Network Working Group   J. Quittek, Ed.
Internet-Draft   R. Winter
Intended status: Standards Track   T. Dietz
Expires: August 7, 2010   D. Dudkowski
                                                             NEC Europe Ltd.
                                                            February 3, 2010

Definiton of Managed Objects for Energy Management
draft-quittek-power-mib-00.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 extensions to the Entity MIB to provide information about the energy consumption, the power states and battery status of managed devices and their components.

Status of this Memo

This Internet-Draft will expire on August 7, 2010.

Copyright Notice

Table of Contents

1. The Internet-Standard Management Framework 3

2. Introduction 3

3. Power State MIB 5
   3.1. Current Power State Table 5
   3.2. Power State Table 6

4. Energy MIB 6
   4.1. Energy Consumption Table 7
   4.2. Energy Consumption Per Power State Table 8

5. Battery MIB 8

6. Relationship to Other MIB Modules 9
   6.1. Entity MIB 9
   6.2. Entity State MIB 10
   6.3. Entity Sensor MIB 10
   6.4. UPS MIB 10
   6.5. Power Ethernet MIB 10

7. Definitions 11
   7.1. Power State MIB 11
   7.2. Energy MIB 17
   7.3. Battery MIB 30

8. Security Considerations 42

9. IANA Considerations 43

10. References 43
   10.1. Normative References 43
   10.2. Informative References 44

Authors' Addresses 44

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.

A basic requirement for many energy management procedures is collecting information on energy consumption and energy storage at managed devices. Most devices consume less energy when they are in standby mode compared to their consumption when providing full service. Therefore, management systems will have a need to monitor power state information when conducting basic energy management functions.

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 or of a device's component. Providing this information requires much more effort than reporting power states, because a probe that measures (electrical) power consumption 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:

- monitoring power states of managed entities,
- monitoring energy consumption of managed entities,
- 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].

All three MIB modules are designed as extensions to the Entity MIB module [RFC4133]. They all contain sparse augments of the entPhysicalTable defined in [RFC4133]. The entPhysicalTable already provides information about physical entities such as their type, name, software revision number, serial number, etc., and about their containment in other entities. This is all re-used for entities that have different power states, for which energy consumption can be measured, or that are batteries.

Sparse augmentation of the entPhysicalTable means that tables in the three MIB modules use the index of the entPhysicalTable as their first index. The augment is sparse, because not for every row in the entPhysicalTable there needs to be a row in the tables defined in the MIB modules in this document. For example, the batteryTable will only have rows for index values of the entPhysicalTable that refer to

an entity that is a battery.

Augmentation also means that rows in the tables defined in this document should be created at the same time when the corresponding entry in the entPhysicalTable is created. They also should be destroyed, when the corresponding rows in the entPhysicalTable are destroyed.

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. 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 of an entity and the powerStateTable providing statistics per power state. Both tables use the object entPhysicalIndex from the Entity MIB module [RFC4133] as first index. 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.

3.1. 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(1), hotStandby(2), coldStandby(3), providingService(4), 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 single object per row:

```
powerCurrentStateTable(1)
+--powerCurrentStateEntry(1) [entPhysicalIndex]
   +-- r-n EntityStandbyStatus powerCurrentState(1)
```

Object powerCurrentState reports the actual power state of an entity at the time the object's value is retrieved. In the current definition of the MIB module, it just reports the four values defined for the Entity State MIB module. It does not make sense to keep it like this by just replicating existing functionality. Either the range of supporting values will be extended or the powerCurrentStateTable will be removed from the Power State MIB module. It is currently there as a placeholder until discussion on the number of power states to be supported comes to a conclusion.

3.2. Power State Table

The second table called powerStateTable provides more detailed statistics for each power state. For this purpose it uses the power state value as another index object next to the entity index. This way, statistics can be reported per entity and per power state.

```
powerStateTable(2)
+--powerStateEntry(1) [entPhysicalIndex,powerState]
   +-- --- EntityStandbyStatus powerState(1)
   +-- r-n TimeTicks           powerStateTotalTime(2)
   +-- r-n TimeStamp           powerStateLastEnterTime(3)
   +-- r-n SnmpAdminString     powerStateLastEnterReason(4)
   +-- r-n Counter64           powerStateEnterCount(5)
```

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) and the number of times a certain state has been entered (powerStateEnterCount).

4. 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 energyConsumpTable and

the energyConsumpPSTable. 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.

4.1. Energy Consumption Table

The first set of managed objects in the energyConsumpTable are needed to help interpreting the energy consumption readings. These include the sampling interval applied by the sensor(s) and the power supply type and voltage.

```
energyConsumpTable(1)
+--energyConsumpEntry(1) [entPhysicalIndex]
   +-- r-n EntitySensorStatus    energyConsumpSensorOperStatus(1)
   +-- r-n Unsigned32            energyConsumpSampleInterval(2)
   +-- r-n Unsigned32            energyConsumpNominalSupplyVoltage(3)
   +-- r-n Enumeration           energyConsumpElectricSupplyType(4)
   +-- r-n EntitySensorValue     energyConsumpTotalEnergy(5)
   +-- r-n EntitySensorDataScale energyConsumpEnergyScale(6)
   +-- r-n EntitySensorPrecision energyConsumpEnergyPrecision(7)
   +-- r-n TimeStamp             energyConsumpDiscontinuityTime(8)
   +-- r-n EntitySensorDataScale energyConsumpPowerScale(9)
   +-- r-n EntitySensorPrecision energyConsumpPowerPrecision(10)
   +-- r-n EntitySensorValue     energyConsumpRealPower(11)
   +-- r-n EntitySensorValue     energyConsumpPeakRealPower(12)
   +-- r-n EntitySensorValue     energyConsumpReactivePower(13)
   +-- r-n EntitySensorValue     energyConsumpApparentPower(14)
   +-- r-n EntitySensorValue     energyConsumpPhaseAngle(15)
   +-- r-n EntitySensorPrecision energyConsumpPhaseAnglePrecision(16)
```

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.

All measured values are defined to be of type EntitySensorValue defined by the Entity Sensor MIB module [RFC3433]. For this data type scale and precision can be specified by additional objects of types EntitySensorDataScale and EntitySensorPrecision. The energyConsumpTable makes use of this mechanism and contains a set of objects for this purpose.

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 energyConsumpDiscontinuityTime is provided for indicating the time of

the last interruption of total energy measurement.

4.2. Energy Consumption Per Power State Table

The second table in this module is called energyConsumpPSTable and it provides values of total energy consumption per power state in a way similar to the powerStateTable in the Power State MIB module.

```
energyConsumpPSTable(2)
+--energyConsumpPSEntry(1) [entPhysicalIndex,powerState]
   +-- r-n EntitySensorValue energyConsumpPSTotalEnergy(1)
```

5. Battery MIB

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.

6. 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.

6.1. Entity MIB

All MIB modules defined in this document implement a sparse augmentation of the entPhysicalTable defined in the Entity MIB module [RFC4133]. This means that tables defined in the MIB modules in this document use the index of the entPhysicalTable called entPhysicalIndex as their first index, in most cases as their only index. The augmentation is sparse meaning that entries in tables defined in this document do not need to create entries for all entries that exist in the entPhysicalTable. Entries can be restricted to relevant ones, for example, the batteryTable can restrict their entries to entities that are a battery.

The advantage of augmenting the Entity MIB instead of defining new tables from scratch is the re-use of many objects in the

entPhysicalTable. For example, the kind of entity (entPhysicalClass), the serial number (entPhysicalSerialNum), the software version (entPhysicalSoftwareRev) and many other properties are covered by objects in the entPhysicalTable as well as the containment relationship between entities. The containment relationship indicates, for example, in which device a battery entity is contained.

6.2. Entity State MIB

The Entity State MIB module [RFC4268] defines object entStateStandby in the entStateTable. This object provides information on the power state. This object may have one of four defined values: unknown(1), hotStandby(2), coldStandby(3), providingService(4). If this number was considered to be sufficient, the powerCurrentStateTable of the Power State MIB module would be obsolete and should be removed. However, there are concerns that this number is not sufficient. Discussions in the IETF will hopefully soon lead to a consensus on which is the better way to go.

6.3. Entity Sensor MIB

The Entity Sensor MIB module [RFC3433] defines generic objects for providing data from sensors such as, for example, an energy consumption meter. Basically, some of the objects defined in the Energy MIB module could be replaced by objects in the Entity Sensor MIB module. However, in the Entity Sensor MIB module more objects are needed to model the same information and flexibility is not fully sufficient. For example, there is no unit for energy supported, such as, for example, watt hours; a sampling interval for the sensor cannot be specified, and there is not support for reporting discontinuities of accumulated measurements, such as the total consumed energy. For these reasons, the Energy MIB was defined as new module instead of using the Entity Sensor MIB module.

6.4. UPS MIB

Relations to UPS MIB module [RFC1628] are still to be done.

6.5. Power Ethernet MIB

Relations to Power Ethernet MIB module [RFC3621] are still to be done.

7. Definitions

7.1. Power State MIB

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

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       mib-2, 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
       entPhysicalIndex
           FROM ENTITY-MIB                              -- RFC4133
       EntityStandbyStatus
           FROM ENTITY-STATE-TC-MIB;                    -- RFC4268

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

           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"
```

```
       DESCRIPTION
           "This MIB module defines a set of objects for monitoring
           the power state of networked devices and their components.

           Copyright (c) 2009 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 xxx }
   -- 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. Current Power State Table
   -------------------------------------------------------------------
   powerCurrentStateTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF PowerCurrentStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
```

```
           "This table provides information on the current power state
           of entities.

           This is a sparse augment of the entPhysicalTable.
           Entries appear in this table for entities for which their
           power state can be reported.

           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."
       ::= { powerStateObjects 1 }

   --------------------------
   -- Open issue: This table duplicates a part of the entStateTable
   --    in the ENTITY-STATE-MIB (RFC 4268).
   --    It does not make sense to keep it as it is.
   --
   --    The entStateTable only supports four power states:
   --    unknown(1), hotStandby(2), coldStandby(3),
   --    providingService(4) (see ENTITY-STATE-TC-MIB, RFC 4268).
   --    If this is considered to be sufficient, then the
   --    powerCurrentPowerStateTable should be removed.
   --    But if there is consensus that supporting more power states
   --    would be needed, as claimed, for example, by
   --    draft-claise-energy-monitoring-mib-00, then this table might
   --    be useful to have with the extended range of power states.
   --------------------------

   powerCurrentStateEntry OBJECT-TYPE
       SYNTAX      PowerCurrentStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An entry providing information on the current power state
           of an entity."
       INDEX  { entPhysicalIndex }    -- SPARSE-AUGMENTS
       ::= { powerCurrentStateTable 1 }

   PowerCurrentStateEntry ::=
       SEQUENCE {
          powerCurrentState   EntityStandbyStatus
       }

   powerCurrentState OBJECT-TYPE
       SYNTAX      EntityStandbyStatus
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
```

```
           "This object indicates the current power state of the
           entity indicated by the index entPhysicalIndex."
       ::= { powerCurrentStateEntry 1 }

   -------------------------------------------------------------------
   -- 1.2. Power State Statistics 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 entities.

           This is a sparse augment of the entPhysicalTable.
           Entries appear in this table for entities for which
           statistics on their power state can be reported.

           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.

           As second index for this table serves the power state of the
           entity indicated by the first index."
       ::= { powerStateObjects 2 }

   powerStateEntry OBJECT-TYPE
       SYNTAX      PowerStateEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Power state information about this physical entity."
       INDEX       { entPhysicalIndex, powerState }
       ::= { powerStateTable 1 }

   PowerStateEntry ::=
       SEQUENCE {
          powerState                 EntityStandbyStatus,
          powerStateTotalTime        TimeTicks,
          powerStateLastEnterTime    TimeStamp,
          powerStateLastEnterReason  SnmpAdminString,
          powerStateEnterCount       Counter64
       }

   powerState OBJECT-TYPE
       SYNTAX      EntityStandbyStatus
       MAX-ACCESS  not-accessible
```

```
       STATUS      current
       DESCRIPTION
           "This index should only be created for power states
           that are actually used by the entity that is identified
           by the first index entPhysicalIndex."
       ::= { powerStateEntry 1 }

   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 powerState."
       ::= { powerStateEntry 2 }

   --------------------------
   -- 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 powerState."
       ::= { powerStateEntry 3 }

   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 powerState."
       ::= { powerStateEntry 4 }

   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 powerState."
```

```
       ::= { powerStateEntry 5 }


   --===================================================================
   -- 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 {
          powerCurrentState,
          powerStateTotalTime,
          powerStateLastEnterTime,
          powerStateLastEnterReason,
          powerStateEnterCount
       }
       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
```

7.2. Energy MIB

```
   ENERGY-MIB DEFINITIONS ::= BEGIN

   IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, mib-2, Unsigned32
           FROM SNMPv2-SMI                                   -- RFC2578
       TimeStamp
           FROM SNMPv2-TC                                    -- RFC2579
       MODULE-COMPLIANCE, OBJECT-GROUP
           FROM SNMPv2-CONF                                  -- RFC2580
       entPhysicalIndex
           FROM ENTITY-MIB                                   -- RFC4133
       EntitySensorDataScale, EntitySensorPrecision,
       EntitySensorValue, EntitySensorStatus
           FROM ENTITY-SENSOR-MIB                            -- RFC3433
       powerState
           FROM POWER-STATE-MIB;

   energyMIB 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
           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     "201001291200Z"         -- 29 January 2010
```

```
       DESCRIPTION
           "Initial version, published as RFC yyyy."
   -- replace yyyy with actual RFC number & remove this notice

       ::= { mib-2 yyy }
   -- 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
   --------------------------------------------------------------------
   energyConsumpTable  OBJECT-TYPE
       SYNTAX      SEQUENCE OF EnergyConsumpEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table provides inforamtion on the current and
           accumulated energy consumption of entities.

           This is a sparse augment of the entPhysicalTable.
           Entries appear in this table for entities for which their
           energy consumption can be reported.

           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."
       ::= { energyObjects 1 }

   energyConsumpEntry OBJECT-TYPE
       SYNTAX      EnergyConsumpEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An entry providing information on the energy consumption
           of physical entity."
       INDEX  { entPhysicalIndex }    -- SPARSE-AUGMENTS
       ::= { energyConsumpTable 1 }
```

```
   EnergyConsumpEntry ::=
       SEQUENCE {
          energyConsumpSensorOperStatus       EntitySensorStatus,
          energyConsumpSampleInterval         Unsigned32,
          energyConsumpNominalSupplyVoltage   Unsigned32,
          energyConsumpElectricSupplyType     INTEGER,
          energyConsumpTotalEnergy            EntitySensorValue,
          energyConsumpEnergyScale            EntitySensorDataScale,
          energyConsumpEnergyPrecision        EntitySensorPrecision,
          energyConsumpDiscontinuityTime      TimeStamp,
          energyConsumpPowerScale             EntitySensorDataScale,
          energyConsumpPowerPrecision         EntitySensorPrecision,
          energyConsumpRealPower              EntitySensorValue,
          energyConsumpPeakRealPower          EntitySensorValue,
          energyConsumpReactivePower          EntitySensorValue,
          energyConsumpApparentPower          EntitySensorValue,
          energyConsumpPhaseAngle             EntitySensorValue,
          energyConsumpPhaseAnglePrecision    EntitySensorPrecision
       }

   energyConsumpSensorOperStatus 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 entPhysicalIndex."
       ::= { energyConsumpEntry 1 }

   energyConsumpSampleInterval OBJECT-TYPE
       SYNTAX      Unsigned32
       UNITS       "milliseconds"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object provides the sampling rate applied by the
           energy consumption sensor for calculating the current power
           value.

           For alternating current (AC) power supply the sampling
           interval should be at least have half the size of a period
           of alternation.

           The sampling interval is provided in units of microseconds.

           A value of 0 indicates that the sampling interval applied by
           the sensor is unknown."
```

```
       ::= { energyConsumpEntry 2 }

   energyConsumpNominalSupplyVoltage 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."
       ::= { energyConsumpEntry 3 }

   energyConsumpElectricSupplyType 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."
       ::= { energyConsumpEntry 4 }

   energyConsumpTotalEnergy OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "watt hours"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total consumed energy measured in
           watt hours at the electrical power supply of the entity.

           Scale and precision of the value are indicated
           by objects energyConsumpEnergyScale and
           energyConsumpEnergyPrecision.
```

```
           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
           energyConsumpDiscontinuityTime."
       ::= { energyConsumpEntry 5 }

   energyConsumpEnergyScale OBJECT-TYPE
       SYNTAX      EntitySensorDataScale
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the scale of the values provided
           by objects energyConsumpTotalEnergy and
           energyConsumpPSTotalEnergy."
       ::= { energyConsumpEntry 6 }

   energyConsumpEnergyPrecision OBJECT-TYPE
       SYNTAX      EntitySensorPrecision
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the precision of the values provided
           by objects energyConsumpTotalEnergy and
           energyConsumpPSTotalEnergy."
       ::= { energyConsumpEntry 7 }

   energyConsumpDiscontinuityTime 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
           energyConsumpTotalEnergy and energyConsumpPSTotalEnergy.  If
           no such discontinuities have occurred since the last re-
           initialization of the local management subsystem, then this
           object contains a zero value."
       ::= { energyConsumpEntry 8 }

   energyConsumpPowerScale OBJECT-TYPE
       SYNTAX      EntitySensorDataScale
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the scale of the values provided by
           objects energyConsumpRealPower, energyConsumpPeakRealPower,
           energyConsumpReactivePower, and energyConsumpApparentPower."
```

```
       ::= { energyConsumpEntry 9 }

   energyConsumpPowerPrecision OBJECT-TYPE
       SYNTAX      EntitySensorPrecision
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the precision of values provided by
           objects energyConsumpRealPower, energyConsumpPeakRealPower,
           energyConsumpReactivePower, and energyConsumpApparentPower."
       ::= { energyConsumpEntry 10 }

   energyConsumpRealPower OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "watts"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the current real power value
           measured in watts at the electrical supply of the entity
           for a time interval indicated by object
           energyConsumpSampleInterval.

           Scale and precision of the value are indicated by objects
           energyConsumpPowerScale and energyConsumpPowerPrecision."
       ::= { energyConsumpEntry 11 }

   energyConsumpPeakRealPower OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "watts"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the highest observed value for
           object energyConsumpRealPower since the last
           re-initialization of the management system.

           Scale and precision of the value are indicated by objects
           energyConsumpPowerScale and energyConsumpPowerPrecision."
       ::= { energyConsumpEntry 12 }

   energyConsumpReactivePower OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "volt-amperes reactive"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the current reactive power value
```

```
           measured in volt-amperes reactive (var) at the electrical
           supply of the entity for a time interval indicated by object
           energyConsumpSampleInterval.

           The value provided by this object is only useful if the
           value of object energyConsumpSupplyType is
           alternatingCurrent(1). In this case it is RECOMMENDED that
           at least one of the three values energyConsumpReactivePower,
           energyConsumpApparentPowerScale, and energyConsumpPhaseAngle
           are provided.

           Scale and precision of the value are indicated by objects
           energyConsumpPowerScale and energyConsumpPowerPrecision.

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

           If object energyConsumpElectricSupplyType 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 -1000000000000."
       ::= { energyConsumpEntry 13 }

   energyConsumpApparentPower OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "volt-ampere"
       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
           energyConsumpSampleInterval.

           The value provided by this object is only useful if the
           value of object energyConsumpSupplyType is
           alternatingCurrent(1). In this case it is RECOMMENDED that
           at least one of the three values energyConsumpReactivePower,
           energyConsumpApparentPowerScale, and energyConsumpPhaseAngle
           are provided.

           Scale and precision of the value are indicated by objects
           energyConsumpPowerScale and energyConsumpPowerPrecision.

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

```
            energyConsumpRealPower.

            If object energyConsumpElectricSupplyType 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."
        ::= { energyConsumpEntry 14 }

    energyConsumpPhaseAngle OBJECT-TYPE
        SYNTAX      EntitySensorValue
        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 energyConsumpSampleInterval.

            The value provided by this object is only useful if the
            value of object energyConsumpSupplyType is
            alternatingCurrent(1). In this case it is RECOMMENDED that
            at least one of the three values energyConsumpReactivePower,
            energyConsumpApparentPowerScale, and energyConsumpPhaseAngle
            are provided.

            The value is provided in units of millidegree (one thousands
            of a degree. The minimum value for this object is -180000,
            the maximum value is 180000. Since the scaling factor is
            constant, there is no object of type EntitySensorDataScale
            provided for object energyConsumpPhaseAngle.

            The precision of the value is indicated by object
            energyConsumpPhaseAnglePrecision.

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

            If object energyConsumpElectricSupplyType 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 -10000000000."
        ::= { energyConsumpEntry 15 }

    energyConsumpPhaseAnglePrecision OBJECT-TYPE
        SYNTAX      EntitySensorPrecision
        MAX-ACCESS  read-only
        STATUS      current
```

```
       DESCRIPTION
           "This object indicates the precision of the value provided
           by object energyConsumpApparentPower."
       ::= { energyConsumpEntry 16 }


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

           This is a sparse augment of the entPhysicalTable.
           Entries appear in this table for entities for which their
           energy consumption can be reported per power state.

           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."
       ::= { energyObjects 2 }

   energyConsumpPSEntry OBJECT-TYPE
       SYNTAX      EnergyConsumpPSEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "Energy consumption information per power state
           for a physical entity."
       INDEX  { entPhysicalIndex, powerState }
       ::= { energyConsumpPSTable 1 }

   EnergyConsumpPSEntry ::=
       SEQUENCE {
          energyConsumpPSTotalEnergy          EntitySensorValue
       }

   energyConsumpPSTotalEnergy OBJECT-TYPE
       SYNTAX      EntitySensorValue
       UNITS       "watt hours"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "This object indicates the total consumed energy measured in
```

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

            Scale and precision of the value are indicated
            by objects energyConsumpEnergyScale and
            energyConsumpEnergyPrecision.

            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
            energyConsumpDiscontinuityTime."
        ::= { energyConsumpPSEntry 1 }


   --==================================================================
   -- 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 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 {
          energyConsumpSensorOperStatus,
          energyConsumpSampleInterval,
          energyConsumpNominalSupplyVoltage,
          energyConsumpElectricSupplyType,
          energyConsumpTotalEnergy,
          energyConsumpEnergyScale,
          energyConsumpEnergyPrecision,
          energyConsumpDiscontinuityTime,
          energyConsumpPowerScale,
          energyConsumpPowerPrecision,
          energyConsumpRealPower,
          energyConsumpPeakRealPower,
          energyConsumpPSTotalEnergy
       }
       STATUS      current
       DESCRIPTION
           "A compliant implementation MUST implement the objects
           contained in this group."
       ::= { energyGroups 1 }

   energyACGroup OBJECT-GROUP
       OBJECTS {
          energyConsumpReactivePower,
          energyConsumpApparentPower,
          energyConsumpPhaseAngle,
          energyConsumpPhaseAnglePrecision
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting details of
           AC power measurement."
       ::= { energyGroups 2 }

   energyReactivePowerGroup OBJECT-GROUP
       OBJECTS {
          energyConsumpReactivePower
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the reactive power
           measured for AC supply."
       ::= { energyGroups 3 }

   energyApparentPowerGroup OBJECT-GROUP
       OBJECTS {
          energyConsumpApparentPower
```

```
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the apparent power
           measured for AC supply."
       ::= { energyGroups 4 }

   energyPhaseAngleGroup OBJECT-GROUP
       OBJECTS {
          energyConsumpPhaseAngle,
          energyConsumpPhaseAnglePrecision
       }
       STATUS      current
       DESCRIPTION
           "The group of object for reporting the phase angler
           measured for AC supply."
       ::= { energyGroups 5 }

   END
```

7.3. 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
```

8. 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:

- 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.

9. 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.

10. References

10.1. 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-00 (work in progress), October 2009.

[I-D.claise-energy-monitoring-mib]
Claise, B., Chandramouli, M., Parello, J., and B. Schoening, "Energy Monitoring MIB", draft-claise-energy-monitoring-mib-00 (work in progress), January 2010.

10.2. 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