



***Society of Cable
Telecommunications
Engineers***

**ENGINEERING COMMITTEE
Hybrid Management Sub-Layer Subcommittee**

AMERICAN NATIONAL STANDARD

ANSI/SCTE 38-5 2008

**Hybrid Fiber/Coax Outside Plant Status Monitoring
SCTE-HMS-FIBERNODE-MIB
Management Information Base (MIB) Definition**

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or nonmember of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members, whether used domestically or internationally.

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

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2008
140 Philips Road
Exton, PA 19341

Contents

1. SCOPE	1
2. COPYRIGHT	1
3. NORMATIVE REFERENCE	1
4. INFORMATIVE REFERENCE	1
5. TERMS AND DEFINITIONS	1
6. REQUIREMENTS	1

1. Scope

This document defines information about HFC optical fiber nodes. This includes information about the functional parts of a standard HFC optical fiber node, such as optical receivers, optical transmitters, ports, and power supplies.

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 37 (formerly HMS 072), Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-ROOTS Management Information Base (MIB) Definition
- 3.2. ANSI/SCTE 38-1 Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-PROPERTY-MIB Management Information Base (MIB) Definition
- 3.3. ANSI/SCTE 38-2 2005 Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-ALARMS-MIB Management Information Base (MIB)
- 3.4. IETF RFC 1155 Structure and Identification of Management Information for TCP/IP-based Internets [RFC1155-SMI]
- 3.5. IETF RFC 1212 Concise MIB Definitions
- 3.6. IETF RFC 1213 MIB for Network Management of TCP/IP-based internets: MIBII

4. Informative References

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

The syntax is given below.

```

-- *****
-- *
-- * Module Name: HMS025R13.MIB
-- *
-- * SCTE Status: ADOPTED April 2, 2002
-- *
-- * Description: Implements SCTE-HMS-FIBERNODE-MIB for Fiber Nodes.
-- *             This MIB intended for use on all fiber node equipment. This includes the US
-- *             strand mount or outside plant fiber nodes, and some pedestal models.
-- *             This MIB does not apply to the pedestal racks used in the European Cable
-- *             networks.
-- *
-- * EXCEPT AS NOTED, THE VOLATILITY OF CONTROL OBJECTS IN THIS MIB IS
-- * DETERMINED BY THE FIBER NODE APPLICATION
-- *
-- * Note:
-- * Objects which are not present must not have the properties present either. This applies to:
-- *
-- * a) Any table(s) not supported by a node; e.g., if fnNumberReturnLaser indicates 0, then
-- *    a GetRequest for any object in fnReturnLaserTable should return the SNMP error NOSUCHNAME,
-- *    and properties for those objects should not be accessible.
-- * b) Any object(s) not supported by a node; e.g., if fnReturnLaserTable is supported, but object
-- *    fnReturnLaserTemp is not, then a GetRequest for that object should return the SNMP error
-- *    NOSUCHNAME and properties for that object should not be accessible.
-- *
-- *****

```

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

IMPORTS

```

OBJECT-TYPE
    FROM RFC-1212
    DisplayString
    FROM RFC1213-MIB
    fnIdent
    FROM SCTE-HMS-ROOTS
;

```

fnAdminGroup OBJECT IDENTIFIER ::= { fnIdent 1 }

```

-- *****
-- * Administration Group
-- *****

```

```

fnVendorOID OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    ACCESS read-only
    STATUS optional
    DESCRIPTION

```

```

    "This object provides a means for a vendor to point to a vendor specific
    extension of this MIB."
 ::= { fnAdminGroup 1 }

fnDeviceId OBJECT-TYPE
    SYNTAX DisplayString ( SIZE(0..32) )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The content of this field is vendor specific. The intent is to provide
        manufacturer and/or product specific ASCII text information that will
        propagate to the manager's console verbatim. "
 ::= { fnAdminGroup 2 }

-- *****
-- * Return lasers
-- *****

fnNumberReturnLaser OBJECT-TYPE
    SYNTAX INTEGER ( 1..8 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of entries in the return laser table.
        A zero entry means the table does not exist and the functional
        area is not present in this device."
 ::= { fnIdent 2 }

fnReturnLaserTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FnReturnLaserEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table containing information about return lasers in a fiber node."
 ::= { fnIdent 3 }

fnReturnLaserEntry OBJECT-TYPE
    SYNTAX FnReturnLaserEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of information about each return laser in a fiber node."
    INDEX { fnReturnLaserIndex }
 ::= { fnReturnLaserTable 1 }

FnReturnLaserEntry ::=
    SEQUENCE
    {
        fnReturnLaserIndex

```

```

    INTEGER,
fnReturnLaserCurrent
    INTEGER,
fnReturnLaserTemp
    INTEGER,
fnReturnLaserControl
    INTEGER,
fnReturnLaserType
    DisplayString,
fnReturnLaserWavelength
    INTEGER,
fnReturnLaserOpticalPower
    INTEGER,
fnReturnLaserRFActive
    INTEGER
}

```

fnReturnLaserIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index into fnReturnLaserTable.

This index is application specific. It can be either the nth transmitter, or a slot number.

For example, a node may have 4 transmitters, numbered 1, 2, 3, 4, in slots 1, 3, 4, and 6.

Thus, the indexes could be .1, .3, .4, .6 (slots), OR .1, .2, .3., .4. (nth)"

::= { fnReturnLaserEntry 1 }

fnReturnLaserCurrent OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the return laser current. Units milliAmps.

This item requires an entry in the properties MIB."

::= { fnReturnLaserEntry 2 }

fnReturnLaserTemp OBJECT-TYPE

SYNTAX INTEGER (-40..80)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the temperature of the return laser. Units in 1 degrees Celsius.

This item requires an entry in the properties MIB."

::= { fnReturnLaserEntry 3 }

fnReturnLaserControl OBJECT-TYPE

SYNTAX INTEGER { off(1), on(2) }

```

ACCESS read-write
STATUS optional
DESCRIPTION
    "Controls the return path laser."
::= { fnReturnLaserEntry 4 }

fnReturnLaserType OBJECT-TYPE
SYNTAX DisplayString ( SIZE(0..20) )
ACCESS read-only
STATUS optional
DESCRIPTION
    "Type of return laser. Some valid values are: unisolated FP, isolated FP,
    uncooled DFB, cooled DFB, ITU(up to 32 colors)."
```

```

::= { fnReturnLaserEntry 5 }

fnReturnLaserWavelength OBJECT-TYPE
SYNTAX INTEGER ( 0..2147483647 )
ACCESS read-only
STATUS optional
DESCRIPTION
    "Wavelength of transmitted light from this return laser. Units 0.01 nanometers.
    Typical value might be 155056 (1550.56)"
::= { fnReturnLaserEntry 6 }

fnReturnLaserOpticalPower OBJECT-TYPE
SYNTAX INTEGER ( 0..65535 )
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Returns the optical power on the return laser. Units 0.1 milliWatts.
    This item requires an entry in the properties MIB."
::= { fnReturnLaserEntry 7 }

fnReturnLaserRFActive OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Returns the RF Active index associated with this return laser.
    If the RF active is undetermined, the value will be 0, and is NOT
    a valid index into the RF active table.

    Write access to this variable should only be provided if the RF Active
    path for this module can not be automatically determined and configured."
::= { fnReturnLaserEntry 8 }

-- *****
-- * Optical Receivers
-- *****
```


fnNumberOpticalReceiver OBJECT-TYPE

SYNTAX INTEGER (1..8)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of entries in the optical receiver table.

A zero entry means the table does not exist and the functional area is not present in the device."

::= { fnIdent 4 }

fnOpticalReceiverTable OBJECT-TYPE

SYNTAX SEQUENCE OF FnOpticalReceiverEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing information about the optical receivers in a fiber node."

::= { fnIdent 5 }

fnOpticalReceiverEntry OBJECT-TYPE

SYNTAX FnOpticalReceiverEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of information about each optical receiver in a fiber node."

INDEX { fnOpticalReceiverIndex }

::= { fnOpticalReceiverTable 1 }

FnOpticalReceiverEntry ::=

SEQUENCE

{

fnOpticalReceiverIndex

INTEGER,

fnOpticalReceiverPower

INTEGER,

fnOpticalReceiverState

INTEGER,

fnOpticalReceiverRFActive

INTEGER,

fnOpticalReceiverCurrent

INTEGER

}

fnOpticalReceiverIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index into fnOpticalReceiverTable.

This index is application specific. It can be either the nth receiver, or a slot number.
For example, a node may have 4 receivers, numbered 1, 2, 3, 4, in slots 1, 3, 4, and 6.
Thus, the indexes could be .1, .3, .4, .6 (slots), OR .1, .2, .3., .4. (nth)"

```
::= { fnOpticalReceiverEntry 1 }
```

fnOpticalReceiverPower OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Returns the received optical power. Units 0.1 milliWatts.

This item requires an entry in the properties MIB."

```
::= { fnOpticalReceiverEntry 2 }
```

fnOpticalReceiverState OBJECT-TYPE

SYNTAX INTEGER { off(1), on(2) }

ACCESS read-only

STATUS optional

DESCRIPTION

"Reports the state of this optical receiver."

```
::= { fnOpticalReceiverEntry 3 }
```

fnOpticalReceiverRFActive OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Returns the RF Active index associated with this optical receiver.

If the RF active is undetermined, the value will be 0, and is NOT
a valid index into the RF active table.

Write access to this variable should only be provided if the RF Active
path for this module can not be automatically determined and configured."

```
::= { fnOpticalReceiverEntry 4 }
```

fnOpticalReceiverCurrent OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the optical receiver current. Units milliAmps.

This item requires an entry in the properties MIB."

```
::= { fnOpticalReceiverEntry 5 }
```

```
-- *****
```

```
-- * EDFAs
```

```
-- *****
```

```

fnOpticalAmpPresent OBJECT-TYPE
    SYNTAX INTEGER { no(1), yes(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Indicates if an Optical Amplifier MIB is present.
         yes - Optical Amplifier MIB is supported by this device
         no - Optical Amplifier MIB is not supported."
 ::= { fnIdent 6 }

```

```

-- *****
-- * Forward RF Actives
-- *****

```

```

fnNumberRFActives OBJECT-TYPE
    SYNTAX INTEGER ( 1..16 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of RF actives for this fiber node.
         There must be a least one RF Active per fiber node.
         It is required to map the fnReturnLaserRFActive
         and fnOpticalReceiverRFActive objects to the ports with the same
         RFActive in the fnRFPortTable table.

```

The purpose of the RF active number is to provide a 'link' between the optical receivers and distribution ports.

Examples (not intended to cover all possibilities):

(Node with redundant receiver, monolithic amplifier)

Rcvr	Active	Ports	fnReturnLaserRFActive	fnOpticalReceiverRFActive	fnRFPortRFActive
a	1	1-4	1	1	1
b	1	1-4	1	1	1

(Small node with single receiver, monolithic amplifier)

Rcvr	Active	Ports	fnReturnLaserRFActive	fnOpticalReceiverRFActive	fnRFPortRFActive
a	1	1-3	1	1	1

(Larger node with 2 amplifier sections)

Rcvr	Active	Ports	fnReturnLaserRFActive	fnOpticalReceiverRFActive	fnRFPortRFActive
a	1	1-4	1	1	1
b	2	5-8	2	2	2

(Larger node with amplifier in ports, internal cables route signal from opticals to ports)

Rcvr	Active	Ports	fnReturnLaserRFActive	fnOpticalReceiverRFActive	fnRFPortRFActive
a	1	1-4	1	1	1
b	2	5-8	2	2	2

In this way no matter how complicated the fiber node the signal path can be traced through it from fiber to RF port.

Note also that it is not necessary to monitor the RF active; this table would show 1 for this object, but all of the objects in the table may not be supported."

```
::= { fnIdent 7 }
```

fnRFActiveTable OBJECT-TYPE

SYNTAX SEQUENCE OF FnRFActiveEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Table containing information about each RF Active."

```
::= { fnIdent 8 }
```

fnRFActiveEntry OBJECT-TYPE

SYNTAX FnRFActiveEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"List of information about each RF Active."

INDEX { fnRFActiveIndex }

```
::= { fnRFActiveTable 1 }
```

FnRFActiveEntry ::=

SEQUENCE

{

fnRFActiveIndex

INTEGER,

fnRFActiveControlType

DisplayString,

fnRFActiveOutputLevel

INTEGER,

fnRFActiveCurrent

INTEGER,

fnRFActiveControlLevel

INTEGER

}

fnRFActiveIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index into fnRFActiveTable."

```
::= { fnRFActiveEntry 1 }
```

fnRFActiveControlType OBJECT-TYPE

```

SYNTAX DisplayString ( SIZE(0..10) )
ACCESS read-only
STATUS optional
DESCRIPTION
    "Returns the control type of this fiber node. Possibilities include, but are not limited to:
        none - No control type inherent to this unit.
        alc - automatic level control
        asc - automatic slope control
        agc - automatic gain control
        alscl - automatic level slope control"
::= { fnRFActiveEntry 2 }

fnRFActiveOutputLevel OBJECT-TYPE
SYNTAX INTEGER ( 0..65535 )
ACCESS read-only
STATUS optional
DESCRIPTION
    "Returns the forward Path output RF level of a pilot signal on the amplifier. Units 0.1 dBmV.
    This item requires an entry in the properties MIB."
::= { fnRFActiveEntry 3 }

fnRFActiveCurrent OBJECT-TYPE
SYNTAX INTEGER ( 0..65535 )
ACCESS read-only
STATUS optional
DESCRIPTION
    "Returns the RF active current. Units milliAmps.
    This item requires an entry in the properties MIB."
::= { fnRFActiveEntry 4 }

fnRFActiveControlLevel OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS optional
DESCRIPTION
    "Returns the control level (as indicated by fnRFActiveControlType) for this RF active. Units 0.1VDC.
    This item has an entry in the properties MIB."
::= { fnRFActiveEntry 5 }

-- *****
-- * RF ports
-- *****

fnNumberRFPort OBJECT-TYPE
SYNTAX INTEGER ( 0..16 )
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Number of entries in the RF port table.

```

A zero entry means the table does not exist and the functional area is not present in the device."

::= { fnIdent 9 }

fnPortMasterAttenuationControl OBJECT-TYPE

SYNTAX INTEGER { low(1), high(2), pad(3) }

ACCESS read-write

STATUS optional

DESCRIPTION

"Reports and Controls the state of a reverse path attenuation switch that affects ALL ports.

low - No attenuation on the reverse path.

high - Typically high amount of attenuation on the reverse path.

This value may not be available for all switches.

pad - Typically a small amount of attenuation on the reverse path.

This value may not be available for all switches.

Note that the values for the object fnRFPortReverseAttenuationControl do NOT change when this object is accessed."

::= { fnIdent 10 }

fnRFPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF FnRFPortEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"Table containing information about the RF ports."

::= { fnIdent 11 }

fnRFPortEntry OBJECT-TYPE

SYNTAX FnRFPortEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"List of information about each RF port."

INDEX { fnRFPortIndex }

::= { fnRFPortTable 1 }

FnRFPortEntry ::=

SEQUENCE

{

fnRFPortIndex

INTEGER,

fnRFPortControlType

DisplayString,

fnRFPortControlLevel

INTEGER,

fnRFPortOutputRFLevel

INTEGER,

fnRFPortRFActive

```

    INTEGER,
    fnRFPortName
    DisplayString,
    fnRFPortReverseAttenuationControl
    INTEGER
}

```

fnRFPortIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index into the fnRFPortTable.

This index is application specific. It can be either the nth port, or a port number.

For example, a node may have 4 ports, numbered 1, 3, 4, 6. Thus, the indexes could be

.1, .3, .4, .6, OR .1, .2, .3., .4. In the latter case, the port name is critical."

::= { fnRFPortEntry 1 }

fnRFPortControlType OBJECT-TYPE

SYNTAX DisplayString (SIZE(0..10))

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the control type of this fiber node. Possibilities include, but are not limited to:

none - No control type inherent to this unit.

alc - automatic level control

asc - automatic slope control

agc - automatic gain control

alsc - automatic level slope control"

::= { fnRFPortEntry 2 }

fnRFPortControlLevel OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the control level (as indicated by fnRFPortControlType) for this RF port. Units 0.1VDC.

This item has an entry in the properties MIB."

::= { fnRFPortEntry 3 }

fnRFPortOutputRFLevel OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the RF Path output RF level of a pilot signal on the amplifier port. Units 0.1 dBmV.

This item requires an entry in the properties MIB.

This object shall report alarms using the value of fnRFPortName in

```

    the alarmText object in the hmsAlarmEvent Trap."
::= { fnRFPortEntry 4 }

fnRFPortRFActive OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Returns the RF Active index associated with this RF Port"
::= { fnRFPortEntry 5 }

fnRFPortName OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Physical name of Port. Some examples are Port 1 and Port 2.
    This name is put into the alarmText object used by hmsAlarmTrap when
    alarms are generated by objects in this table."
::= { fnRFPortEntry 6 }

fnRFPortReverseAttenuationControl OBJECT-TYPE
SYNTAX INTEGER { low(1), high(2), pad(3) }
ACCESS read-write
STATUS optional
DESCRIPTION
    "Reports and Controls the state of the reverse path
    attenuation switch for this port only.
    low - No attenuation on the reverse path.
    high - Typically high amount of attenuation on the reverse path.
    This value may not be available for all switches.
    pad - Typically a small amount of attenuation on the reverse path.
    This value may not be available for all switches."
::= { fnRFPortEntry 7 }

-- *****
-- * AB Switches
-- *****

fnNumberABSwitch OBJECT-TYPE
SYNTAX INTEGER ( 0..8 )
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Number of AB switches in this fiber node
    A zero entry means the table does not exist and the functional
    area is not present in the device."
::= { fnIdent 12 }

```


fnABSwitchTable OBJECT-TYPE
SYNTAX SEQUENCE OF FnABSwitchEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A table containing information about AB switches in this fiber node."
 ::= { fnIdent 13 }

fnABSwitchEntry OBJECT-TYPE
SYNTAX FnABSwitchEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"List of information about each AB switch."
INDEX { fnABSwitchIndex }
 ::= { fnABSwitchTable 1 }

FnABSwitchEntry ::=
SEQUENCE
{
 fnABSwitchIndex
 INTEGER,
 fnOpticalReceiverABSwitchFeedA
 INTEGER,
 fnOpticalReceiverABSwitchFeedB
 INTEGER,
 fnOpticalReceiverABSwitchState
 INTEGER,
 fnOpticalReceiverABSwitchSetting
 INTEGER,
 fnOpticalReceiverABSwitchSettingAccess
 INTEGER,
 fnOpticalReceiverABSwitchControl
 INTEGER
}

fnABSwitchIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Index into the fnABSwitchTable."
 ::= { fnABSwitchEntry 1 }

fnOpticalReceiverABSwitchFeedA OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Identifies the Optical receiver feeding Path A of this AB switch.
The value here is the index into the fnOpticalReceiverTable table."
::= { fnABSwitchEntry 2 }

fnOpticalReceiverABSwitchFeedB OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Identifies the Optical receiver feeding Path B of this AB switch
The value here is the index into the fnOpticalReceiverTable table."

::= { fnABSwitchEntry 3 }

fnOpticalReceiverABSwitchState OBJECT-TYPE

SYNTAX INTEGER { pathA(1), pathB(2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Reports the state of the AB switch.

pathA indicates that the switch is being fed by feed A.

pathB indicates that the switch is being fed by feed B.

This item requires an entry in the discrete properties MIB."

::= { fnABSwitchEntry 4 }

fnOpticalReceiverABSwitchSetting OBJECT-TYPE

SYNTAX INTEGER { forcePathA(1), forcePathB(2), preferPathA(3), preferPathB(4), default(5) }

ACCESS read-write

STATUS optional

DESCRIPTION

"Controls how the AB switch operates with the optical receivers.

Can only be written to when object fnOpticalReceiverABSwitchSettingAccess is 'ok'.

forcePathA - Forces the AB switch to the optical receiver identified by the feedA object.

Any automatic switching performed by the transponder is disabled.

forcePathB - Forces the AB switch to the optical receiver identified by the feedB object.

Any automatic switching performed by the transponder is disabled.

preferPathA - Automatic switching enabled, with path A preferred. Switching to path B occurs in a vendor-specific way, but the AB switch will return to path A when the condition causing the switch is no longer present.

preferPathB - Automatic switching enabled, with path B preferred. Switching to path A occurs in a vendor-specific way, but the AB switch will return to path B when the condition causing the switch is no longer present.

default - This value can only be written; it will never be read. It resets the transponder A/B

control to the factory-default state. After writing this value, this object will report one of the other four values.

Notes

1) If the node has *only* hardware switching, this object will fail all Set requests, indicating hardware has control.
In this case, the object fnOpticalReceiverABSwitchSettingAccess will indicate 'noAccess'. Although this table would not be required in this case, it is still useful, since fnOpticalReceiverABSwitchState can generate an alarm if set up properly.

2) If the node has a manually controlled switch inside, this object will fail all Set requests while the node is controlled by this local switch.
In this case, the object fnOpticalReceiverABSwitchSettingAccess will indicate 'noAccess'.

*** IMPORTANT ***

Options implemented depend on the model of the node.

It is NOT required that a node transponder support all enumerations. If a SET is done with a value that is not supported, a BADVALUE error must be returned.

If this object is supported, the object 'fnOpticalReceiverABSwitchAccess' MUST also be supported."

::= { fnABSwitchEntry 5 }

fnOpticalReceiverABSwitchSettingAccess OBJECT-TYPE

SYNTAX INTEGER { ok(1), noAccess(2) }

ACCESS read-write

STATUS optional

DESCRIPTION

"Controls access to the fnOpticalReceiverABSwitchSetting object.

ok - The fnOpticalReceiverABSwitchSetting can be written to with an actual affect

noAccess - fnOpticalReceiverABSwitchSetting cannot be set. Any attempt to write to fnOpticalReceiverABSwitchSetting while this object has a value of 'noAccess' will result in a BADVALUE error result.

(a) When written to 'noAccess', any write to the 'Setting' object will return a BADVALUE error, 'locking-down' the transponder control. Thus, changing the transponder control setting becomes a 2-step operation:

- 1) Set the fnOpticalReceiverABSwitchSettingAccess object to 'ok'
- 2) Set the fnOpticalReceiverABSwitchSetting object to the new desired value

(b) If the current state of the node prevents transponder control of the AB switch (e.g., a local switch is active), this object will show 'noAccess'.

In this case, neither the 'Setting' object nor the 'Access' object can be written to; attempts to do so will result in a BADVALUE error result.

```

    If supported, this object must be maintained in non-volatile memory."
 ::= { fnABSwitchEntry 6 }

fnOpticalReceiverABSwitchControl OBJECT-TYPE
    SYNTAX INTEGER { enabled(1), disabled(2) }
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "This object enables or disables TRANSPONDER control of the A/B switch. If disabled,
        the setting contained in fnOpticalReceiverABSwitchSetting has no effect.

    If supported, this object must be maintained in non-volatile memory."
 ::= { fnABSwitchEntry 7 }

```

```

-- *****
-- * AC Power
-- *****

```

```

fnLinePowerVoltage1 OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Returns the line power voltage from primary feed. Units 1VAC.
        This item requires an entry in the properties MIB."
 ::= { fnIdent 14 }

```

```

fnLinePowerVoltage2 OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Returns the line power voltage from a secondary feed. Units 1VAC.
        This item requires an entry in the properties MIB."
 ::= { fnIdent 15 }

```

```

fnLinePowerCurrent OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Returns the total current draw of the fiber node. Units 0.1 Amp.
        This item requires an entry in the properties MIB."
 ::= { fnIdent 16 }

```

```

-- *****
-- * Power Supplies
-- *****

```

```

fnNumberDCPowerSupply OBJECT-TYPE
    SYNTAX INTEGER ( 0..16 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Number of entries in the internal DC power supply table
        A zero entry means the table does not exist and the functional
        area is not present in the device."
    ::= { fnIdent 17 }

fnDCPowerSupplyMode OBJECT-TYPE
    SYNTAX INTEGER { loadsharing(1), switchedRedundant(2) }
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Indicates the mode, either load sharing or redundant (switched),
        in which the power supplies operate. This object should not
        be supported if the unit can only support one DC power supply."
    ::= { fnIdent 18 }

fnDCPowerTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FnDCPowerEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table containing information about the Regulated Power."
    ::= { fnIdent 19 }

fnDCPowerEntry OBJECT-TYPE
    SYNTAX FnDCPowerEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of information about the Regulated Power."
    INDEX { fnDCPowerIndex }
    ::= { fnDCPowerTable 1 }

FnDCPowerEntry ::=
    SEQUENCE
    {
        fnDCPowerIndex
            INTEGER,
        fnDCPowerVoltage
            INTEGER,
        fnDCPowerCurrent
            INTEGER,
        fnDCPowerName
            DisplayString
    }

```

fnDCPowerIndex OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index into the fnDCPowerTable."

::= { fnDCPowerEntry 1 }

fnDCPowerVoltage OBJECT-TYPE

SYNTAX INTEGER (-32768..32767)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Returns the regulated power voltage. Units in 0.1 Volts.

This item requires an entry in the properties MIB.

This object shall report alarms using the value of fnDCPowerName in the alarmText object in the hmsAlarmEvent Trap."

::= { fnDCPowerEntry 2 }

fnDCPowerCurrent OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the regulated power current. Units in 0.1 Amps.

This item requires an entry in the properties MIB.

This object shall report alarms using the value of fnDCPowerName in the alarmText object in the hmsAlarmEvent Trap."

::= { fnDCPowerEntry 3 }

fnDCPowerName OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Identifies the Physical name of the Power Supply. For example:

24 VDC Supply A

Actual value of this field is vendor specific, at a minimum it shall identify the nominal voltage expected and distinguish the supplies from one another.

If a single PHYSICAL supply provides multiple voltages, each voltage shall have its own entry in this table, with an appropriate name.

This name is put into the alarmText object in the hmsAlarmEvent Trap when alarms are generated by objects in this table."
::= { fnDCPowerEntry 4 }

END