       in SCTE 38-1 (formerly HMS026)
      "
    ::= { hmtsRevPortEntry 12 }

hmtsRevBackOffPeriod OBJECT-TYPE
    SYNTAX     Integer32 (0..16383)
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This is the backoff period as described in the
       commonBackoffPeriod object in the SCTE 38-3
       (formerly HMS024) SCTE-HMS-COMMON-MIB.
       This value is used by the HMTS to
       preset and maintain the value of commonBackoffPeriod
       on all devices on this receiver.
       This object is non-volatile.
      "
    DEFVAL     { 6 }
    ::= { hmtsRevPortEntry 13 }

hmtsRevACKTimeout OBJECT-TYPE
    SYNTAX     Integer32 (0..255)
    UNITS      "1 ms"
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "This is the acknowledge timeout as described in the
       commonACKTimeoutWindow object in the SCTE 38-3
       (formerly HMS024) SCTE-HMS-COMMON-MIB.
       This value is used by the HMTS to
```

```
        preset and maintain the value of commonACKTimeoutWindow
        on all devices on this receiver.
        This object is non-volatile.
       "
    DEFVAL     { 19 }
    ::= { hmtsRevPortEntry 14 }

hmtsRevMaxMACRetries OBJECT-TYPE
    SYNTAX     Integer32 (0..255)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the maximum retries as described in the
        commonMaximumMACLayerRetries object in the SCTE 38-3
        (formerly HMS024) SCTE-HMS-COMMON-MIB.
        This value is used by the HMTS to
        preset and maintain the value of
        commonMaximumMACLayerRetries
        on all devices on this receiver.
        UNITS: N/A
        This object is non-volatile.
       "
    DEFVAL     { 16 }
    ::= { hmtsRevPortEntry 15 }

hmtsRevBackOffMinExp OBJECT-TYPE
    SYNTAX     Integer32 (0..15)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the minimum exponent as described in the
        commonBackoffMinimumExponent object in the SCTE 38-3
        (formerly HMS024) SCTE-HMS-COMMON-MIB.
        This value is used by the HMTS to
        preset and maintain the value of
        commonBackoffMinimumExponent
        on all devices on this receiver.
        UNITS: N/A
        The value of this object must be less than or
        equal to hmtsRevBackOffMaxExp.
        This object is non-volatile.
       "
    DEFVAL { 6 }
    ::= { hmtsRevPortEntry 16 }

hmtsRevBackOffMaxExp OBJECT-TYPE
    SYNTAX     Integer32 (0..15)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
       "This is the maximum exponent as described in the
        commonBackoffMaximumExponent object in the SCTE 38-3
        (formerly HMS024) SCTE-HMS-COMMON-MIB.
        This value is used by the HMTS to
        preset and maintain the value of
        commonBackoffMaximumExponent
```

```
          on all devices on this receiver.
          UNITS: N/A
          The value of this object must be greater than or
          equal to hmtsRevBackOffMinExp.
          This object is non-volatile.
         "
     DEFVAL { 15 }
     ::= { hmtsRevPortEntry 17 }

-- **********************************************************************
-- *
-- *  HMTS Device Information Group
-- *
-- *  Description:
-- *    This group provides a common frame work for information about
-- *    each network element device that the HMTS has communicated.
-- *    This group is current for all termination systems.
-- *
-- **********************************************************************
hmtsDev OBJECT-TYPE
     SYNTAX     Integer32
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
         "This is the number of devices listed in the hmtsDevTable.
         "
     ::= { hmtsDeviceGroup 1 }

hmtsDevInErr OBJECT-TYPE
     SYNTAX     Integer32
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
         "This is the number of devices that are not registered or have
          a communication fault.  That is the number of devices listed in
          the hmtsComFaultTable.
         "
     ::= { hmtsDeviceGroup 2 }

hmtsDefaultCommString OBJECT-TYPE
     SYNTAX     DisplayString (SIZE(0..64))
     MAX-ACCESS read-write
     STATUS     current
     DESCRIPTION
         "In case hmtsCommManagementMethod has been set to manual, and only in
          this case, then hmtsDefaultCommString is the default community string
          to be assigned to hmtsDevCommString in the hmtsDevTable when a new
          row is created. This value will signal that the community string has
          not yet been assigned by the EMS.
          This object is non-volatile.
         "
     ::= { hmtsDeviceGroup 3 }

hmtsComStatAlarm OBJECT-TYPE
     SYNTAX     HmtsComStatCodes
     MAX-ACCESS not-accessible
```

```
    STATUS     current
    DESCRIPTION
      "This object is a global reference object for all hmtsDevComStat
       objects Properties.  It maintains a common property entry for all
       hmtsDevComStat object.  Changing this object properties will
       change the properties of hmtsDevComStat for each entry in the
       hmtsDevTable.

       This shall have a discrete property in the
       discretePropertyTable as defined in SCTE 38-1 (formerly HMS026).
       This object is never reported in an alarm trap, the specific
       hmtsDevComStat is.
      "
    ::= { hmtsDeviceGroup 4 }

hmtsContNRespCount OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This object is a global reference object for all
       hmtsDevContNRespCount objects properties.
       It maintains a common property entry for all
       hmtsDevContNRespCount objects.  Changing the objects properties
       will change the properties of hmtsDevContNRespCount for each
       entry in the hmtsDevTable.

       This shall have a threshold property in the propertyTable as defined
       in SCTE 38-1 (formerly HMS026).  This object is never reported
       in an alarm trap.  The specific hmtsDevContNRespCount is.
      "
    ::= { hmtsDeviceGroup 5 }

-- **********************************************************************
-- *
-- *  HMTS Device Table
-- *
-- **********************************************************************

hmtsDevTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF HmtsDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "The Device Table is the repository of all information known
       about each transponder.  The termination system keeps track of
       this information base on MAC Address.  The device table is an
       unabridged list of information.

       The contents of the entries in this table are non-volatile.
      "
    ::= { hmtsDeviceGroup 6 }

hmtsDevEntry OBJECT-TYPE
    SYNTAX     HmtsDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "A list of information about each device.
      "
    INDEX { hmtsDevPhysAddr }
    ::= { hmtsDevTable 1 }

HmtsDevEntry ::= SEQUENCE {
    hmtsDevPhysAddr
      MacAddress,
    hmtsDevIPAddr
      IpAddress,
    hmtsDevCommString
      DisplayString,
    hmtsDevFwdPortId
      DisplayString,
    hmtsDevRevPortId
      DisplayString,
    hmtsDevComStat
      HmtsComStatCodes,
    hmtsDevReturnLvl
      Integer32,
    hmtsDevLastStateChg
      Unsigned32,
    hmtsDevLastRespTime
      Unsigned32,
    hmtsDevRqstCount
      Integer32,
    hmtsDevRespTimeoutCount
      Integer32,
    hmtsDevContNRespCount
      Integer32,
    hmtsDevRegStatus
      INTEGER,
    hmtsDevRegTime
      Unsigned32,
    hmtsDevRowStatus
      RowStatus
}

hmtsDevPhysAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This is the physical MAC address of the network element that
       this device entry pertains.
      "
    ::= { hmtsDevEntry 1 }

hmtsDevIPAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the device IP address assigned of the network element
```

```
        that this device entry pertains.  The default IP address is
        0.0.0.0.  When one tries to set an IP address that is not accepted
        by the HMTS a bad value error is to be returned.  It is up to the
        HMTS to decide whether as IP address is acceptable or not (e.g. a HMTS
        using DHCP or a HMTS with router-functionality will use different
        criteria).  Also, two active entries in this Table shall not have the
        same IP address.  These restrictions to IP addresses only apply if the
        HMTS supports the IP based proxy.

        The IP address of an active row shall not be used in an active row of
        the hmtsMulticastAddrTable.
       "
    DEFVAL { '0000'h } -- '0000'h is IP address 0.0.0.0
    ::= { hmtsDevEntry 2 }

hmtsDevCommString OBJECT-TYPE
    SYNTAX     DisplayString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This is the community string assigned to the network element
        that this device entry pertains.  If hmtsCommManagementMethod
        is automatic then how the community string is set is vendor
        specific.  This object can not be set to a value that is
        already in use by another active entry.  Attempts to do so shall
        return a bad value error.  This restriction to community String
        only applies if the HMTS support the community string based proxy.
        DEFAULT: from hmtsDefaultCommString

        The community string of an active row shall not be used in an active
        row of the hmtsMulticastAddrTable.
       "
    ::= { hmtsDevEntry 3 }

hmtsDevFwdPortId OBJECT-TYPE
    SYNTAX     DisplayString (SIZE(0..64))
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This is the Id of the HMTS' forward path port or transmitter
        used to communicate with the corresponding network element.
        This uniquely identifies the table entry in the
        hmtsFwdPortTable. The forward port id is actually determined by
        the selection of the reverse port id, see the hmtsRevPortTable.
       "
    DEFVAL     { "" }
    ::= { hmtsDevEntry 4 }

hmtsDevRevPortId OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This is the id of the HMTS' reverse path port or receiver used
        to communicate with the corresponding network element.  This
        uniquely identifies the table entry in the hmtsRevPortTable.
        This value will default to an empty string upon row creation.
        This value must be assigned to an hmtsRevPortId prior to the
        activation of this entry. This value therefore can not be
        changed while the hmtsDevRowStatus is active.  Attempt to do so
        shall return a bad value error.  The changing of this value
        may result in the hmtsDevFwdPortId changing.
       "
    DEFVAL     { "" }
    ::= { hmtsDevEntry 5 }

hmtsDevComStat OBJECT-TYPE
    SYNTAX HmtsComStatCodes
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This reports the current device communication status with the
        corresponding network element.  Entries with the value of other
        than noError(1) shall be list in the hmtsComFaultTable.
        This object has a discrete property in the discretePropertyTable as
        defined in SCTE 38-1 (formerly HMS026).  The threshold limits and
        enables can be globally set by the hmtsComStatAlarm object's
        thresholds. Individual control of this object's threshold properties
        can be set through the threshold properties of this object.
        Default thresholds of this object shall be set to the current
        threshold settings of the hmtsComStatAlarm object's thresholds.
       "
    ::= { hmtsDevEntry 6 }

hmtsDevReturnLvl OBJECT-TYPE
    SYNTAX     Integer32
    UNITS      "0.1 dBmV"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This is the return level as recorded on the last response from
        this device.  This value only applies to transponders that are
        accessed via the serial RF link, noSuchName shall be returned when
        the device is accessed via other serial interfaces.  This value is
        reported in tenths of dBmV.  The value -999 shall be returned
        if a level has not yet been measured on this receiver.

        This object has a threshold property in the propertyTable as
        defined in SCTE 38-1 (formerly HMS026).  The threshold limits and
        enables can be globally set by the hmtsRevReturnLvl object's
        thresholds. Individual control of this object's threshold properties
        can be set through the threshold properties of this object.
        Alarms for this parameter shall report the hmtsRevPortId as the value
        of heCommonLogText within the heCommonAlarmEvent.
       "
    ::= { hmtsDevEntry 7 }

hmtsDevLastStateChg OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
```

```
      "This is the time of the last state change to this
       hmtsDevComStat object's value. All values are unsigned 32 bit
       integers.   Time since the Epoch (00:00:00, January 1,
       1970), measured in seconds(POSIX Time format).
      "
    ::= { hmtsDevEntry 8 }

hmtsDevLastRespTime OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "This is the time of the last response received from this
       device. All values are unsigned 32 bit integers.
       Time since the Epoch (00:00:00, January 1, 1970),
       measured in seconds(POSIX Time format).
      "
    ::= { hmtsDevEntry 9 }

hmtsDevRqstCount OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the total number of unicast requests made to this
       device.  This unsigned value is reset to zero by setting its
       value to zero.  Setting this value to a non-zero value will
       result in a bad value error.  The resetting of this will also
       reset hmtsDevRespTimeoutCount.
      "
    ::= { hmtsDevEntry 10 }

hmtsDevRespTimeoutCount OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the total number of unicast requests that have timed
       out.  This is an unsigned value that is reset to zero by
       setting its value to zero. Setting this value to a non-zero
       value will result in a bad value error.  The resetting of this
       will also reset hmtsDevRqstCount.
      "
    ::= { hmtsDevEntry 11 }

hmtsDevContNRespCount OBJECT-TYPE
    SYNTAX     Integer32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "This is the contiguous number of unicast requests that have
       timed out for this device.  Upon the receipt of a valid
       response this value is reset to zero.
       This object has a threshold property in the propertyTable as
       defined in SCTE 38-1 (formerly HMS026).  The threshold limits
       and enables can be globally set by the hmtsContNRespCount object's
```

```
          thresholds.  Individual control of this object threshold properties
          can be set through the threshold properties of this object.
          Default thresholds for this object shall be set to the current
          threshold settings of the hmtsContNRespCount object's thresholds.
         "
     ::= { hmtsDevEntry 12 }

hmtsDevRegStatus OBJECT-TYPE
     SYNTAX INTEGER {
        registered (1),
        registering (2),
        notRegistered (3)
     }
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This identifies the status of the device:
          Registered - the device has successfully registered.
          Registering - the device is in the process of registering.
          Not Registered - the device has not yet registered.
         "
     ::= { hmtsDevEntry 13 }

hmtsDevRegTime OBJECT-TYPE
     SYNTAX     Unsigned32
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
        "This is the time the device last completed registration.
          All values are unsigned 32 bit integers.
          Time since the Epoch (00:00:00 UTC, January 1, 1970),
          measured in seconds(POSIX Time format).
         "
     ::= { hmtsDevEntry 14 }

hmtsDevRowStatus OBJECT-TYPE
     SYNTAX     RowStatus
     MAX-ACCESS read-create
     STATUS     current
     DESCRIPTION
        "This is used to add and delete rows from this table dynamically.
          The RowStatus textual convention has been defined in the SNMPv2-TC
          MIB (RFC 2579). According to this RFC, the meaning of the different
          RowStatus states for this table is as follows:
          - active: the HMTS may perform unicast, multicast and broadcast
            communication with the device, using the communication parameters
            specified in the corresponding row. It does not matter whether or
            not there exists a physical device with the specified MAC address,
            or whether it is responding or not.
          - notInService: the HMTS may not perform any communication with the
            device with the specified MAC address. Traps for this device will
            not be forwarded to any EMS.
          - notReady: the RowStatus will be notReady if one or more of the
            following conditions is met:
            * the MAC address in the corresponding row is either
              all-zeroes, or a multicast MAC address.
```

```
           * the IP address is not 0.0.0.0, the HMTS is IP based and there is
             another active row with the same IP address.
           * the community string is not equal to hmtsDefaultCommString, the
             HMTS is community based and there is another active row with the
             same community string.
           * No forward port has been configured, or a non-existing forward
             port has been specified.

         During auto discovery if the row did not have a Reverse port ID
         assigned (that is the row status is notReady) and the reverse
         port id has been discovered then the HMTS may automatically
         activate the row.  However, if the row status is notInService
         than the HMTS shall not automatically activate the row.

         Entries that are not active in this table or have the default
         IP address or community string shall not appear in the
         tables hmtsIPDevTable and hmtsCommDevTable.
        "
    ::= { hmtsDevEntry 15 }

-- *********************************************************************
-- *
-- *  HMTS Device Communication Fault Table
-- *
-- *********************************************************************

hmtsComFaultTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsComFaultEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "This table identifies the MAC addresses of devices that have a
         problem in registering or communicating with the HMTS.  Only
         devices with hmtsDevComStat value other than noError(1) will
         have entries in this table.

         It is the responsability of the HMTS to keep the contents of the
         hmtsDevTable, hmtsComFaultTable, hmtsIPDevTable and hmtsCommDevTable
         consistent according to the following rules:
         - For each active row in hmtsDevTable whose hmtsDevComStat instance
           is different of noError(1), there must be a row with the same
           physical address in hmtsComFaultTable.
         - For each row in hmtsComFaultTable, there must be a
           corresponding row in hmtsDevTable with RowStatus active.
         - Each active row in the hmtsDevTable that has a valid IP address
           must also be present in the hmtsIPDevTable.
         - For each row in hmtsIPDevTable there must be a corresponding
           row in hmtsDevTable with rowstatus active.
         - Each active row in the hmtsDevTable that has a valid community
           string and whose RowStatus is active must also be present in
           hmtsCommDevTable.
         - For each row that exists in hmtsCommDevTable there must
           exist a row with the same community string in hmtsDevTable with
           RowStatus active.
         - An IP address is valid if there does not exist another row in the
           hmtsDevTable with the same IP address and RowStatus active, and if
```

```
          it has been assigned by either an external DHCP server, the
          internal DHCP server (if hmtsIPManagementMethod is set to client)
          or if it fits in one of the ranges defined by hmtsNetAddrTable
          (if hmtsIPManagementMethod is set to manualXP, manualHMTS or
          automatic).
        - A community string is valid if there does not exist another row in
          the hmtsDevTable with the same community string and RowStatus
          active.
        "
    ::= { hmtsDeviceGroup 7 }

hmtsComFaultEntry OBJECT-TYPE
    SYNTAX HmtsComFaultEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "A list of devices that are either not communicating or are not
        registered with the HMTS.
       "
    INDEX { hmtsComStatPhysAddr }
    ::= { hmtsComFaultTable 1 }

HmtsComFaultEntry ::= SEQUENCE {
    hmtsComStatPhysAddr
       MacAddress,
    hmtsComStat
       HmtsComStatCodes
}

hmtsComStatPhysAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "This is the physical MAC address of the network element that
        this entry pertains.
        "
    ::= { hmtsComFaultEntry 1 }

hmtsComStat OBJECT-TYPE
    SYNTAX HmtsComStatCodes
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This is a specific communication Status indicating the reason of this
        records entry exists.  This is the value of hmtsDevComStat in the
        hmtsDevTable
       "
    ::= { hmtsComFaultEntry 2 }

-- ***************************************************************************
-- *
-- *  HMTS Multicast Port Table
-- *
-- ***************************************************************************
```

```
hmtsMulticastAddrTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsMulticastAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "The multicast address table lists the known multicast/broadcast
       MAC addresses and provides a translation to either an IP address
       or a community string.
       The entries in this table are non-volatile.
      "
    ::= { hmtsDeviceGroup 8 }

hmtsMulticastAddrEntry OBJECT-TYPE
    SYNTAX HmtsMulticastAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "A list of information about each known multicast address.
      "
    INDEX { hmtsMulticastPhysAddr }
    ::= { hmtsMulticastAddrTable 1 }

HmtsMulticastAddrEntry ::= SEQUENCE {
    hmtsMulticastPhysAddr
      MacAddress,
    hmtsMulticastIPAddr
      IpAddress,
    hmtsMulticastCommString
      DisplayString,
    hmtsMulticastRowStatus
      RowStatus
}

hmtsMulticastPhysAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This is a known multicast or broadcast MAC address.
      "
    ::= { hmtsMulticastAddrEntry 1 }

hmtsMulticastIPAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is the assigned IP address associated to this Multicast or
       broadcast MAC address.  This is Mandatory for HMTS' that
       support IP based proxies.  This IP address shall not be used in an
       active row of the hmtsDevTable.
      "
    ::= { hmtsMulticastAddrEntry 2 }

hmtsMulticastCommString OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..64))
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This is the community string associated to this Multicast or
        broadcast MAC address.  This is current for HMTS' that
        support community string based proxies.  This Community String shall
        not be used in an active row of the hmtsDevTable.
       "
    ::= { hmtsMulticastAddrEntry 3 }

hmtsMulticastRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
       "This is used to add and delete rows from this table
        dynamically.

        IP address and/or Community Strings should be distinct from other
        entries in this table and in the hmtsDevTable.
       "
    ::= { hmtsMulticastAddrEntry 4 }

-- ***********************************************************************
-- *
-- *  HMTS IP Based Proxy Group
-- *
-- *  Description:
-- *    This group provides a basic frame work for HMTS that provide
-- *    IP based proxies.
-- *    This group is required for all HMTS' that support IP based
-- *    proxies
-- *
-- *  Concepts of Operation:
-- *    An HMTS that supports the IP based proxy as described in section
-- *    A.4.1 of SCTE 25-2 (formerly HMS004) May be implemented in
-- *    several ways.  The main difference is how the HMTS assigns an
-- *    IP address to the device in this mode.  The
-- *    hmtsIPManagementMethod describes the possible options.
-- *
-- *    When a device is first discovered the HMTS will attempt to
-- *    assign the IP address according to the method selected by
-- *    the hmtsIPManagementMethod.  If the selected method can not
-- *    assign IP for some reason then the device registration status
-- *    is put into registration pending and the ManualHmts method is
-- *    used to set the IP address.
-- *
-- *    The manual HMTS IP management method means the HMTS user or the
-- *    element manager are responsible for assigning the IP address.
-- *    The IP address is assigned by setting the hmtsDevIPAddr in the
-- *    hmtsDevTable.  The network device is put into the registration
-- *    pending state until the hmtsDevIPAddr is set with a valid IP
-- *    address.
-- *
-- *    The Automatic IP management method means the HMTS selects an
-- *    unused IP address is selected from the networks described in the
-- *    hmtsNetAddr Table.  If there are none available then the manual
-- *    HMTS IP management method is used.  The IP address that is
-- *    manually assigned must be contained within the networks described
-- *    in the hmtsNetAddr Table.
-- *
-- *    The manualXp IP management method means the HMTS attempts to use
-- *    the IP addressed assigned within the transponder if this IP
-- *    address is not unique then the manual HMTS management method is
-- *    used.
-- *
-- *    The client IP management method means the HMTS requests an IP address
-- *    for the newly discovered MAC address from a DHCP server.  The
-- *    DHCP server may be internal or external to the HMTS.  While this
-- *    request is pending the network element should be put into a
-- *    state of registration pending, until the DHCP server responds
-- *    with the IP Address.  Much care should be taken when using this
-- *    method to ensure that IP address leases do not expire.  The
-- *    IP address assignments must remain static for the life of the
-- *    network elements application.
-- ***********************************************************************

hmtsIPManagementMethod OBJECT-TYPE
    SYNTAX INTEGER {
      client (1),
      manualXp (2),
      manualHmts (3),
      automatic (4)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "Defines how IP addresses are handed out to transponders.
       client     - HMTS provides a gateway-proxy for serial devices
                    to either an external or an internal DHCP server.
       ManualXp   - Configuration of the IP address is manually set
                    prior to the Transponder being put on the network.
                    If the IP is invalid, then the IP address
                    assignment must be made manually using the HMTS
                    MAC device table.
       manualHmts - Configuration of the IP address is to be made
                    manually through the HMTS.  Setting the IP address
                    in the HMTS MAC device table does this.
       automatic  - HMTS automatically assigns IP addresses without
                    the use of a DHCP server.

       Note: is the responsability of the HMTS to keep the IP address stored
       in the device and in the corresponding row of the hmtsDevIPAddr
       column consistent. Each time the EMS changes the IP address in
       the device table, the HMTS must issue a SET_ADDR PDU to the device
       such that both IP addresses remain identical. See also SCTE 25-2
       (formerly HMS004).

       This object is non-volatile.
      "
    ::= { hmtsIPGroup 1 }
```

```
-- ***********************************************************************
-- *
-- *  HMTS Device by IP Table
-- *
-- ***********************************************************************

hmtsIPDevTable  OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsIPDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "The IP address is the index to this view of the hmtsDevTable.
       Only the devices that have a device state of registered are
       listed in this view.  This table is current for HMTS that
       support the IP proxy as defined in SCTE 25-2 (Formerly HMS004).

       It is the responsability of the HMTS to keep the contents of the
       hmtsDevTable, hmtsComFaultTable, hmtsIPDevTable and hmtsCommDevTable
       consistent according to the following rules:
       - For each active row in hmtsDevTable whose hmtsDevComStat instance
         is different of noError(1), there must be a row with the same
         physical address in hmtsComFaultTable.
       - For each row in hmtsComFaultTable, there must be a
         corresponding row in hmtsDevTable with RowStatus active.
       - Each active row in the hmtsDevTable that has a valid IP address
         must also be present in the hmtsIPDevTable.
       - For each row in hmtsIPDevTable there must be a corresponding
         row in hmtsDevTable with rowstatus active.
       - Each active row in the hmtsDevTable that has a valid community
         string and whose RowStatus is active must also be present in
         hmtsCommDevTable.
       - For each row that exists in hmtsCommDevTable there must
         exist a row with the same community string in hmtsDevTable with
         RowStatus active.
       - An IP address is valid if there does not exist another row in the
         hmtsDevTable with the same IP address and RowStatus active, and if
         it has been assigned by either an external DHCP server, the
         internal DHCP server (if hmtsIPManagementMethod is set to client)
         or if it fits in one of the ranges defined by hmtsNetAddrTable
         (if hmtsIPManagementMethod is set to manualXP, manualHMTS or
         automatic).
       - A community string is valid if there does not exist another row in
         the hmtsDevTable with the same community string and RowStatus
         active.
      "
    ::= { hmtsIPGroup 2 }

hmtsIPDevEntry  OBJECT-TYPE
    SYNTAX HmtsIPDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "A list of information about each device.
      "
    INDEX { hmtsIPDevAddr }
    ::= { hmtsIPDevTable 1 }
```

```
HmtsIPDevEntry ::= SEQUENCE {
    hmtsIPDevAddr
      IpAddress,
    hmtsIPPhysAddr
      MacAddress
}

hmtsIPDevAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This is the device IP address assigned of the network element
       that this device entry pertains.

      "
    ::= { hmtsIPDevEntry 1 }

hmtsIPPhysAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
      "This is the physical MAC address of the network element that
       this device entry pertains.
      "
    ::= { hmtsIPDevEntry 2 }

-- ************************************************************************
-- *
-- *  HMTS IP Table Address Assignment Table
-- *
-- ************************************************************************

hmtsNetAddrTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsNetAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
      "This table provides a list of networks from which the HMTS
       may select an IP address to automatically assign to a device.
       This table shall also be used to verify manually entered
       IP addresses (Device or Multicast) and IP Address originating
       from the transponders. This table will be ignored by the HMTS
       when hmtsIPManagementMethod is set to client.

       This table is mandatory if the HMTS
       supports IP address assignment as described in
       hmtsIPManagementMethod.

       Changing an existing entry in this table may cause IP address
       currently in use to become invalid.  The HMTS shall support the
       reassignment of IP addresses, by reassigning IP addresses of
       devices that no longer have valid IPs.  How this is accomplished
       is vender specific.
```

```
         This is a non-volatile table.
        "
    ::= { hmtsIPGroup 3 }

hmtsNetAddrEntry OBJECT-TYPE
    SYNTAX HmtsNetAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "A list of information about each network that this
         HMTS participates in for the access of the network
         elements that it acts as a gateway.
        "
    INDEX { hmtsNetStartAddr }
    ::= { hmtsNetAddrTable 1 }

HmtsNetAddrEntry ::= SEQUENCE {
    hmtsNetStartAddr
        IpAddress,
    hmtsNetEndAddr
        IpAddress,
    hmtsNetMask
        IpAddress,
    hmtsNetRowStatus
        RowStatus
}

hmtsNetStartAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "This is the first usable network IP address in the
         range of addresses that this entry describes.
       "
    ::= { hmtsNetAddrEntry 1 }

hmtsNetEndAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This is the last usable network IP address in the
         range of addresses that this entry describes.
       "
    ::= { hmtsNetAddrEntry 2 }

hmtsNetMask OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This is the network mask for the addresses that
         this entry describes.
       "
    ::= { hmtsNetAddrEntry 3 }

hmtsNetRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
      "This is used to add and delete rows from this table
       dynamically.
      "
    ::= { hmtsNetAddrEntry 4 }

-- ********************************************************************
-- *
-- *  HMTS Community Based Proxy Group
-- *
-- *  Description:
-- *    This group provides a basic frame work for HMTS that provide
-- *    community based proxies.
-- *    This group is current for all HMTS' that support Community
-- *    string based proxies.
-- *
-- *  Concept of Operation:
-- *     During auto discovery processing of the HMTS there are only two
-- *     methods of setting the community string used to perform the
-- *     proxy.  Neither of these methods are dependent upon the
-- *     registration state of the Transponder.  Therefore HMTS using
-- *     this method may be able to issue a registration complete
-- *     immediately.
-- *
-- ********************************************************************

hmtsCommManagementMethod OBJECT-TYPE
    SYNTAX INTEGER {
      automatic (1),
      manual (2)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
      "Defines how community strings are assigned to transponders.
       automatic  - HMTS automatically assigns each device a
                    its community string.  The user may not be able
                    to modify the community string in the hmtsDevTable.
                    The community string generated is normally based
                    upon the device's MAC address. In the SCTE 25-2
                    document (formerly HMS004), section A.4.2, there is
                    a recommended method.
       manualHmts - Configuration of the community string is to be made
                    manually through the HMTS.  Setting the community string
                    in the hmtsDevTable table does this.
       manualXp   - Configuration of the community string is manually set
                    prior to the transponder being put on the network.
                    If the community string is invalid, then the community
                    string assignment must be made manually using the
                    hmtsDevTable table.
```

```
         Note: is the responsability of the HMTS to keep the community strings
         in the device and in the corresponding row of the hmtsDevCommString
         column consistent. Each time the EMS changes the community string in
         the device table, the HMTS must issue an SNMP set command to the
         device (commonTrapCommunityString) such that these community strings
         remain equal.

         This object is non-volatile.
        "
    ::= { hmtsCommGroup 1 }

-- **********************************************************************
-- *
-- *  HMTS Device by Community String Table
-- *
-- **********************************************************************

hmtsCommDevTable OBJECT-TYPE
    SYNTAX SEQUENCE OF HmtsCommDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The Community String is the index to this view of the
         hmtsDevTable.  Only the devices that have successfully
         registered at least once are listed in this view.  This table is
         current for HMTS that support the community string proxy as
         defined in SCTE 25-2 (Formerly HMS004).

         It is the responsability of the HMTS to keep the contents of the
         hmtsDevTable, hmtsComFaultTable, hmtsIPDevTable and hmtsCommDevTable
         consistent according to the following rules:
         - For each active row in hmtsDevTable whose hmtsDevComStat instance
           is different of noError(1), there must be a row with the same
           physical address in hmtsComFaultTable.
         - For each row in hmtsComFaultTable, there must be a
           corresponding row in hmtsDevTable with RowStatus active.
         - Each active row in the hmtsDevTable that has a valid IP address
           must also be present in the hmtsIPDevTable.
         - For each row in hmtsIPDevTable there must be a corresponding
           row in hmtsDevTable with rowstatus active.
         - Each active row in the hmtsDevTable that has a valid community
           string and whose RowStatus is active must also be present in
           hmtsCommDevTable.
         - For each row that exists in hmtsCommDevTable there must
           exist a row with the same community string in hmtsDevTable with
           RowStatus active.
         - An IP address is valid if there does not exist another row in the
           hmtsDevTable with the same IP address and RowStatus active, and if
           it has been assigned by either an external DHCP server, the
           internal DHCP server (if hmtsIPManagementMethod is set to client)
           or if it fits in one of the ranges defined by hmtsNetAddrTable
           (if hmtsIPManagementMethod is set to manualXP, manualHMTS or
           automatic).
         - A community string is valid if there does not exist another row in
           the hmtsDevTable with the same community string and RowStatus
         active.
       "
    ::= { hmtsCommGroup 2 }

hmtsCommDevEntry OBJECT-TYPE
    SYNTAX HmtsCommDevEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "A list of information about each device.
       "
    INDEX { hmtsCommString }
    ::= { hmtsCommDevTable 1 }

HmtsCommDevEntry ::= SEQUENCE {
    hmtsCommString
       DisplayString,
    hmtsCommPhysAddr
       MacAddress
}

hmtsCommString OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..64))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
       "This is the community string assigned of the network element
        that this device entry pertains.
      "
    ::= { hmtsCommDevEntry 1 }

hmtsCommPhysAddr OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
       "This is the physical MAC address of the network element that
        this device entry pertains.
      "
    ::= { hmtsCommDevEntry 2 }


-- ************************************************************************
-- *
-- *  Traps sent by the HMTS
-- *
-- *
-- ************************************************************************

hmtsRegistrationFailedEvent NOTIFICATION-TYPE
    OBJECTS {
       hmtsDevComStat
       }
    STATUS current
    DESCRIPTION
      "The SNMP notification that is generated when registration of a device
```

```
        failed. There is only one mandatory varbind: hmtsDevComStat, e.g.
        (OID hmtsDevComStat.16.35.69.103.137.171, value ASN-INTEGER invIP).
        Since the hmtsDevTable index must be included in the OID, the physical
        address of a device is automatically included in the notification.
        In the above example, the physical address is 10-23-45-67-89-ab.
        Note: SMI rules forbid to include hmtsDevPhysAddr, since this column
        is not-accessible.
        The following varbinds must also be included if these apply:
        - For IP-based HTMS's, the IP-address hmtsDevIPAddr.
        - For community-string based HMTS's, the community string
          hmtsDevCommString.
       "
    ::= { hmtsNotifications 1 }


-- ********************************************************************
-- *
-- *  HMTS Conformance Information
-- *
-- *
-- ********************************************************************
heHMTSConformance    OBJECT IDENTIFIER ::= { heHMTSMIB 2 }

hmtsCompliances      OBJECT IDENTIFIER ::= { heHMTSConformance 1 }
hmtsGroups           OBJECT IDENTIFIER ::= { heHMTSConformance 2 }

-- Compliance Statements
heHMTSCompliance  MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
       "The compliance statement for the HMTS agents that support this MIB.
       "
    MODULE  -- this module
       MANDATORY-GROUPS {
          hmtsInformationGroup,
          hmtsMacProtocolInformationGroup,
          hmtsSnmpProtocolInformationGroup,
          hmtsReqManagementGroup,
          hmtsReqDeviceGroup,
          hmtsEventGroup
          }

       GROUP hmtsIPDeviceGroup
       DESCRIPTION
         "This group is mandatory if the HMTS supports the IP based Proxy.
         "

       GROUP hmtsCommDeviceGroup
       DESCRIPTION
         "This group is mandatory if the HMTS supports the community string
          based Proxy.
         "

       GROUP hmtsExtendedRegistrationGroup
       DESCRIPTION
         "This group is optional, objects extend the registration
```

```
          control functionality.
         "

       GROUP hmtsTrapControlGroup
       DESCRIPTION
         "This group is mandatory if the HMTS support asynchronous SNMP trap control
          of the HMS devices under its control.  That is the HMTS regularly broadcasts
          CONTMODE ON, OFF, RESUME, or INHIBIT PDUs for the collection of asynchronous
          SNMP Trap notification.
         "

       GROUP hmtsExtendedTrapControlGroup
       DESCRIPTION
         "This group is optional, objects extend content of the
          hmtsTrapControlGroup.
         "

       GROUP hmtsExtendedFwdPortGroup
       DESCRIPTION
         "This group is optional, objects extend content of the
          hmtsFwdPortTable.
         "

       GROUP hmtsExtendedRevPortGroup
       DESCRIPTION
         "This group is optional, objects extend content of the
          hmtsRevPortTable.
         "

     MODULE SCTE-HMS-HE-COMMON-MIB
       MANDATORY-GROUPS { heCommonTime,
                    heCommonAlarmDetectionControl,
                    heCommonParamsGroup,
                    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
                          }

     ::= { hmtsCompliances 1 }

-- MIB Compliance Groupings
hmtsReqManagementGroup  OBJECT-GROUP
     OBJECTS {
       hmtsRegInterval,
       hmtsRegContinuity,
       hmtsFwdPortAdminState,
       hmtsFwdPortDescr,
       hmtsFwdPortOperState,
       hmtsFwdPortType,
       hmtsFwdHmtsFrequency,
       hmtsFwdXpndrFrequency,
       hmtsRevPortAdminState,
       hmtsRevFwdPortId,
       hmtsRevPortDescr,
       hmtsRevPortType,
       hmtsRevFrequency,
       hmtsRevPortOperState,
       hmtsRevReturnLvl }
     STATUS     current
     DESCRIPTION
       "The collection of management objects which are required by all
         HMTS managers.
              "
     ::= { hmtsGroups 1 }

hmtsReqDeviceGroup  OBJECT-GROUP
     OBJECTS {
       hmtsDev,
       hmtsDevInErr,
       hmtsDefaultCommString,
       hmtsDevComStat,
       hmtsDevIPAddr,
       hmtsDevCommString,
       hmtsDevFwdPortId,
       hmtsDevRevPortId,
       hmtsDevReturnLvl,
       hmtsDevLastStateChg,
       hmtsDevLastRespTime,
       hmtsDevRqstCount,
       hmtsDevRespTimeoutCount,
       hmtsDevContNRespCount,
       hmtsDevRegStatus,
       hmtsDevRegTime,
       hmtsDevRowStatus,
       hmtsComStat,
       hmtsMulticastRowStatus
     }
     STATUS     current
     DESCRIPTION
       "This group defines the Device entry item required by all termination
         systems.
       "
     ::= { hmtsGroups 2 }

hmtsIPDeviceGroup  OBJECT-GROUP
     OBJECTS {
       hmtsMulticastIPAddr,
       hmtsIPManagementMethod,
       hmtsIPPhysAddr,
       hmtsNetEndAddr,
       hmtsNetMask,
       hmtsNetRowStatus
     }
     STATUS     current
     DESCRIPTION
       "This list the items required for an IP based HMTS.
       "
     ::= { hmtsGroups 3 }

hmtsCommDeviceGroup  OBJECT-GROUP
     OBJECTS {
       hmtsDefaultCommString,
       hmtsMulticastCommString,
       hmtsCommManagementMethod,
       hmtsCommPhysAddr
     }
     STATUS     current
     DESCRIPTION
       "This lists the items required for community string based HMTS.
       "
     ::= { hmtsGroups 4 }

hmtsInformationGroup  OBJECT-GROUP
     OBJECTS {
       hmtsAdminState,
       hmtsOperState,
       hmtsProxyType,
       hmtsFreqSwitchMethod,
       hmtsModel,
       hmtsSerialNumber,
       hmtsSoftwareVersion,
       hmtsTimeServerAddress,
       hmtsTimeServerSyncInterval
     }
     STATUS     current
     DESCRIPTION
       "The collection of info objects which are required by all
         HMTS entities.
       "
     ::= { hmtsGroups 5 }

hmtsMacProtocolInformationGroup  OBJECT-GROUP
     OBJECTS { hmtsMacPduTimeout,
       hmtsTalkPduTimeout,
       hmtsMacBroadcastDelay,
       hmtsAlarmDiscoveryMode,
       hmtsChnldescPduInt,
       hmtsTimePduInt,
       hmtsDeviceAccessMode  }
     STATUS     current
     DESCRIPTION
       "The collection of MAC protocol info objects which are required by all
         HMTS entities.
       "
     ::= { hmtsGroups 6 }

hmtsSnmpProtocolInformationGroup  OBJECT-GROUP
     OBJECTS { hmtsSnmpTimeout,
       hmtsSnmpBroadcastDelay }
     STATUS     current
     DESCRIPTION
       "The collection of SNMP protocol info objects which are required by all
         HMTS entities.
       "
     ::= { hmtsGroups 7 }

hmtsExtendedRegistrationGroup OBJECT-GROUP
     OBJECTS {
       hmtsRegMinDuration,
       hmtsRegMaxDuration
     }
     STATUS     current
     DESCRIPTION
```

```
        "The collection of registration control objects which are optional.
        "
    ::= { hmtsGroups 8 }

hmtsTrapControlGroup OBJECT-GROUP
    OBJECTS {
        hmtsTControlInterval,
        hmtsTControlMinDuration,
        hmtsTControlChainId,
        hmtsTControlContinuity,
        hmtsTControlRowStatus,
        hmtsTControlMulticastAddr
    }
    STATUS     current
    DESCRIPTION
        "The collection of objects which are required if the HMTS supports
         contention for the collection of SNMP traps.
        "
    ::= { hmtsGroups 9 }

hmtsExtendedTrapControlGroup OBJECT-GROUP
    OBJECTS {
        hmtsTControlMaxDuration
    }
    STATUS     current
    DESCRIPTION
        "The collection of trap control objects which are optional.
        "
    ::= { hmtsGroups 10 }

hmtsExtendedFwdPortGroup OBJECT-GROUP
    OBJECTS {
        hmtsFwdProvPwrLvl,
        hmtsFwdMaxPwrLvl,
        hmtsFwdPollTime
    }
    STATUS     current
    DESCRIPTION
        "The collection of hmtsFwdPortTable objects which are optional
        "
    ::= { hmtsGroups 11 }

hmtsExtendedRevPortGroup OBJECT-GROUP
    OBJECTS {
        hmtsRevMuteLvl,
        hmtsRevMulticastAddr,
        hmtsRevFrameErrors,
        hmtsRevCRCErrors,
        hmtsRevBackOffPeriod,
        hmtsRevACKTimeout,
        hmtsRevMaxMACRetries,
        hmtsRevBackOffMinExp,
        hmtsRevBackOffMaxExp
    }
    STATUS     current
    DESCRIPTION
```

```
       "The collection of hmtsRevPortTable objects which are optional
       "
    ::= { hmtsGroups 12 }

hmtsEventGroup OBJECT-GROUP
    OBJECTS {
       hmtsRegistrationFailedEvent
    }
    STATUS     current
    DESCRIPTION
       "Traps that can be generated by the HMTS.
       "
    ::= { hmtsGroups 13 }

END
```