SCTE·ISBE

STANDARDS

Network Operations Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 38-4 2017

Hybrid Fiber/Coax Outside Plant Status Monitoring
SCTE-HMS-PS-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.

Contents

1. Scope

This document is identical to SCTE 38-4 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 information commonly available from HFC power supplies. Its structure permits multiple power supplies to be monitored by a single transponder.

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   IETF RFC 1155, Structure and identification of management information for TCP/IP-based internets

3.2   ANSI/SCTE 37 2010 Hybrid Fiber/Coax Outside Plant Status Monitoring SCTE-HMS-ROOTS Management Information Base (MIB) Definition

4. Informative References

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

4.1   ANSI/SCTE 25-3 2011, Hybrid Fiber Coax Outside Plant Status Monitoring – Power Supply to Transponder Interface Bus (PSTIB) Specification v1.1

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-PS-MIB. It follows the IETF Simple Network Management Protocol (SNMP) for defining the managed objects. The syntax is given below.

-- ***
-- *
-- * Module Name: HMS027R13.MIB
-- *
-- * SCTE Status: ADOPTED FEBRUARY 15, 2002
-- *
-- * Description: This MIB contains information commonly available from HFC power supplies.
-- *   Its structure permits multiple power supplies to be monitored by a single transponder.
-- *   See SCTE 25-3 (formerly HMS 022) for additional information.
-- *
-- *   This MIB is not restricted to only power supplies with serial interfaces.
-- *
-- * Revisions from previous release:
-- * Edited OID descriptions only. For many of the OIDs ending with the word "support",   enhanced the description to indicate
-- * exactly what other OID is supported. Clarified the description for psEquipmentControl and psOutputCurrent.   -- * Expanded the definition of   psProtocolVersion to allow for the case where
there is no Transponder Interface Bus as defined by 25-3.
-- *
-- * Note:
-- *
-- * Objects which are not present must not have the properties present either. This applies to:
-- *
-- * a) Any object(s) not supported by a supply (e.g., if psOutputPowerSupport.1 indicates 'none', then
-- *   a GetRequest for psPowerOut.1 should return the SNMP error NoSuchName, and properties for
-- *   the psPowerOut object should not be accessible.
-- *
-- * b) All objects for a supply that is not present (e.g., if power supply 2 does not exist, then objects
-- *   indexed by power supply address 2 should not exist, nor should the properties.)
-- *
-- ***

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

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

-- /**********************************
-- * The Power Supply Group   *
-- **********************************/

psMonitored OBJECT-TYPE
    SYNTAX INTEGER (0..8)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION

        "Number of power supply connected to this NE."
    ::= { psIdent 1 }

-- * NOTE: The following description is standard SNMP, and is meant as clarification.
-- *
-- * If psMonitored is 0, the NOSUCHNAME error must be returned for any Get/Set against objects in the
-- * tables listed below. Likewise, the NOSUCHNAME error must be returned for any Get/Set using a psDeviceAddress
-- * index for a non-existant power supply. This is also the case if the device at the address specified
-- * is not a power supply.
-- *
-- * For example, if 2 power supplies are available, with addresses 3 and 4, then a Get/Set using
-- * address 2 MUST return NOSUCHNAME. Returning an actual value with no error under these conditions is
-- * misleading and incorrect.
-- *
-- * psDeviceTable
-- * psStringTable
-- * psBatteryTable
-- * psOutputTable
-- *
-- * These rules apply to other tables as well.

psDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PsDeviceEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Table containing information about the individual
         power supplies being monitored"
    ::= { psIdent 2 }

psDeviceEntry OBJECT-TYPE
    SYNTAX PsDeviceEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "List of information about each power supply being monitored."
    INDEX { psDeviceAddress }
    ::= { psDeviceTable 1 }

PsDeviceEntry ::=
    SEQUENCE
    {
        psDeviceAddress
            INTEGER,

-- * Power Supply Configuration

        psProtocolVersion
            INTEGER,

```
psSoftwareVersion
    DisplayString,

psDeviceId
    OCTET STRING,

psBatteries
    INTEGER,

psBatteryStrings
    INTEGER,

psTempSensors
    INTEGER,

psOutputs
    INTEGER,

psBatteryCurrentSupport
    INTEGER,

psFloatCurrentSupport
    INTEGER,

psOutputVoltageSupport
    INTEGER,

psInputVoltageSupport
    INTEGER,

psPowerSupplyTest
    INTEGER,

psMajorAlarmSupport
    INTEGER,

psMinorAlarmSupport
    INTEGER,

psTamperSupport
    INTEGER,

psBatteryVoltageSupport
    INTEGER,

psOutputPowerSupport
    INTEGER,
```

```
        psOutputFrequencySupport
            INTEGER,

        psInputCurrentSupport
            INTEGER,

        psInputPowerSupport
            INTEGER,

        -- * Power Supply Data

        psOutputVoltage
            INTEGER,

        psInputVoltage
            INTEGER,

        psInverterStatus
            INTEGER,

        psMajorAlarm
            INTEGER,

        psMinorAlarm
            INTEGER,

        psTamper
            INTEGER,

        psTotalStringVoltage
            INTEGER,

        psEquipmentControl
            INTEGER,

        psPowerOut
            INTEGER,

        psFrequencyOut
            INTEGER,

        psRMSCurrentIn
            INTEGER,

        psPowerIn
            INTEGER,

        psInputVoltagePresence
            INTEGER,
```

```
        psFrequencyIn
            INTEGER


    }

psDeviceAddress OBJECT-TYPE
    SYNTAX INTEGER ( 1..8 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into the psDeviceTable. For devices that implement ANSI/SCTE 25-3 (formally HMS022),
        this is the address of this device on  the RS-485 path."
    ::= { psDeviceEntry 1 }

-- * Power Supply Configuration

psProtocolVersion OBJECT-TYPE
    SYNTAX INTEGER ( 1..254 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Version of the SCTE HMS protocol implemented in the monitored
        equipment.  The 'Protocol Version' implementation will comply
        with the defined protocol in the SCTE 25-3 (formerly HMS 022) document with the
        corresponding revision number.

        Example:  A power supply implementing all commands and responses defined in
        SCTE 25-3 version 1.0 (formerly HMS 022) would return a value of 10 (decimal) in this field,
        reflecting major revision 1, minor revision 0.

        Any number returned that is less than 10 reflects a version of the SCTE 25-3 specification
        that had not yet been approved by SCTE.

        Transponders which are capable of appropriately rendering the data as defined by this MIB
        without implementing an interface as defined by SCTE 25-3 may respond with one of two values:
        [a] the transponder may return a value of zero (0), or [b] the transponder may return a value
        consistent with the SCTE 25-3 version that the transponder wants to make it appear it is
        supporting."

    ::= { psDeviceEntry 2 }

psSoftwareVersion OBJECT-TYPE
    SYNTAX DisplayString (SIZE(8))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The content of this field is vendor specific. The intent is to provide
         a text representation of the power supply or generator system software
```

version. Any printable ASCII characters can be included in this field.
NULL (0x00) characters are non-printable and are used to fill any unused
locations following the text data"

    ::= { psDeviceEntry 3 }

psDeviceId OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(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. The following special
      characters are defined in association with this field:'\' Used to cause
      a new line on the console display. Example: 'ALPHA\XM2 9015'
      would appear at the monitoring station as :
        ALPHA
        XM2 9015"
    ::= { psDeviceEntry 4 }

psBatteries OBJECT-TYPE
    SYNTAX INTEGER (0..8)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Current number of batteries per battery string."
    ::= { psDeviceEntry 5 }

psBatteryStrings OBJECT-TYPE
    SYNTAX INTEGER (0..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Current number of battery strings."
    ::= { psDeviceEntry 6 }

psTempSensors OBJECT-TYPE
    SYNTAX INTEGER (0..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Number of Battery temperature sensors."
    ::= { psDeviceEntry 7}

psOutputs OBJECT-TYPE
    SYNTAX INTEGER (1..5)
    ACCESS read-only
    STATUS mandatory

    DESCRIPTION
       "Number of power supply outputs."
   ::= { psDeviceEntry 8 }

psBatteryCurrentSupport OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
     "Bit Map that defines if battery current is measured in this installation.
     Bit set means this particular string supports this measurement.
     Bits   Addresses
     0   Not used
     1   1   String 1 has battery current support
     2   2   String 2 has battery current support
     3   3   String 3 has battery current support
     4   4   String 4 has battery current support
     .   .   ..
     .   .   ..
     .   .   ..
     n   n   String n has battery current support
     "
   ::= { psDeviceEntry 9 }

psFloatCurrentSupport OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
     "Bit Map that defines if float current is measured in this installation.
     Bit set means this particular string supports this measurement.
     Bits   Addresses
     0   Not used
     1   1   String 1 has float current support
     2   2   String 2 has float current support
     3   3   String 3 has float current support
     4   4   String 4 has float current support
     .   .   ..
     .   .   ..
     .   .   ..
     n   n   String n has float current support
     "
   ::= { psDeviceEntry 10 }

psOutputVoltageSupport OBJECT-TYPE
   SYNTAX INTEGER { none(1), supported(2) }
   ACCESS read-only

STATUS mandatory
DESCRIPTION
   "Defines if power supply supports monitoring of output voltage and psOutputVoltage:
   1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
      associated value in Get_Power_Supply_Data response.
   2 = Field is supported in this installation."
::= { psDeviceEntry 11 }

psInputVoltageSupport OBJECT-TYPE
   SYNTAX INTEGER { none(1), binary(2), analog(3) }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Defines if power supply supports monitoring of input or line voltage and psInputVoltage:
      1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
         associated value in Get_Power_Supply_Data response.
      2 = Field is supported - value in psInputVoltagePresence.
      3 = Field is supported - analog representation. value in psInputVoltage."
   ::= { psDeviceEntry 12 }

psPowerSupplyTest OBJECT-TYPE
   SYNTAX INTEGER { none(1), supported(2) }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Defines if power supply supports the remote test feature:
      1 = Function not supported.
      2 - Function is supported."
   ::= { psDeviceEntry 13 }

psMajorAlarmSupport OBJECT-TYPE
   SYNTAX INTEGER { none(1), supported(2) }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Defines if the power supply supports the major alarm indicator and psMajorAlarm:
      1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
         associated value in Get_Power_Supply_Data response.
      2 = Field is supported in this installation."
   ::= { psDeviceEntry 14 }

psMinorAlarmSupport OBJECT-TYPE
   SYNTAX INTEGER { none(1), supported(2) }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION

"Defines if the power supply supports the minor alarm indicator and psMinorAlarm:
    1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
      associated value in Get_Power_Supply_Data response.
    2 = Field is supported in this installation."
  ::= { psDeviceEntry 15 }

psTamperSupport OBJECT-TYPE
  SYNTAX INTEGER { none(1), supported(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Defines if the enclosure door switch is installed in this location and psTamper is supported:
     1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
       associated value in Get_Power_Supply_Data response.
     2 = Field is supported in this installation."
  ::= { psDeviceEntry 16 }

psBatteryVoltageSupport OBJECT-TYPE
  SYNTAX INTEGER { noMonitoring(1), totalString(2), both(3) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Defines the if batteries or string voltage are available:
      1 = No battery voltage monitoring is available.
      2 = Only full string battery voltage is available.
      3 = Both individual battery and full string voltages are available."
  ::= { psDeviceEntry 17 }

psOutputPowerSupport OBJECT-TYPE
  SYNTAX INTEGER { none(1), supported(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Defines if the output power measurement and psPowerOut is supported:
     1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
       associated value in Get_Power_Supply_Data response.
     2 = Field is supported in this installation."
  ::= { psDeviceEntry 18 }

psOutputFrequencySupport OBJECT-TYPE
  SYNTAX INTEGER { none(1), supported(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "Defines if the output frequency measurement and psFrequencyOut is supported
     1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
       associated value in Get_Power_Supply_Data response.

2 = Field is supported in this installation."
    ::= { psDeviceEntry 19 }

psInputCurrentSupport OBJECT-TYPE
    SYNTAX INTEGER { none(1), supported(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Defines if the input current measurement and psRMSCurrentIn is supported
       1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
         associated value in Get_Power_Supply_Data response.
       2 = Field is supported in this installation."
    ::= { psDeviceEntry 20 }

psInputPowerSupport OBJECT-TYPE
    SYNTAX INTEGER { none(1), supported(2) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Defines if the input power measurement and psPowerIn is supported
       1 = No support. For devices that implement ANSI/SCTE 25-3 (formally HMS022) discard
         associated value in Get_Power_Supply_Data response.
       2 = Field is supported in this installation."
    ::= { psDeviceEntry 21 }

-- * Power Supply Data

psOutputVoltage OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Power supply output voltage in 1/100 Volts units.
      This RMS value is common for all outputs in a multiple output system.
      This item requires an entry in the properties MIB"
    ::= { psDeviceEntry 22 }

psInputVoltage OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
      "Scaled representation of input 'line' or 'grid' voltage. This is an
      RMS value in 1/100 Volts units. This item requires an entry in the
      properties MIB."
    ::= { psDeviceEntry 23 }

psInverterStatus OBJECT-TYPE

```
     SYNTAX INTEGER { off(1), lineFail(2), testCycle(3), testStarted(4), testFailed(5) }
     ACCESS read-only
     STATUS optional
     DESCRIPTION
        "Status of power supply inverter.  Enumerated value indicates current
         status of inverter.
         1 = OFF,
         2 = ON: AC Line Fail,
         3 = ON: Local Test Cycle,
         4 = ON: Remote test initiated
         5 = ALARM:  Last Test Failed

         This item requires entries in the discrete properties MIB."

     ::= { psDeviceEntry 24 }

psMajorAlarm OBJECT-TYPE
     SYNTAX INTEGER { noAlarm(1), alarm(2) }
     ACCESS read-only
     STATUS optional
     DESCRIPTION
        "Service has been dropped or a service interruption is imminent.
         Indicates that an immediate truck roll is appropriate.
         Specific alarms and alarm nomenclature varies between vendors.
         Vendors should disclose all conditions that contribute to this
         alarm in appropriate product literature.

         1 = OK,
         2 = ALARM.

         This item requires entries in the discrete properties MIB."

     ::= { psDeviceEntry 25 }

psMinorAlarm OBJECT-TYPE
     SYNTAX INTEGER { noAlarm(1), alarm(2) }
     ACCESS read-only
     STATUS optional
     DESCRIPTION
        "A non-service effecting condition has occurred and should
         be monitored. Specific alarms and alarm nomenclature varies
         between vendors.  Vendors should disclose all conditions that
         contribute to this alarm in appropriate product literature.

         1 = OK,
         2 = ALARM

         This item requires entries in the discrete properties MIB."
```

```
   ::= { psDeviceEntry 26 }


psTamper OBJECT-TYPE
    SYNTAX INTEGER { closed(1), open(2) }
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Indicates status of enclosure door.  This notification is
        NOT included in the 'Major' or 'Minor' alarm fields.
        Individual users / installations must determine if a door
        open status represents an alarm and if so, of what severity.

        1 = CLOSED,
        2 = OPEN

        This item requires entries in the discrete properties MIB."

    ::= { psDeviceEntry 27 }

psTotalStringVoltage OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Scaled representation of the full battery string in 1/100 Volts units.

        This item requires an entry in the properties MIB."

    ::= { psDeviceEntry 28 }

psEquipmentControl OBJECT-TYPE
    SYNTAX INTEGER { stopTest(1), startTest(2) }
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Inverter test control. During AC Fail (psInverterStatus = 2)
        the result of changing this value is unspecified and vendor specific.

        1 = Discontinue inverter operation,
        2 = Begin inverter operation"

    ::= { psDeviceEntry 29 }

psPowerOut OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
```

"Representation of power supply output power in 1 W.
This item requires an entry in the properties MIB."

   ::= { psDeviceEntry 30 }

psFrequencyOut OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Scaled representation of the power supply output frequency in 1/100 Hz.
        This item requires an entry in the properties MIB."

   ::= { psDeviceEntry 31 }

psRMSCurrentIn OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Scaled representation of the power supply RMS input current in 1/100 A.
        This item requires an entry in the properties MIB."

   ::= { psDeviceEntry 32 }

psPowerIn OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Representation of the power supply input power in 1 W.
        This item requires an entry in the properties MIB."

   ::= { psDeviceEntry 33 }

psInputVoltagePresence OBJECT-TYPE
    SYNTAX INTEGER { lost(1), ok(2) }
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Digital value indicating that line voltage is present and within
        tolerance or not.
            1 = lost
            2 = ok.
        This item requires entries in the discrete properties MIB."
    ::= { psDeviceEntry 34 }

psFrequencyIn OBJECT-TYPE

```
      SYNTAX INTEGER { fiftyHz(1), sixtyHz(2) }
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
         "Operational frequency for input voltage"

      ::= { psDeviceEntry 35 }


-- * String data

psStringTable OBJECT-TYPE
      SYNTAX SEQUENCE OF PsStringEntry
      ACCESS not-accessible
      STATUS mandatory
      DESCRIPTION
         "Table containing strings data"
      ::= { psIdent 3 }

psStringEntry OBJECT-TYPE
      SYNTAX PsStringEntry
      ACCESS not-accessible
      STATUS mandatory
      DESCRIPTION
         "List of information about each string. Indexed by device and string number"
      INDEX { psStringDeviceAddress, psString }
      ::= { psStringTable 1 }

PsStringEntry ::=
      SEQUENCE
      {
         psStringDeviceAddress
            INTEGER,

         psString
            INTEGER,

         psStringChargeCurrent
            INTEGER,

         psStringDischargeCurrent
            INTEGER,

         psStringFloat
            INTEGER
      }


psStringDeviceAddress OBJECT-TYPE
```

```
        SYNTAX INTEGER ( 1..8 )
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Index into the psStringTable. Corresponds to psDeviceAddress in psDeviceTable."
        ::= { psStringEntry 1 }

psString  OBJECT-TYPE
        SYNTAX INTEGER ( 1..2 )
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Index into the psStringTable."
        ::= { psStringEntry 2 }

psStringChargeCurrent OBJECT-TYPE
        SYNTAX INTEGER ( 0..65535 )
        ACCESS read-only
        STATUS optional
        DESCRIPTION
            "Scaled representation of battery string charge current.
            This is an RMS value in 1/100 Amps.  When batteries being discharged,
            this value will = 0.
            This item requires an entry in the properties MIB."

        ::= { psStringEntry 3 }

psStringDischargeCurrent OBJECT-TYPE
        SYNTAX INTEGER ( 0..65535 )
        ACCESS read-only
        STATUS optional
        DESCRIPTION
            "Scaled representation of battery string discharge current. This is
            an RMS value in 1/100 Amps.If  multiple strings are installed but only
            one measurement sensor is used, this value represents the total battery
            discharge current. When batteries are being charged, this value will = 0.
            This item requires an entry in the properties MIB."

        ::= { psStringEntry 4 }


psStringFloat OBJECT-TYPE
        SYNTAX INTEGER ( 0..65535 )
        ACCESS read-only
        STATUS optional
        DESCRIPTION
            "Scaled representation of battery 'float' charge current in 1/100 Amps.
            This field will be '0' under conditions other than during
            actual float charging.  When this field is non-zero (reporting
```

float current), other battery current values (charge and discharge)
should be discarded. If multiple strings are installed but only
one measurement sensor is used, this field represents the total float
current.
This item requires an entry in the properties MIB."

   ::= { psStringEntry 5 }

-- * Battery Data

psBatteryTable OBJECT-TYPE
   SYNTAX SEQUENCE OF PsBatteryEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
     "Table containing batteries voltages"
   ::= { psIdent 4 }

psBatteryEntry OBJECT-TYPE
   SYNTAX PsBatteryEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
     "List of information about each battery.Indexed by device number and string"
   INDEX { psBatteryDeviceAddress, psBatteryString, psBattery }
   ::= { psBatteryTable 1 }

PsBatteryEntry ::=
   SEQUENCE
   {
      psBatteryDeviceAddress
         INTEGER,

      psBatteryString
         INTEGER,

      psBattery
         INTEGER,

      psBatteryVoltage
         INTEGER
   }

psBatteryDeviceAddress OBJECT-TYPE
   SYNTAX INTEGER (1..8)
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION

        "Index into the psBatteryTable. Corresponds to psDeviceAddress in psDeviceTable."
    ::= { psBatteryEntry 1 }

psBatteryString OBJECT-TYPE
    SYNTAX INTEGER (1..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into the psBatteryTable. Corresponds to psString in psStringTable."
    ::= { psBatteryEntry 2 }

psBattery OBJECT-TYPE
    SYNTAX INTEGER (1..8)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into the psBatteryTable."
    ::= { psBatteryEntry 3 }

psBatteryVoltage OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Scaled representation of an individual battery voltage in 1/100 Volts.
         String 'A' is used if only one battery string is active.
         This item requires an entry in the properties MIB."
    ::= { psBatteryEntry 4 }

-- * Output Data

psOutputTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PsOutputEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Table containing output currents"
    ::= { psIdent 5 }

psOutputEntry OBJECT-TYPE
    SYNTAX PsOutputEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "List of information about each Output port. Indexed by device and port number"
    INDEX { psOutputDeviceAddress, psOutput }
    ::= { psOutputTable 1 }

```
PsOutputEntry ::=
    SEQUENCE
    {
        psOutputDeviceAddress
            INTEGER,

        psOutput
            INTEGER,

        psOutputCurrent
            INTEGER
    }

psOutputDeviceAddress OBJECT-TYPE
    SYNTAX INTEGER ( 1..8 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into the psOutputTable.Corresponds to psDeviceAddress in psDeviceTable."
    ::= { psOutputEntry 1 }

psOutput  OBJECT-TYPE
    SYNTAX INTEGER ( 1..5 )
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Index into the psOutputTable.  Output number"
    ::= { psOutputEntry 2 }

psOutputCurrent OBJECT-TYPE
    SYNTAX INTEGER ( 0..65535 )
    ACCESS read-only
    STATUS optional
    DESCRIPTION
        "Scaled representation of power supply RMS output current in 1/100 Amps.

        If psOutputTable contains only one entry for a given power supply, this
        value is the total power supply output current (the power supply may
        have one or more outputs). If psOutputTable contains multiple entries
        for a power supply, the power supply has more than one output and each
        entry represents the current on a unique and separate output.
        This item requires an entry in the properties MIB."
    ::= { psOutputEntry 3 }


psTemperatureSensorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PsTemperatureSensorEntry
    ACCESS not-accessible
    STATUS mandatory
```

DESCRIPTION
    "Table containing temperature sensors information"
::= { psIdent 6 }

psTemperatureSensorEntry OBJECT-TYPE
    SYNTAX PsTemperatureSensorEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
      "List of information about each Temperature sensor. Indexed by device and Sensor number"
    INDEX { psTempDeviceAddress, psTemperatureSensor }
    ::= { psTemperatureSensorTable 1 }

PsTemperatureSensorEntry ::=
    SEQUENCE
    {
      psTempDeviceAddress
        INTEGER,

      psTemperatureSensor
        INTEGER,

      psTemperature
        INTEGER
    }

psTempDeviceAddress OBJECT-TYPE
    SYNTAX INTEGER (1..8)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Index into the psTemperatureSensorTable.Corresponds to psDeviceAddress in psDeviceTable."
    ::= { psTemperatureSensorEntry 1 }

psTemperatureSensor OBJECT-TYPE
    SYNTAX INTEGER (1..2)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "Index into the psTemperatureSensorTable. Temperature sensor number"
    ::= { psTemperatureSensorEntry 2 }

psTemperature OBJECT-TYPE
    SYNTAX INTEGER (-40..80)
    ACCESS read-only
    STATUS optional
    DESCRIPTION
      "Scaled representation of temperature.
      in degrees C with a range of -40 to + 80 degrees C.

This item requires an entry in the properties MIB."
    ::= { psTemperatureSensorEntry 3 }

END