Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

J. Quittek, Ed.
R. Winter
T. Dietz
D. Dudkowski
NEC Europe Ltd.

October 25, 2010

Definition of Managed Objects for Energy Management
draft-quittek-power-mib-02.txt

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects providing information about the energy consumption, the power states, and the battery status of managed devices and their components.

Status of this Memo

Copyright Notice

Table of Contents

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies MIB modules that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58,RFC 2580 [RFC2580].

2. Introduction

Energy management in communication networks is a topic that has been neglected for many years when energy was cheap and global warming not recognized. This has changed recently. Energy management is becoming a significant component of network planning, operations and management and new energy management strategies are currently being explored.

An essential requirement for energy management is collecting information on energy consumption and energy storage at managed devices.

An elementary step into this direction is monitoring power states. A power state defines a limitations of services provided by a device and implicitly limits energy consumption. Examples for commonly implemented power states include 'on', 'full power', 'low power', 'sleep', 'stand-by', and 'off'. There is no commonly agreed convention for power states naming and semantics. Therefore power states with the same names may have different semantics and different names may be in use for the same power state.

But the actual energy consumption of a device depends on more than just its power state. Also the current load, the kind of load, and many other factors influence energy consumption. If instrumentation is available, it is very helpful to receive information on the actual energy consumption of a device and its component. Providing this information requires much more effort than reporting power states, because a probe that measures (electrical) power is required. Typically this means not just adding several lines of software to a device, but also adding costly sensor hardware to it.

A third aspect to be considered for energy management is energy storage in batteries. It is helpful, for example, to monitor which device is running on batteries and which is charging its battery. Fortunately, the problem of instrumentation is often an easy one for devices with rechargeable batteries. Controlling the charging cycles needs instrumentation anyway and this instrumentation can also be used for providing battery status information.

This document defines a portion of the Management Information Base (MIB) that serves the three purposes sketched above:

- o monitoring power states of managed entities,
- o monitoring energy consumption of managed entities,
- o monitoring the status of batteries contained in or controlled by managed devices.

Supporting all three monitoring task will not make sense for every device. Many networked devices do not have batteries to be monitored and thus it would not make sense for them to implement managed objects for this purpose.

As mentioned above, instrumentation for measuring actual energy consumption is relatively expensive and it will not make sense for every managed device to provide sufficient instrumentation. In such a case it would not be appropriate to still implement managed objects for energy consumption monitoring.

This leads to the conclusion that the portions of the MIB for the three monitoring tasks listed above should be rather independent of each other and not combined in a single one. This document contains three MIB modules called Power State MIB, Energy MIB, and Battery MIB. The Energy MIB module uses an object defined in the Power State MIB module, but beyond that there is no dependency between the three modules. Obviously, any combination of the three modules is possible.

The definitions in this document are based on the requirements outlined in [I-D.quittek-power-monitoring-requirements].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Identifying Monitored Devices and Components

As argued in [I-D.quittek-power-monitoring-requirements] it is often required or at least desirable to not just monitor energy consumption and power state of an entire devices, but also of its contained

individual components. Furthermore it is argued in [I-D.quittek-power-monitoring-requirements] that there are cases where it is required that a managed device reports about energy consumption of one or more other, potentially remote devices. An example is a power strip reporting actual power and accumulated energy consumption of devices plugged into it.

It is not the purpose of MIB modules in this document to solve the problem of identifying components of the managed device that implements these modules or of components remote to this managed device. The task of identifying the entity that is subject of monitoring is left to other MIB modules, such as the ENERGY AWARE MIB module [I-D.parello-eman-energy-aware-mib], and the Entity MIB module [RFC4133].

As an open and flexible way of identifying the monitored entity, the MIB modules in this document use an OID as index that points into a MIB module used for identifying the monitored entity. For simplifying the trivial case that the monitored entity is identical with the device that implements the MIB module, an empty OID may be used.

4. Power State MIB

A number of devices today can operate in a number of different power states by reducing performance or going into standby mode or sleep mode. The Power State MIB module can be used for monitoring these states. Typically, not much instrumentation is needed for supporting the power state MIB module, because most devices with different power states are already equipped with means for controlling their these.

The Power State MIB module is structured into two tables, the powerCurrentStateTable reporting the current power state per entity and the powerStateTable providing statistics per power state. In addition, the Power State MIB module defines a notification that can be sent for informing the receiver about a change of an entity's current power state. For identifying the entity for which power state information is provided, OIDs are used, as explained in the previous section. Both tables use such an OID as their first index.

4.1. Current Power State Table

For basic monitoring of the actual power state of an entity, there is already a MIB module available: the Entity State MIB [RFC4268]. It reports the power state of an entity in object entStateStandby. It can have four different values: unknown(0), off(1), nonOperational(2), operational(3), see ENTITY-STATE-TC-MIB in

[RFC4268].

If this was considered to be sufficient, there would be no need for replicating this object in the power state MIB module. However, there is a concern that the three "known" states are too few for reflecting the variety of power saving states available today. For PCs, for example, there are several more states defined for the Advanced Configuration & Power Interface (ACPI). It might be useful to support several or all of these power states as suggested by [I-D.claise-energy-monitoring-mib].

The powerCurrentStateTable contains just a two objects per row:

```
powerStateTable(1)
+--powerStateEntry(1) [powerStateEnergyConsumerId]
   +-- --- Integer32        powerStateEnergyConsumerId(1)
   +-- --- ObjectIdentifier powerStateEnergyConsumerOid(2)
   +-- r-n SnmpAdminString  powerStateOperationalState(3)
   +-- rwn SnmpAdminString  powerStateAdminState(4)
```

Object powerStateOperationalState reports the actual power state of an entity at the time the object's value is retrieved. Object powerStateAdminState indicates a desired power state that the entity has been requested to enter, for example, by a network management system.

4.2. Power State Table

The second table called powerStateTable provides more detailed statistics for each power state. For this purpose it uses the power state name as another index object next to the entity index. This way, statistics can be reported per entity and per power state. The second index has the syntax of a SnmpAdminString and can be defined by the manufacturer of the device or MIB. In this way the index can fit many devices because the characteristics of the power state can be defined per device. The characteristics of the power state SHOULD be described as closely as possible in the object powerStateDescription.

```
      powerStateAllStatesTable(2)
      +--powerStateAllStatesEntry(1)
         [powerStateEnergyConsumerId,powerStateName]
         +-- --- SnmpAdminString powerStateName(1)
         +-- r-n Enumeration     powerStateType(2)
         +-- r-n SnmpAdminString powerStateDescription(3)
         +-- r-n Integer32       powerStateAveragePower(4)
         +-- r-n Integer32       powerStateMaximumPower(5)
         +-- r-n TimeTicks       powerStateTotalTime(6)
         +-- r-n TimeStamp       powerStateLastEnterTime(7)
         +-- r-n SnmpAdminString powerStateLastEnterReason(8)
         +-- r-n Counter64       powerStateEnterCount(9)
```

The offered statistics include the total time that the entity spent in a certain power state (powerStateTotalTime), the last time at which the entity entered a power state (powerStateLastEnterTime), the reason for entering it at the last time (powerStateLastEnterReason), the number of times a certain state has been entered (powerStateEnterCount), the average power consumed by the entity (powerStateAveragePower) and the maximum power consumed by the entity (powerStateMaximumPower).

5. Energy MIB

Devices that have instrumentation for measuring electrical energy consumption of entities can implement the Energy MIB module. Entities for which energy consumption is reported can be the entire devices, a component thereof or even an external entity for which the reporting devices observes the energy consumption.

The Energy MIB module defines two tables, the energyTable and the energyPerStateTable. The first one provides information on the instrumentations and on measured energy consumption of the entity. The second one provides energy consumption information for each individual power state.

5.1. Energy Consumption Table

The first set of managed objects in the energyTable are needed to help interpreting the energy consumption readings. These include the power supply type and voltage.

```
energyTable(1)
+--energyEntry(1) [energyConsumerId]
   +-- --- Integer32          energyConsumerId(1)
   +-- --- ObjectIdentifier   energyConsumerOid(2)
   +-- r-n EntitySensorStatus energySensorOperStatus(3)
   +-- r-n Unsigned32         energyNominalSupplyVoltage(4)
   +-- r-n Enumeration        energyElectricSupplyType(5)
   +-- r-n Unsigned32         energyTotalEnergy(6)
   +-- r-n UnitMultiplier     energyEnergyUnitMultiplier(7)
   +-- r-n Integer32          energyEnergyPrecision(8)
   +-- r-n Enumeration        energyMeasurementMethod(9)
   +-- r-n TimeStamp          energyDiscontinuityTime(10)
   +-- r-n Unsigned32         energySampleInterval(11)
   +-- r-n Unsigned32         energyMaxHistory(12)
   +-- r-n UnitMultiplier     energyPowerUnitMultiplier(13)
   +-- r-n Integer32          energyPowerPrecision(14)
   +-- r-n Unsigned32         energyRealPower(15)
   +-- r-n Unsigned32         energyPeakRealPower(16)
   +-- r-n Unsigned32         energyReactivePower(17)
   +-- r-n Unsigned32         energyApparentPower(18)
   +-- r-n Integer32          energyPhaseAngle(19)
   +-- r-n Integer32          energyPhaseAnglePrecision(20)
```

The main measured values provided by the table are the total energy consumed by the device and the current power (energy consumption rate). For entities supplied with alternating current (AC) there are also objects defined for reporting apparent power, reactive power and phase angle.

Provided energy and power values need to be multiplied by a unit multiplier given by a corresponding unit multiplier object in order to determine a measured value.

Measurements of the total energy consumed by an entity may suffer from interruptions in the continuous measurement of the current energy consumption. In order to indicate such interruptions, object energyDiscontinuityTime is provided for indicating the time of the last interruption of total energy measurement.

Time series of energy consumption values for past points in time are stored in the energyHistoryTable. Objects energySampleInterval and energyMaxHistory control the generation of entries in this table, see below.

5.2. Energy Consumption Per Power State Table

The second table in this module is called energyPerStateTable and it provides values of total energy consumption per power state in a way

similar to the powerStateTable in the Power State MIB module.

```
energyPerStateTable(2)
+--energyPerStateEntry(1) [energyConsumerId,powerStateName]
   +-- r-n Unsigned32 energyPerStateTotalEnergy(1)
```

5.3. Power History Table

The third table in this module is the energyHistoryTable. It stores total energy consumtion values for past points in time.

```
energyHistoryTable(3)
+--energyHistoryEntry(1) [energyConsumerId,energyHistoryIndex]
   +-- --- Unsigned32 energyHistoryIndex(1)
   +-- r-n TimeStamp  energyHistoryTimestamp(2)
   +-- r-n Unsigned32 energyHistoryTotalEnergy(3)
```

Creation of entries in this table is controlled by the values of corresponding objects energySampleInterval and energyMaxHistory in the energyTable.

Entries are indexed by the the entity (energyConsumerId) and by energyHistoryIndex. The first entry created for a certain entity in the table always has an energyHistoryIndex with a value of 1. Further entries for the same entity get increasing consecutive indices until the maximum index value given by object energyMaxHistory is reached. Then, no further indices will be used, but the entry with the oldest timestamp will be overwritten each time a new entry needs to be created.

A new entry is created with a time difference given by object energySampleInterval after creation of the previous entry. Hence, the difference between timestamps energyHistoryTimestamp of two consecutive entries SHOULD be equal to the value of object energySampleInterval.

6. Battery MIB

Editor's note: The Battery MIB module still uses the entPhysicalIndex from the ENTITY MIB. This will be changed in the next revision.

The third MIB module defined in this document defines objects for reporting information about batteries. The batteryTable contained in the Batter MIB module is again a sparse augment of the Entity MIB module [RFC4133]. It uses one row per battery and require that every battery for which information is provided has its own entry in the entPhysicalTable of the Entity MIB module.

The kind of entity in the entPhysicalTable is indicated by the value of enumeration object entPhysicalClass. Since there is no value called 'battery' defined for this object, it is RECOMMENDED that for batteries the value of this object is chosen to be powerSupply(6).

The batteryTable contains three groups of objects. The first group describes the battery in more detail than the generic objects in the entPhysicalTable. The second group of objects report on the current battery state, if it is charging or discharging, how much it is charged, its remaining capacity, the number of experienced charging cycles, etc.

```
batteryTable(1)
+--batteryEntry(1) [entPhysicalIndex]
   +-- r-n Enumeration batteryType(1)
   +-- r-n Enumeration batteryTechnology(2)
   +-- r-n Unsigned32  batteryNominalVoltage(3)
   +-- r-n Unsigned32  batteryNumberOfCells(4)
   +-- r-n Unsigned32  batteryNominalCapacity(5)
   +-- r-n Unsigned32  batteryRemainingCapacity(6)
   +-- r-n Counter32   batteryChargingCycleCount(7)
   +-- r-n DateAndTime batteryLastChargingCycleTime(8)
   +-- r-n Enumeration batteryState(9)
   +-- r-n Unsigned32  batteryCurrentCharge(10)
   +-- r-n Unsigned32  batteryCurrentChargePercentage(11)
   +-- r-n Unsigned32  batteryCurrentVoltage(12)
   +-- r-n Integer32   batteryCurrentCurrent(13)
   +-- r-n Unsigned32  batteryLowAlarmPercentage(14)
   +-- r-n Unsigned32  batteryLowAlarmVoltage(15)
   +-- r-n Unsigned32  batteryReplacementAlarmCapacity(16)
   +-- r-n Unsigned32  batteryReplacementAlarmCycles(17)
```

The third group of objects in this table indicates thresholds which can be used to raise an alarm if a property of the battery exceeds one of them. Raising an alarm may include sending a notification. The Battery MIB defines two notifications, one indicating a low battery charging state and one indicating an aged battery that may need to be replaced.

7. Relationship to Other MIB Modules

The three MIB modules described above relate to a number of existing standard MIB modules and complements them where necessary.

This section needs to be revised.

8. Definitions

8.1. Power State MIB

```
   POWER-STATE-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       mib-2, Integer32, Counter64, TimeTicks
           FROM SNMPv2-SMI                                -- RFC2578
       TimeStamp
           FROM SNMPv2-TC                                 -- RFC2579
       MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
           FROM SNMPv2-CONF                               -- RFC2580
       SnmpAdminString
           FROM SNMP-FRAMEWORK-MIB;                       -- RFC3411

   powerStateMIB MODULE-IDENTITY
       LAST-UPDATED "201010231200Z"         -- 23 October 2010
       ORGANIZATION "IETF OPSAWG Working Group"
       CONTACT-INFO
           "General Discussion: opsawg@ietf.org
           To Subscribe: https://www.ietf.org/mailman/listinfo/opsawg
           Archive: http://www.ietf.org/mail-archive/web/opsawg

           Co-editor:
             Juergen Quittek
             NEC Europe Ltd.
             NEC Laboratories Europe
             Kurfuersten-Anlage 36
             69115 Heidelberg
             Germany
             Tel: +49 6221 4342-115
             Email: quittek@neclab.eu

           Co-editor:
             Thomas Dietz
             NEC Europe Ltd.
             NEC Laboratories Europe
             Kurfuersten-Anlage 36
             69115 Heidelberg
             Germany
             Phone: +49 6221 4342-128
             Email: Thomas.Dietz@neclab.eu"

       DESCRIPTION
           "This MIB module defines a set of objects for monitoring
           the power state of managed entitites.
```

```
        Copyright (c) 2010 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this MIB module is part of RFC yyyy; see
        the RFC itself for full legal notices."
   -- replace yyyy with actual RFC number & remove this notice

   --  Revision history

       REVISION     "201010231200Z"         -- 23 October 2010
       DESCRIPTION
           "Initial version, published as RFC yyyy."
   -- replace yyyy with actual RFC number & remove this notice

       ::= { mib-2 9991 }
   -- xxx to be assigned by IANA.

   --******************************************************************
   -- Top Level Structure of the MIB module
   --******************************************************************

   powerStateNotifications OBJECT IDENTIFIER ::= { powerStateMIB 0 }
   powerStateObjects       OBJECT IDENTIFIER ::= { powerStateMIB 1 }
   powerStateConformance   OBJECT IDENTIFIER ::= { powerStateMIB 2 }

   --==================================================================
   -- 1. Object Definitions
   --==================================================================

   -------------------------------------------------------------------
   -- 1.1. Actual Power State Table
   -------------------------------------------------------------------
   powerStateTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF PowerStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table provides information on the current power state
           of managed entities.

           The table is indexed by an ID of the entity on which power
```

```
            state information is provided. IDs can be provided by
            another MIB module, such as the ENERGY AWARE MIB module
            or the ENTITY MIB module. If not ID provisioning from other
            MIB modules is available, the table can only have one entry
            for reporting the local power state of the device that tuns
            an instance of this table."
        ::= { powerStateObjects 1 }

    powerStateEntry OBJECT-TYPE
        SYNTAX      PowerStateEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An entry providing information on the current power state
            of an entity."
        INDEX  { powerStateEnergyConsumerId }
        ::= { powerStateTable 1 }

    PowerStateEntry ::=
        SEQUENCE {
           powerStateEnergyConsumerId   Integer32,
           powerStateEnergyConsumerOid  OBJECT IDENTIFIER,
           powerStateOperationalState   SnmpAdminString,
           powerStateAdminState         SnmpAdminString
        }

    powerStateEnergyConsumerId OBJECT-TYPE
        SYNTAX      Integer32 (0..2147483647)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An integer that identifies an entity that is subject of
            power state monitoring. Index values MUST be locally unique
            for each identified entity.

            If an implementation of the ENERGY AWARE MIB module is
            available in the local SNMP context, then the same index of
            an entity MUST be chosen as assigned to the entity by object
            pmIndex in the ENERGY AWARE MIB module. In this case,
            entities without an assigned value for pmIndex cannot be
            indexted by the powerCurrentStateTable.

            If there is no implementation of the ENERGY AWARE MIB module
            but one of the ENTITY MIB module is available in the local
            SNMP context, then the same index of an entity MUST be chosen
            as assigned to the entity by object entPhysicalIndex in the
            ENTITY MIB module. In this case, entities without an assigned
            value for pmIndex cannot be indexted by the
```

```
            powerCurrentStateTable.

            If neither the ENERGY AWARE MIB module nor of the ENTITY MIB
            module is available in the local SNMP context, then this MIB
            module may choose identity values from a further MIB module
            providing entity identities. In this case the value for each
            pmIndex must remain constant at least from one
            re-initialization of the entity's network management system
            to the next re-initialization.

            In case that no other MIB module has been chosen for
            providing entity identities, power state can be reported
            exclusively for the local device on which this table is
            instantiated. Then this table will have a single entry only
            and an index value of 0 MUST be used.

            The identity provisioning method that has been chosen can be
            retrived by reading the value of powerStateEnergyConsumerOid.
            In case of identities provided by the ENERGY AWARE MIB
            module, this OID points to an exising instance of pmIndex,
            in case of the ENTITY MIB, the object points to a valid
            instance of entPhysicalIndex, and in a similar way, it points
            to a value of another MIB module if this is used for
            identifying entities. If no other MIB module has been chosen
            for providing entity identities, then the value of
            powerStateEnergyConsumerOid MUST be 0.0 (zeroDotZero)."
        ::= { powerStateEntry 1 }

    powerStateEnergyConsumerOid OBJECT-TYPE
        SYNTAX      OBJECT IDENTIFIER
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An OID that identifies an entity that is subject of
            power state monitoring. The value MUST be an OID that
            points to an existing managed object or 0.0 (zeroDotZero).

            If another MIB module is chosen for providing identities
            for managed entities, then the value of this object
            points to an existing instance of an entity identifier,
            such as an instance of pmIndex in the ENERGY AWARE MIB or an
            instance of entPhysicalIndex in the ENTITY MIB module.

            If power state information is provided only for the local
            device on which this table is instantiated, then the value
            of this object MUST be 0.0 (zeroDotZero)."
        ::= { powerStateEntry 2 }
```

```
   powerStateOperationalState OBJECT-TYPE
       SYNTAX      SnmpAdminString (SIZE(1..32))
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the current power state of the
           entity. The given SnmpAdminString MUST match the
           powerStateName object of an enry in the
           powerStateAllStatesTable."
       ::= { powerStateEntry 3 }

   powerStateAdminState OBJECT-TYPE
       SYNTAX      SnmpAdminString (SIZE(0..32))
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
           "This object indicates the desired power state of the
           entity. This object may be set by a network management
           system in order to request changing the actual power state
           to the desired one.

           If this object has not been set by an administrative action
           requesting a certain power state, then its value is an
           empty string of length 0."
       ::= { powerStateEntry 4 }


   --------------------------------------------------------------------
   -- 1.2. All Power States Table
   --------------------------------------------------------------------
   powerStateAllStatesTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF PowerStateAllStatesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table provides information on all available power
           states of managed entities.

           The table extends the powerStateTable by sharing the first
           index. The first index serves for identifying an entity for
           which power state information is provided. The second index
           identifies a single power state by its name."
       ::= { powerStateObjects 2 }

   powerStateAllStatesEntry OBJECT-TYPE
       SYNTAX      PowerStateAllStatesEntry
       MAX-ACCESS  not-accessible
       STATUS      current
```

```
       DESCRIPTION
           "Power state information about this physical entity."
       INDEX       { powerStateEnergyConsumerId, powerStateName }
       ::= { powerStateAllStatesTable 1 }

   PowerStateAllStatesEntry ::=
       SEQUENCE {
          powerStateName               SnmpAdminString,
          powerStateType               INTEGER,
          powerStateDescription        SnmpAdminString,
          powerStateAveragePower       Integer32,
          powerStateMaximumPower       Integer32,
          powerStateTotalTime          TimeTicks,
          powerStateLastEnterTime      TimeStamp,
          powerStateLastEnterReason    SnmpAdminString,
          powerStateEnterCount         Counter64
       }

   powerStateName OBJECT-TYPE
       SYNTAX      SnmpAdminString (SIZE(1..32))
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This index should only be created for power states
           that are actually implemented by the entity that is
           identified by the first index powerStateEnergyConsumerOid.

           This index is the name of the power state and is limited
           to 32 characters.

           If possible the name SHOULD already give a rough idea of
           the characteristica of this power state."
       ::= { powerStateAllStatesEntry 1 }

   powerStateType OBJECT-TYPE
       SYNTAX      INTEGER {
                       unknown(0),
                       off(1),
                       nonOperational(2),
                       operational(3)
                   }
   --------------------------
   -- Open issue: Shall we replace the syntax by textual convention
   -- PowerMonitorLevel from draft-claise-energy-monitoring-mib?
   --------------------------
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
```

```
            "Object classifies the power state. It helps to clearly
            distinguish non-operational power states (sleep, standby,
            etc.) from operational ones. In a nonOperational(2) state
            an entity provides non of its primary services except
            for bringing it into operational(3) states or off(1)
            states.

            A device in state off(1) cannot report its state on its own.
            But state off(1) may be reported by managed devices
            reporting on the power state of other managed devices."
        ::= { powerStateAllStatesEntry 2 }

   powerStateDescription OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Power states are identified by their names. However,
            semantics of power states may vary between different
            entities. Reasons for variations can be different
            hardware and software architectures of managed devices.

            Object powerStateDescription SHOULD describe the power
            state and its characteristica as closely as possible."
        ::= { powerStateAllStatesEntry 3 }

   powerStateAveragePower OBJECT-TYPE
        SYNTAX      Integer32
        UNITS       "milliwatt"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the average power (energy consumption
            rate) in milliwatt at the electrical power supply of the
            entity in the power state indicated by powerStateName.

            A value of -1 indicates that the average power in this state
            is unknown."
        ::= { powerStateAllStatesEntry 4 }

   powerStateMaximumPower OBJECT-TYPE
        SYNTAX      Integer32
        UNITS       "milliwatt"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the maximum power (energy consumption
            rate) in milliwatt at the electrical power supply of the
```

```
           entity in the power state indicated by powerStateName.

           A value of -1 indicates that the maximum power in this state
           is unknown."
       ::= { powerStateAllStatesEntry 5 }

   powerStateTotalTime OBJECT-TYPE
       SYNTAX      TimeTicks
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total time in hundreds
           of seconds that the entity has been in the state
           indicated by index powerStateName."
       ::= { powerStateAllStatesEntry 6 }

   --------------------------
   -- Open issue: Shall we use DateAndTime instead of timeTicks?
   --------------------------
   powerStateLastEnterTime OBJECT-TYPE
       SYNTAX      TimeStamp
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This time stamp object indicates the last
           time a which the entity entered the state
           indicated by index powerStateName."
       ::= { powerStateAllStatesEntry 7 }

   powerStateLastEnterReason OBJECT-TYPE
       SYNTAX      SnmpAdminString
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This string object describes the reason for the last
           power state transition into the power state
           indicated by index powerStateName."
       ::= { powerStateAllStatesEntry 8 }

   powerStateEnterCount OBJECT-TYPE
       SYNTAX      Counter64
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates how often the entity
           indicated by index entPhysicalIndex entered the
           power state indicated by index powerStateName."
       ::= { powerStateAllStatesEntry 9 }
```

```
   --==================================================================
   -- 2. Notifications
   --==================================================================

   powerStateChangeEvent NOTIFICATION-TYPE
       OBJECTS     { powerStateLastEnterReason }
       STATUS      current
       DESCRIPTION
           "This notification can be generated when the power state of
           an entity changes.

           Note that the state that has been entered is indicated by
           the OID of object powerStateLastEnterReason."
       ::= { powerStateNotifications 1 }


   --==================================================================
   -- 3. Conformance Information
   --==================================================================

   powerStateCompliances OBJECT IDENTIFIER
       ::= { powerStateConformance 1 }
   powerStateGroups      OBJECT IDENTIFIER
       ::= { powerStateConformance 2 }

   -------------------------------------------------------------------
   -- 3.1. Compliance Statements
   -------------------------------------------------------------------

   powerCompliance MODULE-COMPLIANCE
       STATUS      current
       DESCRIPTION
           "The compliance statement for implementations of the
           POWER-STATE-MIB module.

           A compliant implementation MUST implement the objects
           defined in the mandatory group powerRequiredGroup."
       MODULE  -- this module
       MANDATORY-GROUPS { powerStateRequiredGroup }
       GROUP   powerStateNotificationsGroup
       DESCRIPTION
          "A compliant implementation does not have to implement
           the powerNotificationsGroup."
       ::= { powerStateCompliances 1 }

   -------------------------------------------------------------------
   -- 3.2. MIB Grouping
   -------------------------------------------------------------------
```

```
   powerStateRequiredGroup OBJECT-GROUP
       OBJECTS {
          powerStateOperationalState,
          powerStateAdminState,
          powerStateType,
          powerStateDescription,
          powerStateTotalTime,
          powerStateLastEnterTime,
          powerStateLastEnterReason,
          powerStateEnterCount,
          powerStateAveragePower,
          powerStateMaximumPower
       }
       STATUS      current
       DESCRIPTION
           "A compliant implementation MUST implement the objects
           contained in this group."
       ::= { powerStateGroups 1 }

   powerStateNotificationsGroup NOTIFICATION-GROUP
       NOTIFICATIONS { powerStateChangeEvent }
       STATUS      current
       DESCRIPTION
           "A compliant implementation does not have to implement the
           notification contained in this group."
       ::= { powerStateGroups 2 }
   END
```

8.2. Energy MIB

```
   ENERGY-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, mib-2,
       Unsigned32, Integer32
           FROM SNMPv2-SMI                                 -- RFC2578
       TimeStamp
           FROM SNMPv2-TC                                  -- RFC2579
       MODULE-COMPLIANCE, OBJECT-GROUP
           FROM SNMPv2-CONF                                -- RFC2580
       EntitySensorStatus
           FROM ENTITY-SENSOR-MIB                          -- RFC3433
       powerStateName
           FROM POWER-STATE-MIB
       UnitMultiplier
           FROM POWER-MONITOR-MIB;

   energyMIB MODULE-IDENTITY
```

```
       LAST-UPDATED "201010231200Z"         -- 23 October 2010
       ORGANIZATION "IETF OPSAWG Working Group"
       CONTACT-INFO
           "General Discussion: opsawg@ietf.org
           To Subscribe: https://www.ietf.org/mailman/listinfo/opsawg
           Archive: http://www.ietf.org/mail-archive/web/opsawg

           Co-editor:
             Juergen Quittek
             NEC Europe Ltd.
             NEC Laboratories Europe
             Kurfuersten-Anlage 36
             69115 Heidelberg
             Germany
             Tel: +49 6221 4342-115
             Email: quittek@neclab.eu

           Co-editor:
             Thomas Dietz
             NEC Europe Ltd.
             NEC Laboratories Europe
             Kurfuersten-Anlage 36
             69115 Heidelberg
             Germany
             Phone: +49 6221 4342-128
             Email: Thomas.Dietz@neclab.eu"

       DESCRIPTION
           "This MIB module defines a set of objects for monitoring
           the energy consumption of networked devices and their
           components.

           Copyright (c) 2010 IETF Trust and the persons identified as
           authors of the code.  All rights reserved.

           Redistribution and use in source and binary forms, with or
           without modification, is permitted pursuant to, and subject
           to the license terms contained in, the Simplified BSD License
           set forth in Section 4.c of the IETF Trust's Legal Provisions
           Relating to IETF Documents
           (http://trustee.ietf.org/license-info).

           This version of this MIB module is part of RFC yyyy; see
           the RFC itself for full legal notices."
    -- replace yyyy with actual RFC number & remove this notice

    --  Revision history
```

```
     REVISION     "201010231200Z"         -- 23 October 2010
     DESCRIPTION
         "Initial version, published as RFC yyyy."
 -- replace yyyy with actual RFC number & remove this notice

     ::= { mib-2 9992 }
 -- yyy to be assigned by IANA.

 --******************************************************************
 -- Top Level Structure of the MIB module
 --******************************************************************

 energyObjects       OBJECT IDENTIFIER ::= { energyMIB 1 }
 energyConformance   OBJECT IDENTIFIER ::= { energyMIB 2 }

 --==================================================================
 -- 1. Object Definitions
 --==================================================================

 --------------------------------------------------------------------
 -- 1.1. Energy Consumption Table
 --------------------------------------------------------------------
 energyTable  OBJECT-TYPE
     SYNTAX      SEQUENCE OF EnergyEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "This table provides inforamtion on the current and
         accumulated energy consumption of entities.

         The table is indexed by an ID of the entity on which
         energy information is provided. IDs can be provided by
         another MIB module, such as the ENERGY AWARE MIB module
         or the ENTITY MIB module. If not ID provisioning from
         other MIB modules is available, the table can only have
         one entry for reporting the local power state of the
         device that tuns an instance of this table."
     ::= { energyObjects 1 }

 energyEntry OBJECT-TYPE
     SYNTAX      EnergyEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "An entry providing information on the energy consumption
         of a physical entity."
     INDEX  { energyConsumerId }
     ::= { energyTable 1 }
```

```
   EnergyEntry ::=
       SEQUENCE {
          energyConsumerId             Integer32,
          energyConsumerOid            OBJECT IDENTIFIER,
          energySensorOperStatus       EntitySensorStatus,
          energyNominalSupplyVoltage   Unsigned32,
          energyElectricSupplyType     INTEGER,
          energyTotalEnergy            Unsigned32,
          energyEnergyUnitMultiplier   UnitMultiplier,
          energyEnergyPrecision        Integer32,
          energyMeasurementMethod      INTEGER,
          energyDiscontinuityTime      TimeStamp,
          energySampleInterval         Unsigned32,
          energyMaxHistory             Unsigned32,
          energyPowerUnitMultiplier    UnitMultiplier,
          energyPowerPrecision         Integer32,
          energyRealPower              Unsigned32,
          energyPeakRealPower          Unsigned32,
          energyReactivePower          Unsigned32,
          energyApparentPower          Unsigned32,
          energyPhaseAngle             Integer32,
          energyPhaseAnglePrecision    Integer32
       }

   energyConsumerId OBJECT-TYPE
       SYNTAX      Integer32 (0..2147483647)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An integer that identifies an entity that is subject of
           energy monitoring. Index values MUST be locally unique
           for each identified entity.

           If an implementation of the ENERGY AWARE MIB module is
           available in the local SNMP context, then the same index of
           an entity MUST be chosen as assigned to the entity by object
           pmIndex in the ENERGY AWARE MIB module. In this case,
           entities without an assigned value for pmIndex cannot be
           indexted by the powerCurrentStateTable.

           If there is no implementation of the ENERGY AWARE MIB module
           but one of the ENTITY MIB module is available in the local
           SNMP context, then the same index of an entity MUST be
           chosen as assigned to the entity by object entPhysicalIndex
           in the ENTITY MIB module. In this case, entities without an
           assigned value for pmIndex cannot be indexted by the
           powerCurrentStateTable.
```

```
            If neither the ENERGY AWARE MIB module nor of the ENTITY MIB
            module is available in the local SNMP context, then this MIB
            module may choose identity values from a further MIB module
            providing entity identities. In this case the value for each
            pmIndex must remain constant at least from one
            re-initialization of the entity's network management system
            to the next re-initialization.

            In case that no other MIB module has been chosen for
            providing entity identities, power state can be reported
            exclusively for the local device on which this table is
            instantiated. Then this table will have a single entry only
            and an index value of 0 MUST be used.

            The identity provisioning method that has been chosen can be
            retrived by reading the value of object
            powerStateEnergyConsumerOid. In case of identities provided
            by the ENERGY AWARE MIB module, this OID points to an
            exising instance of pmIndex, in case of the ENTITY MIB, the
            object points to a valid instance of entPhysicalIndex, and
            in a similar way, it points to a value of another MIB module
            if this is used for identifying entities. If no other MIB
            module has been chosen for providing entity identities, then
            the value of powerStateEnergyConsumerOid MUST be 0.0
            (zeroDotZero)."
        ::= { energyEntry 1 }

   energyConsumerOid OBJECT-TYPE
        SYNTAX      OBJECT IDENTIFIER
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An OID that identifies an entity that is subject of energy
            monitoring. The value MUST be an OID that points to an
            existing managed object or 0.0 (zeroDotZero).

            If another MIB module is chosen for providing identities
            for managed entities, then the value of this object
            points to an existing instance of an entity identifier,
            such as an instance of pmIndex in the ENERGY AWARE MIB or an
            instance of entPhysicalIndex in the ENTITY MIB module.

            If power state information is provided only for the local
            device on which this table is instantiated, then the value
            of this object MUST be 0.0 (zeroDotZero)."
        ::= { energyEntry 2 }

   energySensorOperStatus OBJECT-TYPE
```

```
       SYNTAX      EntitySensorStatus
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the operational status of the
           sensor that is used for measuring the energy consumption
           of the entity indicated by energyConsumerId."
       ::= { energyEntry 3 }

   energyNominalSupplyVoltage OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "millivolt"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the nominal voltage of the power
           supply of the entity. It is provided in units of
           millivolt (mV).

           The nominal voltage actual of an entity is assumed to be
           fixed, while the actual power supply voltage may vary over
           time, for example, caused by changing load conditions.

           A value of 0 indicates that the nominal supply voltage
           is unknown."
       ::= { energyEntry 4 }

   energyElectricSupplyType OBJECT-TYPE
       SYNTAX      INTEGER {
                       alternatingCurrent(1),
                       directCurrent(2),
                       unknown(3)
                   }
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the type of electrical power
           supply for the entity. It is used for distinguishing
           between alternating current (AC) supply and direct
           current (DC) supply."
       ::= { energyEntry 5 }

   energyTotalEnergy OBJECT-TYPE
       SYNTAX      Unsigned32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total consumed energy measured
```

```
           at the electrical power supply of the entity.

           In order to determine the measured value in watt hours,
           the value of this object needs to be multiplied by a unit
           mulitplier given by the value of object
           energyEnergyUnitMultiplier.

           Discontinuities in the value of this counter can occur at
           re-initialization of the management system, and at other
           times as indicated by the value of
           energyDiscontinuityTime."
       ::= { energyEntry 6 }

   energyEnergyUnitMultiplier OBJECT-TYPE
       SYNTAX      UnitMultiplier
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides unit ultiplier for measured energy
           values. Reported values need to be multiplied with this
           multiplier in order to determine the measured value in
           watt hours.

           This object serves as unit multiplier for objects
           energyTotalEnergy, energyPSTotalEnergy,
           ..."
       ::= { energyEntry 7 }

   energyEnergyPrecision OBJECT-TYPE
       SYNTAX      Integer32 (0..10000)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates a the precision of a measured energy
           value. The precision is indicated as a percentage value,
           in 100ths of a percent. A value of 0 indicates that the
           precision is unknown or not applicable to the measured
           value.

           This object serves precision indicator for the values
           provided by objects energyTotalEnergy,
           energyPSTotalEnergy, ..."
       ::= { energyEntry 8 }

   energyMeasurementMethod OBJECT-TYPE
       SYNTAX      INTEGER {
                       directEnergyMeasurement(1),
                       powerOversampling(2),
```

```
                        powerSampling(3),
                        loadBasedEstimation(4),
                        deviceBasedEstimation(5),
                        unknown(6)
                    }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the method used for measuring energy
            consumption.  A device may not be equipped with capabilities
            to measure its energy consumption directly, but rather
            relies on other input in order to conduct more or less
            precise estimations of its power consumption.

            The measurement methods concerns values of objects
            energyTotalEnergy, energyPSTotalEnergy, and
            energyPowerHistoryAverageValue.

            Five different measurement methods are specified.

            - directEnergyMeasurement(1) indicates that the entity is
              instrumented to directly measure its energy consumption.

            - powerOversampling(2) indicates that energy is measured
              by sampling power values more frequently than indicated
              by the value of object energySampleInterval.

            - powerSampling(3) indicates that energy is measured
              by sampling power values according to the value of object
              energySampleInterval.

            - loadBasedEstimation(4) indicates that power is estimated
              based on measurements of the load of the entity.

            - deviceBasedEstimation(5) indicates that power is estimated
              based on static properties of the entity. In this case,
              reported power only depednds on the pwoer state of the
              devices as indicated by object powerCurrentState in the
              powerCurrentStateTable of the Power State MIB module."
        ::= { energyEntry 9 }

    energyDiscontinuityTime OBJECT-TYPE
        SYNTAX      TimeStamp
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of sysUpTime on the most recent occasion at which
            any one or more of this entity's energy consumption counters
```

```
          suffered a discontinuity.  The relevant counters are
          energyTotalEnergy and energyPerStateTotalEnergy.  If
          no such discontinuities have occurred since the last re-
          initialization of the local management subsystem, then this
          object contains a zero value."
      ::= { energyEntry 10 }

   energySampleInterval OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates is the difference of time
           stamps between two consecutive entries in the
           energyHistoryTable for this entity.

           The interval lenght provided by this object indicates the
           or maximum interval length (or minimal samping rate) at which
           the power sensor measures values of the current power.
           Implementations of the Energy MIB module may choose higher
           sampling rates (or shorter sampling intervals) in order to
           provide higher precision of the measurement. Preferably,
           shorter intervals may be chosen such that the sampling
           interval indicated by this object is a multiple of the actual
           sampling interval.

           The sampling interval is provided in units of microseconds.

           A value of 0 indicates that the sampling interval applied by
           the sensor is unknown or not constant."
       ::= { energyEntry 11 }

   energyMaxHistory OBJECT-TYPE
       SYNTAX      Unsigned32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates is the maximum number of
           corresponding entries in the energyPowerHistoryTable.
           An entry in the energyHistoryTable is corresponding
           if it has the same value for object energyConsumerId
           as index.

           An implementation of the Energy MIB module will remove the
           oldest correaponding entry in the energyHistoryTable
           to allow the addition of a new entry once the number of
           corresponding entries in the energyHistoryTable
```

```
            reaches this value.

            Entries are added to the energyHistoryTable until
            energyMaxHistory is reached before entries
            begin to be removed.

            A value of 0 for this object disables creation of
            corresponding energyHistoryTable entries."
        DEFVAL      { 0 }
        ::= { energyEntry 12 }

    energyPowerUnitMultiplier OBJECT-TYPE
        SYNTAX      UnitMultiplier
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides unit ultiplier for measured energy
            values. Reported values need to be multiplied with this
            multiplier in order to determine the measured value in
            watt hours.

            This object serves as unit multiplier for the values provided
            by objects energyRealPower, energyPeakRealPower,
            energyReactivePower, and energyApparentPower."
        ::= { energyEntry 13 }

    energyPowerPrecision OBJECT-TYPE
        SYNTAX      Integer32 (0..10000)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates a the precision of a measured power
            value. The precision is indicated as a percentage value,
            in 100ths of a percent. A value of 0 indicates that the
            precision is unknown or not applicable to the measured
            value.

            This object serves precision indicator for the values
            provided by objects energyRealPower, energyPeakRealPower,
            energyReactivePower, and energyApparentPower."
        ::= { energyEntry 14 }

    energyRealPower OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the current real power value
```

```
            at the electrical supply of the entity indicated by index
            energyConsumerId.

            In order to determine the measured value in watts,
            the value of this object needs to be multiplied by a unit
            mulitplier given by the value of object
            energyEnergyUnitMultiplier.

            Measured values of this object are stored in the
            energyPowerTable with a rate determined by object
            energySampleInterval."
        ::= { energyEntry 15 }

   energyPeakRealPower OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the highest observed value for
            object energyRealPower since the last
            re-initialization of the management system.

            In order to determine the measured value in watts,
            the value of this object needs to be multiplied by a unit
            mulitplier given by the value of object
            energyEnergyUnitMultiplier."
        ::= { energyEntry 16 }

   energyReactivePower OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "volt-amperes reactive"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the current reactive power value
            at the electrical supply of the entity indicated by index
            energyConsumerId.

            In order to determine the measured value in volt-amperes
            (var), the value of this object needs to be multiplied by
            a unit mulitplier given by the value of object
            energyEnergyUnitMultiplier.

            The value provided by this object is only useful if the
            value of object energySupplyType is
            alternatingCurrent(1). In this case it is RECOMMENDED that
            at least one of the three values energyReactivePower,
            energyApparentPowerScale, and energyPhaseAngle
```

```
            are provided.

            If object energyElectricSupplyType of this row has a
            value other than alternatingCurrent(1), then the value of
            this object MUST be 0.

            If object energyElectricSupplyType of this row has the
            value alternatingCurrent(1) and if no value for the current
            reactive power is provided, then the value of this object
            MUST be 0xFFFF."
        ::= { energyEntry 17 }

    energyApparentPower OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "volt-amperes"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the current apparent power value
            measured in volt-ampere (VA) at the electrical supply of the
            entity for a time interval indicated by object
            energySampleInterval.

            The value provided by this object is only useful if the
            value of object energySupplyType is
            alternatingCurrent(1). In this case it is RECOMMENDED that
            at least one of the three values energyReactivePower,
            energyApparentPowerScale, and energyPhaseAngle
            are provided.

            Scale and precision of the value are indicated by objects
            energyPowerScale and energyPowerPrecision.

            If object energyElectricSupplyType of this row has a
            value other than alternatingCurrent(1), then the value of
            this object MUST be equal to the value of object
            energyRealPower.

            If object energyElectricSupplyType of this row has the
            value alternatingCurrent(1) and if no value for the current
            apparent power is provided, then the value of this object
            MUST be -10000000000."
        ::= { energyEntry 18 }

    energyPhaseAngle OBJECT-TYPE
        SYNTAX Integer32 (-1..360000)
        UNITS       "millidegrees"
        MAX-ACCESS  read-only
```

```
       STATUS      current
       DESCRIPTION
           "This object indicates the current phase angle value
           measured at the electrical supply of the entity for a time
           interval indicated by object energySampleInterval.

           The value provided by this object is only useful if the
           value of object energySupplyType is
           alternatingCurrent(1). In this case it is RECOMMENDED that
           at least one of the three values energyReactivePower,
           energyApparentPowerScale, and energyPhaseAngle
           are provided.

           The value is provided in units of millidegree (one thousands
           of a degree.  This is equivalent to an assiciated object
           of type EntitySensorDataScale with the value of milli(8) and
           an associated object of type EntitySensorPrecision with a
           value of 0.

           The minimum value for this object when indicating an actual
           angle is 0, the maximum value is 360000.

           The maximum error of of the value is indicated by object
           energyPhaseAngleMaxError.

           If object energyElectricSupplyType of this row has a
           value other than alternatingCurrent(1), then the value of
           this object MUST be 0.

           If object energyElectricSupplyType of this row has the
           value alternatingCurrent(1) and if no value for the phase
           angle is provided, then the value of this object
           MUST be -1."
       ::= { energyEntry 19 }

   energyPhaseAnglePrecision OBJECT-TYPE
       SYNTAX      Integer32 (0..10000)
       UNITS       "millidegrees"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates a the precision of a measured phase
           angle value. The precision is indicated as a percentage
           value, in 100ths of a percent. A value of 0 indicates that
           the precision is unknown or not applicable to the measured
           value.

           This object serves precision indicator for the values
```

```
            provided by object energyPhaseAngle."
        ::= { energyEntry 20 }


   -------------------------------------------------------------------
   -- 1.2. Energy Consumption Per Power State Table
   -------------------------------------------------------------------
   energyPerStateTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF EnergyPerStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table provides information on the accumulated energy
           consumption of an entity.

           This table extends the energyTable by sharing the
           first index. The first index serves for identifying an
           entity for which energy information is provided. The second
           index identifies a single power state by its name."
       ::= { energyObjects 2 }

   energyPerStateEntry OBJECT-TYPE
       SYNTAX      EnergyPerStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Energy consumption information per power state for a
           physical entity."
       INDEX  { energyConsumerId, powerStateName }
       ::= { energyPerStateTable 1 }

   EnergyPerStateEntry ::=
       SEQUENCE {
          energyPerStateTotalEnergy          Unsigned32
       }

   energyPerStateTotalEnergy OBJECT-TYPE
       SYNTAX      Unsigned32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total consumed energy value
           at the electrical supply of the entity indicated by index
           energyConsumerId while being in a specific power state
           indicated by index powerStateName.

           In order to determine the measured value in watts, the value
           of this object needs to be multiplied by a unit mulitplier
```

```
            given by the value of object
            energyEnergyUnitMultiplier of table
            energyTable.

            Discontinuities in the value of this counter can occur at
            re-initialization of the management system, and at other
            times as indicated by the value of
            energyDiscontinuityTime."
        ::= { energyPerStateEntry 1 }


    -------------------------------------------------------------------
    -- 1.3. Energy Power History Table
    -------------------------------------------------------------------
    energyHistoryTable  OBJECT-TYPE
        SYNTAX      SEQUENCE OF EnergyHistoryEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table stores results of energy consumption
            measurements for multiple entities.

            This table extends the energyTable by sharing the
            first index. The first index serves for identifying an
            entity for which energy information is provided. The second
            index energyHistoryIndex identifies a single measurement
            consiting of an energy consumptiopn value and a timestamp.

            Creation of entries in this row is controlled indivdually
            for each entity by two parameters: energyMaxHistory and
            energySamplingInterval.

            The energySamplingInterval controls the difference in time
            between the creation of two consecutive entries in this
            table. Object energyMaxHistory limits the number of entries
            in this table that can be created for the corresponding
            entity.

            An implementation of the Energy MIB module will remove the
            oldest entry for an entity in the energyHistoryTable to
            allow the addition of a new entry once the number of
            entries for this entity reaches the value indicated by
            object energyMaxHistory.

            Entries for a specific entity are added to this table
            until energyMaxHistory is reached before
            entries begin to be removed.
```

```
           Entries for the same entity are indexed by
           energyHistoryIndex. The first entry for an entity MUST have
           an index value of 1. Further new entries MUST be indexed by
           consecutive numbers in the order in which they are created
           until the value of energyMaxHistory is reached. Then no
           further new indices will be assigned, but existing ones will
           be re-used."
       ::= { energyObjects 3 }

   energyHistoryEntry OBJECT-TYPE
       SYNTAX      EnergyHistoryEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An entry indicating consumed energy for an entity
           at a certain point in time."
       INDEX  { energyConsumerId, energyHistoryIndex }
       ::= { energyHistoryTable 1 }

   EnergyHistoryEntry ::=
       SEQUENCE {
          energyHistoryIndex          Unsigned32,
          energyHistoryTimestamp      TimeStamp,
          energyHistoryTotalEnergy    Unsigned32
       }

   energyHistoryIndex OBJECT-TYPE
       SYNTAX      Unsigned32 (1..4294967295)
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
               "The index for this entry per entity.
               Values of this index MUST be unique per entity used
               as first index."
       ::= { energyHistoryEntry 1 }

   energyHistoryTimestamp OBJECT-TYPE
       SYNTAX      TimeStamp
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the time at which the
           energy consumption value provided by object
           energyHistoryTotalEnergy was measured."
       ::= { energyHistoryEntry 2 }

   energyHistoryTotalEnergy OBJECT-TYPE
       SYNTAX      Unsigned32
```

```
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total consumed energy measured
           at the electrical power supply of the entity.

           In order to determine the measured value in watt hours,
           the value of this object needs to be multiplied by a unit
           mulitplier given by the value of object
           energyEnergyUnitMultiplier in the corresponding entry
           for this entity in table energyTable.

           Discontinuities in the value of this counter can occur at
           re-initialization of the management system, and at other
           times as indicated by the value of
           energyDiscontinuityTime in the corresponding entry
           for this entity in table energyTable."
       ::= { energyHistoryEntry 3 }


   --==================================================================
   -- 2. Conformance Information
   --==================================================================

   energyCompliances OBJECT IDENTIFIER ::= { energyConformance 1 }
   energyGroups      OBJECT IDENTIFIER ::= { energyConformance 2 }

   --------------------------------------------------------------------
   -- 2.1. Compliance Statements
   --------------------------------------------------------------------

   energyCompliance MODULE-COMPLIANCE
       STATUS      current
       DESCRIPTION
           "The compliance statement for implementations of the
           ENERGY-MIB module.

           A compliant implementation MUST implement the objects
           defined in the mandatory group energyRequiredGroup.

           If one of the entities for which energy consumption is
           reported are supplied by alternating current (AC) then it
           is recommended that not just real power is reported
           (REQUIRED) but it is also RECOMMENDED that at least one
           of three other related values (reactive power, apparent
           power, and phase angle) is reported by implementing at least
           one of the three groups energyReactivePowerGroup,
           energyApparentPowerGroup, and energyPhaseAngleGroup."
```

```
        MODULE  -- this module
        MANDATORY-GROUPS { energyRequiredGroup }

        GROUP energyPowerHistoryGroup
        DESCRIPTION
            "This group is only needed for implementations that
            support storing time series of measured power values
            in the energyPowerHistoryTable."

        GROUP energyACGroup
        DESCRIPTION
            "This group is only needed for implementations that report
            consumption of electric energy provided by alternating
            current (AC) supply.

            Implementations for devices supplied with direct current (DC)
            only and implementations that do only report real power
            reporting for alternative current do not need to implement
            objects in this group."

        GROUP energyReactivePowerGroup
        DESCRIPTION
            "Information provided by elements in this group is redundant
            to information provided by elements in the
            energyApparentPowerGroup and the energyPhaseAngleGroup.

            For compliant implementations that report consumption of
            electric energy provided by alternating current (AC) supply
            it is RECOMMENDED to at least one of the three groups
            energyReactivePowerGroup, energyApparentPowerGroup, and
            energyPhaseAngleGroup."

        GROUP energyApparentPowerGroup
        DESCRIPTION
            "Information provided by elements in this group is redundant
            to information provided by elements in the
            energyReactivePowerGroup and the energyPhaseAngleGroup.

            For compliant implementations that report consumption of
            electric energy provided by alternating current (AC) supply
            it is RECOMMENDED to at least one of the three groups
            energyReactivePowerGroup, energyApparentPowerGroup, and
            energyPhaseAngleGroup."

        GROUP energyPhaseAngleGroup
        DESCRIPTION
            "Information provided by elements in this group is redundant
            to information provided by elements in the
```

```
        energyReactivePowerGroup and the energyApparentPowerGroup.

        For compliant implementations that report consumption of
        electric energy provided by alternating current (AC) supply
        it is RECOMMENDED to at least one of the three groups
        energyReactivePowerGroup, energyApparentPowerGroup, and
        energyPhaseAngleGroup."

    ::= { energyCompliances 1 }

--------------------------------------------------------------------
-- 2.2. Object Grouping
--------------------------------------------------------------------

energyRequiredGroup OBJECT-GROUP
    OBJECTS {
       energySensorOperStatus,
       energyNominalSupplyVoltage,
       energyElectricSupplyType,
       energyTotalEnergy,
       energyEnergyUnitMultiplier,
       energyEnergyPrecision,
       energyMeasurementMethod,
       energyDiscontinuityTime,
       energyPowerUnitMultiplier,
       energyPowerPrecision,
       energyRealPower,
       energyPeakRealPower,
       energyPerStateTotalEnergy
    }
    STATUS      current
    DESCRIPTION
        "A compliant implementation MUST implement the objects
        contained in this group."
    ::= { energyGroups 1 }

energyPowerHistoryGroup OBJECT-GROUP
    OBJECTS {
       energySampleInterval,
       energyMaxHistory,
       energyHistoryTimestamp,
       energyHistoryTotalEnergy
    }
    STATUS      current
    DESCRIPTION
        "The group of object for reporting details of
        AC power measurement."
    ::= { energyGroups 2 }
```

```
   energyACGroup OBJECT-GROUP
       OBJECTS {
          energyReactivePower,
          energyApparentPower,
          energyPhaseAngle,
          energyPhaseAnglePrecision
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting details of
           AC power measurement."
       ::= { energyGroups 3 }

   energyReactivePowerGroup OBJECT-GROUP
       OBJECTS {
          energyReactivePower
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the reactive power
           measured for AC supply."
       ::= { energyGroups 4 }

   energyApparentPowerGroup OBJECT-GROUP
       OBJECTS {
          energyApparentPower
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the apparent power
           measured for AC supply."
       ::= { energyGroups 5 }

   energyPhaseAngleGroup OBJECT-GROUP
       OBJECTS {
          energyPhaseAngle,
          energyPhaseAnglePrecision
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the phase angler
           measured for AC supply."
       ::= { energyGroups 6 }

   END
```

8.3. Battery MIB

```
BATTERY-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    mib-2, Integer32, Unsigned32, Counter32
        FROM SNMPv2-SMI                                -- RFC2578
    DateAndTime
        FROM SNMPv2-TC                                 -- RFC2579
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF                               -- RFC2580
    entPhysicalIndex
        FROM ENTITY-MIB;                                -- RFC4133

batteryMIB MODULE-IDENTITY
    LAST-UPDATED "201001291200Z"         -- 29 January 2010
    ORGANIZATION "IETF OPSAWG Working Group"
    CONTACT-INFO
        "General Discussion: opsawg@ietf.org
        To Subscribe: https://www.ietf.org/mailman/listinfo/opsawg
        Archive: http://www.ietf.org/mail-archive/web/opsawg

        Co-editor:
          Juergen Quittek
          NEC Europe Ltd.
          NEC Laboratories Europe
          Kurfuersten-Anlage 36
          69115 Heidelberg
          Germany
          Tel: +49 6221 4342-115
          Email: quittek@neclab.eu

        Co-editor:
          Thomas Dietz
          NEC Europe Ltd.
          NEC Laboratories Europe
          Kurfuersten-Anlage 36
          69115 Heidelberg
          Germany
          Phone: +49 6221 4342-128
          Email: Thomas.Dietz@neclab.eu"

    DESCRIPTION
        "This MIB module defines a set of objects for monitoring
        batteries of networked devices and of their components.

        Copyright (c) 2010 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.
```

```
        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this MIB module is part of RFC yyyy; see
        the RFC itself for full legal notices."
-- replace yyyy with actual RFC number & remove this notice

--  Revision history

    REVISION     "201001291200Z"         -- 29 January 2010
    DESCRIPTION
        "Initial version, published as RFC yyyy."
-- replace yyyy with actual RFC number & remove this notice

    ::= { mib-2 zzz }
-- zzz to be assigned by IANA.

--******************************************************************
-- Top Level Structure of the MIB module
--******************************************************************

batteryNotifications OBJECT IDENTIFIER ::= { batteryMIB 0 }
batteryObjects       OBJECT IDENTIFIER ::= { batteryMIB 1 }
batteryConformance   OBJECT IDENTIFIER ::= { batteryMIB 2 }

--==================================================================
-- 1. Object Definitions
--==================================================================

--------------------------------------------------------------------
-- 1.1. Battery Table
--------------------------------------------------------------------
batteryTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF BatteryEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides information on batteries in networked
        devices. It is designed as a sparse augment of the
        entPhysicalTable defined in the ENTITY-MIB module and assumes
        that each battery is represented by an individual row in the
        entPhysicalTable with an individual value for the index
        entPhysicalIndex.
```

```
           Entries appear in this table only for entities that represent
           a battery.  An entry in this table SHOULD be created at the
           same time as the associated entPhysicalEntry.  An entry
           SHOULD be destroyed if the associated entPhysicalEntry is
           destroyed."
       ::= { batteryObjects 1 }

   batteryEntry OBJECT-TYPE
       SYNTAX      BatteryEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An entry providing information on a battery."
       INDEX  { entPhysicalIndex }    -- SPARSE-AUGMENTS
       ::= { batteryTable 1 }

   BatteryEntry ::=
       SEQUENCE {
          batteryType                     INTEGER,
          batteryTechnology               INTEGER,
          batteryNominalVoltage           Unsigned32,
          batteryNumberOfCells            Unsigned32,
          batteryNominalCapacity          Unsigned32,
          batteryRemainingCapacity        Unsigned32,
          batteryChargingCycleCount       Counter32,
          batteryLastChargingCycleTime   DateAndTime,
          batteryState                    INTEGER,
          batteryCurrentCharge            Unsigned32,
          batteryCurrentChargePercentage  Unsigned32,
          batteryCurrentVoltage           Unsigned32,
          batteryCurrentCurrent           Integer32,
          batteryLowAlarmPercentage       Unsigned32,
          batteryLowAlarmVoltage          Unsigned32,
          batteryReplacementAlarmCapacity Unsigned32,
          batteryReplacementAlarmCycles   Unsigned32
       }

   batteryType OBJECT-TYPE
       SYNTAX      INTEGER {
                       primary(1),
                       rechargeable(2),
                       capacitor(3),
                       other(4),
                       unknown(5)
                   }
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
```

```
            "This object indicates the type of battery.  It distinguishes
            between one-way primary batteries, rechargeable secondary
            batteries and capacitors which are not really batteries but
            often used in the same way as a battery.

            The value other(4) can be used if the battery type is known
            but none of the ones above.  Value unknown(5) is to be used
            if the type of battery cannot be determined."
        ::= { batteryEntry 1 }

    batteryTechnology OBJECT-TYPE
        SYNTAX      INTEGER {
                        zincCarbon(1),
                        zincChloride(2),
                        oxyNickelHydroxide(3),
                        lithiumCopper(4),
                        lithiumIron(5),
                        lithiumManganese(6),
                        zincAir(7),
                        silverOxide(8),
                        alcaline(9),
                        leadAcid(10),
                        nickelCadmium(12),
                        nickelMetalHybride(13),
                        nickelZinc(14),
                        lithiumIon(15),
                        lithiumPolymer(16),
                        doubleLayerCapacitor(17),
                        other(18),
                        unknown(19)
                    }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the technology used by the battery.
            Values 1-8 are primary battery technologies, values 10-16
            are rechargeable battery technologies and value alkaline(9)
            is used for primary batteries as well as for rechargeable
            batteries.

            The value other(18) can be used if the battery type is known
            but none of the ones above.  Value unknown(19) is to be used
            if the type of battery cannot be determined."
        ::= { batteryEntry 2 }

    batteryNominalVoltage OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "millivolt"
```

```
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the nominal voltage of the battery
           in units of millivolt (mV).

           Note that the nominal voltage is a constant value and
           typically different from the actual voltage of the battery.

           A value of 0 indicates that the nominal voltage is unknown."
       ::= { batteryEntry 3 }

   batteryNumberOfCells OBJECT-TYPE
       SYNTAX      Unsigned32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the number of cells contained in the
           battery.

           A value of 0 indicates that the number of cells is unknown."
       ::= { batteryEntry 4 }

   batteryNominalCapacity OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliampere hours"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the nominal capacity of the battery
           in units of milliampere hours (mAh).

           Note that the nominal capacity is a constant value and
           typically different from the actual capacity of the battery.

           A value of 0 indicates that the nominal capacity is unknown."
       ::= { batteryEntry 5 }

   batteryRemainingCapacity OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliampere hours"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the ACTUAL REMAINING capacity of the
           battery in units of milliampere hours (mAh).

           Note that the actual capacity needs to be measured and is
```

```
            typically an estimate based on observed discharging and
            charging cycles of the battery.

            A value of 'ffffffff'H indicates that the actual capacity
            cannot be determined."
        ::= { batteryEntry 6 }

    batteryChargingCycleCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object indicates the number of charging cycles that
            that the battery underwent. Please note that the precise
            definition of a rechsarge cycle varies for different kinds
            of batteries and of devices containing batteries.

            For batteries of type primary(1) the value of this object is
            always 0.

            A value of 'ffffffff'H indicates that the number of charging
            cycles cannot be determined."
        ::= { batteryEntry 7 }

    batteryLastChargingCycleTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The date and time of the last charging cycle.  The value
            '0000000000000000'H is returned if the battery has not been
            charged yet or if the last charging time cannot be
            determined.

            For batteries of type primary(1) the value of this object is
            always '0000000000000000'H."
        ::= { batteryEntry 8 }

    batteryState OBJECT-TYPE
        SYNTAX      INTEGER {
                        full(1),
                        partiallyCharged(2),
                        empty(3),
                        charging(4),
                        discharging(5),
                        unknown(6)
                    }
        MAX-ACCESS  read-only
```

```
        STATUS      current
        DESCRIPTION
            "This object indicates the current state of the battery.
            Value full(1) indicates a full battery with a capacity
            given by onject batteryRemainingCapacity.  Value empty(3)
            indicates a battery that cannot be used for providing
            electric power before charging it.  Value partiallyCharged(2)
            is provided if the battery is neither empty nor full and if
            no charging or discharging is in progress.  Charging or
            discharging of hte battery is indicated by values charging(3)
            or discharging(4), respectively.

            Value unknown(6) is to be used if the state of the battery
            cannot be determined."
        ::= { batteryEntry 9 }

    batteryCurrentCharge OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "milliampere hours"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the current charge of the battery
            in units of milliampere hours (mAh).

            Note that the current charge needs to be measured and is
            typically an estimate based on observed discharging and
            charging cycles of the battery.

            A value of 'ffffffff'H indicates that the current charge
            cannot be determined."
        ::= { batteryEntry 10 }

    batteryCurrentChargePercentage OBJECT-TYPE
        SYNTAX      Unsigned32 (0..10000)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the current charge of the battery
            relative to the nominal capacity in units of a hundreds
            of a percent.

    --------------------------
    -- Open issue:
    --    Should it be the percentage of the nominal capacity
    --    or of the current capacity?
    --------------------------
```

```
            Note that this value needs to be measured and is
            typically an estimate based on observed discharging and
            charging cycles of the battery.

            A value of 'ffffffff'H indicates that the relative current
            charge cannot be determined."
        ::= { batteryEntry 11 }

   batteryCurrentVoltage OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "millivolt"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the current voltage of the battery
            in units of millivolt (mV).

            A value of 'ffffffff'H indicates that the current voltage
            cannot be determined."
        ::= { batteryEntry 12 }

   batteryCurrentCurrent OBJECT-TYPE
        SYNTAX      Integer32
        UNITS       "milliampere"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the current charging or discharging
            current of the batteryin units of milliampere (mA).  Charging
            current is indicated by positive values, discharging current
            is indicated by negative values.

            A value of '7fffffff'H indicates that the current current
            cannot be determined."
        ::= { batteryEntry 13 }

   batteryLowAlarmPercentage OBJECT-TYPE
        SYNTAX      Unsigned32 (0..10000)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the lower threshold value for object
            batteryCurrentChargePercentage.  If the value of object
            batteryCurrentChargePercentage falls below this threshold,
            a low battery alarm will be raised.  The alarm procedure may
            include generating a batteryLowNotification.

            A value of 0 indicates that the no alarm will be raised for
```

```
            any value of object batteryCurrentChargePercentage."
        ::= { batteryEntry 14 }

    batteryLowAlarmVoltage OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "millivolt"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the lower threshold value for object
            batteryCurrentVoltage.  If the value of object
            batteryCurrentVoltage falls below this threshold,
            a low battery alarm will be raised.  The alarm procedure may
            include generating a batteryLowNotification.

            A value of 0 indicates that the no alarm will be raised for
            any value of object batteryCurrentVoltage."
        ::= { batteryEntry 15 }

    batteryReplacementAlarmCapacity OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "milliampere hours"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the lower threshold value for object
            batteryRemainingCapacity.  If the value of object
            batteryRemainingCapacity falls below this threshold,
            a battery aging alarm will be raised.  The alarm procedure
            may include generating a batteryAgingNotification.

            A value of 0 indicates that the no alarm will be raised for
            any value of object batteryRemainingCapacity."
        ::= { batteryEntry 16 }

    batteryReplacementAlarmCycles OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "milliampere hours"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object provides the upper threshold value for object
            batteryChargingCycleCount.  If the value of object
            batteryChargingCycleCount rises above this threshold,
            a battery aging alarm will be raised.  The alarm procedure
            may include generating a batteryAgingtNotification.

            A value of 0 indicates that the no alarm will be raised for
```

```
            any value of object batteryChargingCycleCount."
        ::= { batteryEntry 17 }


    --==================================================================
    -- 2. Notifications
    --==================================================================

    batteryLowNotification NOTIFICATION-TYPE
        OBJECTS     {
            batteryCurrentChargePercentage,
            batteryCurrentVoltage
        }
        STATUS      current
        DESCRIPTION
            "This notification can be generated when the current charge
            (batteryCurrentChargePercentage) or the current voltage
            (batteryCurrentVoltage) of the battery falls below a
            threshold defined by object batteryLowAlarmPercentage or
            object batteryLowAlarmVoltage, respectively."
        ::= { batteryNotifications 1 }

    batteryAgingNotification NOTIFICATION-TYPE
        OBJECTS     {
            batteryRemainingCapacity,
            batteryChargingCycleCount
        }
        STATUS      current
        DESCRIPTION
            "This notification can be generated when the remaining
            capacity (batteryRemainingCapacity) falls below a threshold
            defined by object batteryReplacementAlarmCapacity
            or when the charging cycle count of the battery
            (batteryChargingCycleCount) exceeds the threshold defined
            by object batteryLowAlarmPercentage."
        ::= { batteryNotifications 2 }


    --==================================================================
    -- 3. Conformance Information
    --==================================================================

    batteryCompliances OBJECT IDENTIFIER ::= { batteryConformance 1 }
    batteryGroups      OBJECT IDENTIFIER ::= { batteryConformance 2 }

    --------------------------------------------------------------------
    -- 3.1. Compliance Statements
    --------------------------------------------------------------------
```

```
   batteryCompliance MODULE-COMPLIANCE
       STATUS      current
       DESCRIPTION
           "The compliance statement for implementations of the
           POWER-STATE-MIB module.

           A compliant implementation MUST implement the objects
           defined in the mandatory group psmRequiredGroup."
       MODULE  -- this module
       MANDATORY-GROUPS {
           batteryDescriptionGroup,
           batteryStatusGroup,
           batteryAlarmThresholdsGroup
       }
       GROUP   batteryNotificationsGroup
       DESCRIPTION
          "A compliant implementation does not have to implement
           the psmNotificationsGroup."
       ::= { batteryCompliances 1 }

   --------------------------------------------------------------------
   -- 3.2. MIB Grouping
   --------------------------------------------------------------------

   batteryDescriptionGroup OBJECT-GROUP
       OBJECTS {
          batteryType,
          batteryTechnology,
          batteryNominalVoltage,
          batteryNumberOfCells,
          batteryNominalCapacity
       }
       STATUS      current
       DESCRIPTION
           "A compliant implementation MUST implement the objects
           contained in this group."
       ::= { batteryGroups 1 }

   batteryStatusGroup OBJECT-GROUP
       OBJECTS {
          batteryRemainingCapacity,
          batteryChargingCycleCount,
          batteryLastChargingCycleTime,
          batteryState,
          batteryCurrentCharge,
          batteryCurrentChargePercentage,
          batteryCurrentVoltage,
          batteryCurrentCurrent
```

```
        }
        STATUS      current
        DESCRIPTION
            "A compliant implementation MUST implement the objects
            contained in this group."
        ::= { batteryGroups 2 }

    batteryAlarmThresholdsGroup OBJECT-GROUP
        OBJECTS {
           batteryLowAlarmPercentage,
           batteryLowAlarmVoltage,
           batteryReplacementAlarmCapacity,
           batteryReplacementAlarmCycles
        }
        STATUS      current
        DESCRIPTION
            "A compliant implementation MUST implement the objects
            contained in this group."
        ::= { batteryGroups 3 }

    batteryNotificationsGroup NOTIFICATION-GROUP
        NOTIFICATIONS {
           batteryLowNotification,
           batteryAgingNotification
        }
        STATUS      current
        DESCRIPTION
            "A compliant implementation does not have to implement the
            notification contained in this group."
        ::= { batteryGroups 4 }
    END
```

9. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

o This list is still to be done.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

10. IANA Considerations

The MIB modules in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
powerStateMIB	{ mib-2 xxx }
energyMIB	{ mib-2 yyy }
batteryMIB	{ mib-2 zzz }

Other than that this document does not impose any IANA considerations.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC4268] Chisholm, S. and D. Perkins, "Entity State MIB", RFC 4268, November 2005.

[RFC3621] Berger, A. and D. Romascanu, "Power Ethernet MIB", RFC 3621, December 2003.

[RFC3433] Bierman, A., Romascanu, D., and K. Norseth, "Entity Sensor Management Information Base", RFC 3433, December 2002.

[RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

[I-D.quittek-power-monitoring-requirements] Quittek, J., Winter, R., Dietz, T., Claise, B., and M. Chandramouli, "Requirements for Power Monitoring", draft-quittek-power-monitoring-requirements-01 (work in progress), July 2010.

[I-D.claise-energy-monitoring-mib] Claise, B., Chandramouli, M., Parello, J., and B. Schoening, "Power and Energy Monitoring MIB", draft-claise-energy-monitoring-mib-06 (work in progress), October 2010.

[I-D.parello-eman-energy-aware-mib] Claise, B. and J. Parello, "Energy-aware Networks and Devices MIB", draft-parello-eman-energy-aware-mib-00 (work in progress), October 2010.

11.2. Informative References

[RFC1628] Case, J., "UPS Management Information Base", RFC 1628, May 1994.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Authors' Addresses

Juergen Quittek (editor)
NEC Europe Ltd.
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 4342-115
Email: quittek@nw.neclab.eu

Rolf Winter
NEC Europe Ltd.
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 4342-121
Email: Rolf.Winter@nw.neclab.eu

Thomas Dietz
NEC Europe Ltd.
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 4342-128
Email: Thomas.Dietz@nw.neclab.eu

Dominique Dudkowski
NEC Europe Ltd.
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 4342-233
Email: Dominique.Dudkowski@nw.neclab.eu