

# SCTE • ISBE<sup>®</sup>

## S T A N D A R D S

---

**Network Operations Subcommittee**

---

**AMERICAN NATIONAL STANDARD**

**ANSI/SCTE 38-3 2017**

**Hybrid Fiber/Coax Outside Plant Status Monitoring  
SCTE-HMS-COMMON-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>.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2017  
140 Philips Road  
Exton, PA 19341

## Contents

<b>1. SCOPE</b>	<b>4</b>
<b>2. COPYRIGHT</b>	<b>4</b>
<b>3. NORMATIVE REFERENCES</b>	<b>4</b>
<b>4. INFORMATIVE REFERENCES</b>	<b>4</b>
<b>5. TERMS AND DEFINITIONS</b>	<b>4</b>
<b>6. REQUIREMENTS</b>	<b>5</b>

## 1.

### Scope

This document is identical to SCTE 38-3 2012 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 defines common information about NEs. This includes administrative information such as name, ID, model number, serial numbers vendor, and location; health indicators such as status and service state; and functional information such as power level and frequency range.

## 2.

### Copyright

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

## 3. Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this standard. At the time of subcommittee approval, the editions indicated were valid. All standards are subject to revision, and parties to agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

- 3.1. ANSI/SCTE 25-2 Hybrid Fiber Coax Outside Plant Status Monitoring - Media Access Control (MAC) Layer Specification v1.0
- 3.2. ANSI/SCTE 36 (formerly HMS 028), SCTE-ROOT Management Information Base (MIB) Definitions
- 3.3. ANSI/SCTE 37 (formerly HMS 072), Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-ROOTS Management Information Base (MIB) Definition
- 3.4. ANSI/SCTE 38-1 Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-PROPERTY-MIB Management Information Base (MIB) Definition
- 3.5. ANSI/SCTE 38-2 Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-ALARMS-MIB Management Information Base (MIB)
- 3.6. IETF RFC 1155 Structure and Identification of Management Information for TCP/IP-based Internets [RFC1155-SMI]
- 3.7. IETF RFC 1212 Concise MIB Definitions
- 3.8. IETF RFC 1213 MIB for Network Management of TCP/IP-based internets: MIBII
- 3.9. IETF RFC 1215 A Convention for Defining Traps for use with the SNMP
- 3.10. IETF RFC 4001 Textual Conventions for Internet Network Addresses

## 4.

### Informative References

The following documents may provide valuable information to the reader but are not required when complying with this standard.

None

## 5.

### 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.

## 6.

### **Requirements**

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

The syntax is given below.

```

-- *****
-- *
-- * Module Name: HMS024R14.MIB
-- *
-- * SCTE Status: ADOPTED FEBRUARY 15, 2002
-- *
-- * Description: Implements SCTE-HMS-COMMON-MIB definitions
-- *
-- * This MIB contains common information about NEs.
-- * MIB items cover administrative information
-- * such as name, ID, model number, serial numbers vendor, and location;
-- * health indicators such as status and service state; and functional
-- * information such as power level and frequency range.
-- *
-- * CAUTION: A SET OF ANY OBJECT STORED IN NON-VOLATILE MEMORY MAY RESULT IN AN ACTUAL WRITE TO NON-VOLATILE MEMORY,
-- * WHICH TYPICALLY HAS A FINITE LIFETIME. EXCESSIVE WRITING TO THESE OBJECTS COULD SHORTEN THE USEFUL LIFE OF THE
-- * DEVICE.
-- *
-- * Objects which are not present must not have the properties present either.
-- *
-- *****

```

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

IMPORTS

```

OBJECT-TYPE FROM RFC-1212
TRAP-TYPE FROM RFC-1215
Counter FROM RFC1155-SMI
NetworkAddress FROM RFC1155-SMI
DisplayString FROM RFC1213-MIB
scteHmsTree FROM SCTE-ROOT

```

```

-- *
-- * Common MIB prefix
-- *
commonIdent FROM SCTE-HMS-ROOTS
;

```

```

-- *
-- * Groups in the Common MIB
-- *

```

```

commonAdminGroup OBJECT IDENTIFIER ::= { commonIdent 1 }
commonMACGroup OBJECT IDENTIFIER ::= { commonIdent 2 }
commonMulticastGroup OBJECT IDENTIFIER ::= { commonIdent 3 }
commonStatsGroup OBJECT IDENTIFIER ::= { commonIdent 4 }
commonRfGroup OBJECT IDENTIFIER ::= { commonIdent 5 }

```

```

commonMacStats OBJECT IDENTIFIER ::= { commonStatsGroup 1 }

```

```
-- *
-- *
-- * The Transponder Administrative Group
-- *
-- * Implementation of the Transponder Administrative Group is mandatory.
-- * Within the group a number of objects are optional.
-- *
```

```
commonLogicalID OBJECT-TYPE
  SYNTAX OCTET STRING (SIZE(0..40))
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "Specifies the logical ID for the network element used by network
    management systems. The ID will often remain with a specific location
    regardless of any specific network element's characteristics.
    The value for this object is maintained in non-volatile memory."
  ::= { commonAdminGroup 1 }
```

```
commonVendor OBJECT-TYPE
  SYNTAX DisplayString (SIZE(0..255))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Displays the manufacturer of the network element.
    Displayed in human-readable format."
  ::= { commonAdminGroup 2 }
```

```
commonModelNumber OBJECT-TYPE
  SYNTAX DisplayString (SIZE(0..255))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Displays the model number of the network element.
    Displayed in human-readable format."
  ::= { commonAdminGroup 3 }
```

```
commonSerialNumber OBJECT-TYPE
  SYNTAX DisplayString (SIZE(0..255))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Returns the Serial Number of the Network Element.
    Displayed in human-readable format."
  ::= { commonAdminGroup 4 }
```

```
commonVendorInfo OBJECT-TYPE
  SYNTAX DisplayString (SIZE(0..255))
```

ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
 "Returns vendor-specific information.  
 Displayed in human-readable format."  
 ::= { commonAdminGroup 5 }

commonNEStatus OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(1))  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "Bit mask, identical to that used in the Status field of  
 the STATRESP PDU. See SCTE 25-2 (formerly HMS004).

Bit 0: CHNLRQST  
 Bit 1: CNTNRM  
 Bit 2: CNTCUR  
 Bit 3: MAJOR ALARMS  
 Bit 4: MINOR ALARMS  
 Bit 5: RSVD1  
 Bit 6: RSVD2  
 Bit 7: RSVD3"

::= { commonAdminGroup 6 }

commonReset OBJECT-TYPE  
 SYNTAX INTEGER { reset (1) }  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "This object is used to reset the NE by writing 1 to it.  
 Other values written to the object have no effect.  
 Reading the object returns a 1, and has no effect on the unit."  
 ::= { commonAdminGroup 7 }

commonAlarmDetectionControl OBJECT-TYPE  
 SYNTAX INTEGER {  
 detectionDisabled (1),  
 detectionEnabled (2),  
 detectionEnabledAndRegenerate (3)  
 }  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "This object is used to control the detection of alarms in this NE.  
  
 When a threshold from either the property table or the discrete  
 property table is crossed in a manner described by that MIB, then  
 an alarm is said to have occurred. When the alarm is detected, an



entry is placed in the alarm log table, which serves as a log of the most recent alarm events. When an alarm is detected, then an alarm trap is also generated. A property which is not in the nominal state will have an entry in the currentAlarmTable.

The detectionDisabled(1) value prevents the threshold detection process associated with the property table and discrete property table from running. The NE will not generate alarms. The contents of the alarmLogTable, currentAlarmTable, each instance of discreteAlarmState, and each instance of currentAlarmState remain in the state prior to detectionDisabled being applied.

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

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

The detectionEnabledAndRegenerate(3) value is transient. When setting commonAlarmDetectionControl to detectionEnabledAndRegenerate(3), the response will contain detectionEnabledAndRegenerate(3). Subsequent reads of the object report detectionEnabled(2).

The detectionDisabled(1) value does not affect the generation of hmsColdStart or hmsWarmStart traps. Traps added in the future are assumed to be unaffected by this object, unless stated in the description of that trap.

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

The value for this object is maintained in non-volatile memory."

```
::= { commonAdminGroup 8 }
```

commonNetworkAddress OBJECT-TYPE

SYNTAX NetworkAddress

ACCESS read-only

STATUS deprecated

DESCRIPTION

"This variable reports the network address (e.g., IP address) of the NE. This object was added to support the generation of traps. The value is retained in non-volatile memory. The value may be

set during the registration process by a MAC level command or via a vendor's craft interface.

With the introduction of IPv6 this object has been deprecated and `commonNetworkIpAddress` is to be used instead. See the description of `commonNetworkIpAddress` for details on how this object is to be used in conjunction with IPv6 addresses."

::= { commonAdminGroup 9 }

`commonCheckCode` OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object reports the Check Code for the transponder's configuration. The check code is generated from objects which are stored in non-volatile memory as well as those objects which reports the transponder's physical configuration, including monitored devices.

The algorithm used to generate the Check Code is vendor specific.

Upon a restart of the transponder, the current value of the check code is compared against the value previous to the restart to determine if a `hmsColdStart` or `hmsWarmStart` trap is to be generated. This requires that the value of the object be maintained in non-volatile memory for the comparison operation after a restart.

When a value is written to this object, the check code is recalculated, and the new value is returned in the `GetResponse` returned for the `SetRequest`. Since this optimization is NOT SNMP compliant, the transponder MAY respond in the SNMP compliant manner. Since this object description may impact an SNMP manager, it is recommended that the value be subsequently retrieved with a `GetRequest`, to guarantee that the desired value has been obtained.

In this case (on-demand recalculation), the `hmsColdStart` or `hmsWarmStart` traps are NOT generated.

The value for this object is maintained in non-volatile memory."

::= { commonAdminGroup 10 }

`commonTrapCommunityString` OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..64))

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object defines the community string reported in a trap.

The default value is 'public'.

Use of this object does not preclude the use of enterprise specific mechanisms to define community string values.

The value for this object is maintained in non-volatile memory."

```
::= { commonAdminGroup 11 }
```

commonTamperStatus OBJECT-TYPE

SYNTAX INTEGER { intact (1), compromised (2) }

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the status of the anti-tamper device of the network element.

this object requires an entry in the Discrete Property Table.

the intact(1) value represents ok, and the compromised(2) value

indicates an alarm."

```
::= { commonAdminGroup 12 }
```

commonInternalTemperature OBJECT-TYPE

SYNTAX INTEGER (-60..130)

ACCESS read-only

STATUS optional

DESCRIPTION

"Measured temperature inside the NE's case/housing.

Units degrees Celsius.

This item requires an entry in the properties MIB"

```
::= { commonAdminGroup 13 }
```

commonTime OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS optional

DESCRIPTION

"All values are unsigned 32 bit integers.

Time since the Epoch (00:00:00 UTC, January 1, 1970),

measured in seconds(POSIX)."

```
::= { commonAdminGroup 14 }
```

commonVarBindings OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object indicates the maximum number of entries in the

variable-binding list which can be accepted by the unit.

A value of 0 indicates no specified limit."

```
::= { commonAdminGroup 15 }
```

commonResetCause OBJECT-TYPE

SYNTAX INTEGER {

```

    other (1),
    powerup (2),
    command (3),
    watchdog (4),
    craft (5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This object reports the cause of the last reset."
::= { commonAdminGroup 16 }

```

commonCraftStatus OBJECT-TYPE

```

SYNTAX INTEGER {
    disconnected (1),
    connected (2)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This object reports the state of the craft interface.
    The status of this object does not imply any change to
    the transponder functionality through the MAC interface.
    Not all transponders support a craft interface.
    If the interface is not supported, the objects reports
    disconnected(2).

    This object requires an entry in the Discrete Property Table."
::= { commonAdminGroup 17 }

```

commonNetworkIpAddressType OBJECT-TYPE

```

SYNTAX InetAddressType
ACCESS read-only
STATUS mandatory
DESCRIPTION "Address mode of
commonNetworkIpAddress."
::= { commonAdminGroup 18 }

```

commonNetworkIpAddress OBJECT-TYPE

```

SYNTAX InetAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "This variable reports the network address
(e.g., IP address) of the NE. This object was added
to support the generation of traps. The value is
retained in non-volatile memory. The value may be
set during the registration process by a MAC level
command or via a vendor's craft interface."

```

If commonNetworkAddress is also implemented in this agent, this object is tied to it. If this object contains an IPv4 address, commonNetworkAddress, if implemented, will return the same address. If this object contains an IPv6 address, commonNetworkAddress will return 0.0.0.0."  
 ::= { commonAdminGroup 19 }

```
-- *
-- *
-- * The MAC Layer Group
-- *
-- * Implementation of the MAC Layer Group is mandatory.
-- * The objects are required to support the protocol defined
-- * by the Media Access Layer specification.
-- *
```

commonBackoffPeriod OBJECT-TYPE

SYNTAX INTEGER (0..16383)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The backoff period in milliseconds. This parameter is used as part of the random backoff period as defined by the Media Access Control Layer specification SCTE 25-2(formerly HMS004).

Initial Power up default: 6 milliseconds.

The value for this object is maintained in non-volatile memory."

::= { commonMACGroup 1 }

commonACKTimeoutWindow OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The Acknowledgement timeout window in milliseconds.

Initial Power up default: 19 milliseconds.

The value for this object is maintained in non-volatile memory."

::= { commonMACGroup 2 }

commonMaximumMALayerRetries OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The maximum number of retries a NE can make for a specific packet.

This limit is intended to protect the network from a unit with a faulty receiver.

Initial Power up default: 16.

The value for this object is maintained in non-volatile memory."

::= { commonMACGroup 3 }

commonMaxPayloadSize OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object reports the maximum length of the payload supported for a packet on the reverse or forward channel."

::= { commonMACGroup 4 }

commonBackoffMinimumExponent OBJECT-TYPE

SYNTAX INTEGER (0..15)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to define the minimum value of the exponent used for the Backoff Algorithm as defined by the Media Access Control Layer specification SCTE 25-2(formerly HMS004).

The default value is 6.

The value must be less than or equal to commonBackoffMaximumExponent

The value for this object is maintained in non-volatile memory."

::= { commonMACGroup 5 }

commonBackoffMaximumExponent OBJECT-TYPE

SYNTAX INTEGER (0..15)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object is used to define the maximum value of the exponent used for the Backoff Algorithm as defined by the Media Access Control Layer specification SCTE 25-2(formerly HMS004).

The default value is 15.

The value must be greater than or equal to commonBackoffMinimumExponent.

The value for this object is maintained in non-volatile memory."

::= { commonMACGroup 6 }

commonPhysAddress OBJECT-TYPE

SYNTAX OCTET STRING

ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "This variable reports the media dependant 'physical' address of  
 the NE. This object was added to support the generation of traps."  
 ::= { commonMACGroup 7 }

-- \*  
 -- \*  
 -- \* The Multicast Address Group  
 -- \*  
 -- \* Implementation of the Multicast Address Group is mandatory.  
 -- \*

commonMaxMulticastAddresses OBJECT-TYPE  
 SYNTAX INTEGER (4..255)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "Maximum number of multicast data link layer addresses  
 supported by this NE."  
 ::= { commonMulticastGroup 1 }

commonMulticastAddressTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF CommonMulticastAddressEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "The table containing the multicast addresses.  
  
 The value for all objects in this table are maintained  
 in non-volatile memory."  
 ::= { commonMulticastGroup 2 }

commonMulticastAddressEntry OBJECT-TYPE  
 SYNTAX CommonMulticastAddressEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "A list of information for multicast addresses."  
 INDEX { commonMulticastAddressIndex }  
 ::= { commonMulticastAddressTable 1 }

CommonMulticastAddressEntry ::=  
 SEQUENCE  
 {  
 commonMulticastAddressIndex  
 INTEGER,

```

        commonMulticastAddressNumber
            OCTET STRING
    }

commonMulticastAddressIndex OBJECT-TYPE
    SYNTAX INTEGER (1..255)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into commonMulticastAddressTable."
    ::= { commonMulticastAddressEntry 1 }

commonMulticastAddressNumber OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(6))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The multicast address. Only entries with the I/G bit set are
        accepted values. An empty entry is designated by setting
        the object to the broadcast address."
    ::= { commonMulticastAddressEntry 2 }

-- *
-- *
-- * The MAC Statistics Group
-- *
-- * Implementation of the MAC Statistics Group is optional.
-- *

commonForwardPathLOSEvents OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Count of forward path loss of signal (LOS) events. Can be reset to 0."
    ::= { commonMacStats 1 }

commonForwardPathFramingErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Count of framing errors on the forward path. Can be reset to 0."
    ::= { commonMacStats 2 }

commonForwardPathCRCErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-write

```



STATUS optional

DESCRIPTION

"Count of CRC errors on the forward path. Can be reset to 0."

::= { commonMacStats 3 }

commonInvalidMacCmds OBJECT-TYPE

SYNTAX Counter

ACCESS read-write

STATUS optional

DESCRIPTION

"Count of invalid MAC commands. Can be reset to 0."

::= { commonMacStats 4 }

-- \*

-- \*

-- \* The RF Interface Group

-- \*

-- \* Implementation of the RF Interface Group is mandatory for  
 -- \* those transponders which use RF Modems for the physical layer.

-- \*

-- \* Within the group a number of objects are optional. These  
 -- \* optional objects are used to support auto-negotiation of  
 -- \* power levels, forward channel frequency, and reverse channel  
 -- \* frequency. The algorithms associated with these objects  
 -- \* are vendor specific.

-- \*

commonReturnPathFrequency OBJECT-TYPE

SYNTAX INTEGER (0..1000000000)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Return path frequency. Units 1 Hz.

The value for this object is maintained in non-volatile memory."

::= { commonRfGroup 1 }

commonForwardPathFrequency OBJECT-TYPE

SYNTAX INTEGER (0..1000000000)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Forward path frequency. Units 1 Hz.

The value for this object is maintained in non-volatile memory."

::= { commonRfGroup 2 }

commonProvisionedReturnPowerLevel OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Return path power level to be used. Units 0.1 dBmV.

The value used internally will be rounded to the nearest supported value. The object reports the requested value, not the rounded value.

The value for this object is maintained in non-volatile memory."

::= { commonRfGroup 3 }

commonForwardPathReceiveLevel OBJECT-TYPE

SYNTAX INTEGER (-1000..1000)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the received forward path power level. Units 0.1 dBmV."

::= { commonRfGroup 4 }

commonMaxReturnPower OBJECT-TYPE

SYNTAX INTEGER (200..600)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The maximum return path power to be used by the NE. Units 0.1 dBmV.

The value for this object is maintained in non-volatile memory."

::= { commonRfGroup 5 }

-- \*

-- \*

-- \* The following definitions use the TRAP-TYPE macro as

-- \* defined in RFC1215.

-- \*

-- \* The community string is defined by commonTrapCommunityString.

-- \*

-- \* The hmsColdStart and hmsWarmStart traps are used rather than

-- \* the standard SNMP coldStart and warmStart, to permit the

-- \* physical address of the transponder to be reported in the trap.

-- \*

-- \*

hmsColdStart TRAP-TYPE

ENTERPRISE scteHmsTree

VARIABLES { commonPhysAddress, commonLogicalID }

DESCRIPTION

"A hmsColdStart trap signifies that the sending protocol entity is reinitializing itself such that the agent's configuration or the protocol entity implementation may be altered.

This trap is only issued by the transponder once the

registration has been completed successfully."  
::= 0

hmsWarmStart TRAP-TYPE

ENTERPRISE scteHmsTree

VARIABLES { commonPhysAddress, commonLogicalID }

DESCRIPTION

"A hmsWarmStart trap signifies that the sending protocol entity is reinitializing itself such that neither the agent's configuration nor the protocol entity implementation is altered.

This trap is only issued by the transponder once the registration has been completed successfully."

::= 2

END