Network Working Group
Request for Comments: 3433
Category: Standards Track

A. Bierman
Cisco Systems, Inc.
D. Romascanu
Avaya Inc.
K.C. Norseth
L-3 Communications

December 2002

Entity Sensor Management Information Base

Status of this Memo

Copyright Notice

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 for extending the Entity MIB (RFC 2737) to provide generalized access to information related to physical sensors, which are often found in networking equipment (such as chassis temperature, fan RPM, power supply voltage).

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 a MIB module 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. Overview

There is a need for a standardized way of obtaining information related to the physical sensors which are commonly found in networking equipment. Information such as the current value of the sensor, the current operational status, and the data units precision associated with the sensor, should be represented in a consistent manner for any type of sensor.

Physical sensors are represented in the Entity MIB with entPhysicalEntry and an entPhysicalClass value of 'sensor(8)'. The information provided in the ENTITY-SENSOR-MIB module (defined in this document) defines a sparse augmentation of the entPhysicalTable, for entries which represent physical sensors.

2.1. Terms

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 BCP 14, RFC 2119. [RFC2119]

2.2. Relationship to the Entity MIB

The MIB objects defined in this document provide a sparse augmentation to the entPhysicalTable in the Entity MIB, for entries in which the associated entPhysicalClass object is equal to 'sensor(8)'. An agent is expected to maintain an entPhySensorEntry with the same entPhysicalIndex value for each entPhysicalEntry representing a physical sensor. Therefore, implementation of the entityPhysicalGroup is required for agents that implement the Entity Sensor MIB.

2.3. Relationship to General Thresholding Mechanisms

There are no specialized sensor value thresholding mechanisms defined in this MIB module. Instead, it is recommended that a generalized thresholding MIB, such as the mechanisms defined by the Alarm and Events groups of the Remote Network Monitoring MIB [RFC2819], be used for this purpose.

3. MIB Structure

The Entity Sensor MIB contains a single group called the entitySensorValueGroup, which allows objects to convey the current value and status of a physical sensor.

The entitySensorValueGroup contains a single table, called the entPhySensorTable, which provides a small number of read-only objects:

entPhySensorType
: This object identifies the type of data units associated with the sensor value.

entPhySensorScale
: This object identifies the (power of 10) scaling factor associated with the sensor value.

entPhySensorPrecision
: This object identifies the number of decimal places of precision associated with the sensor value.

entPhySensorValue
: This object identifies the current value of the sensor.

entPhySensorOperStatus
: This object identifies the current operational status of the sensor (as it’s known to the agent).

entPhySensorUnitsDisplay
: This object provides a textual description of the data units represented by the entPhySensorType and entPhySensorScale objects.

entPhySensorValueTimeStamp
: The object identifies the value of sysUpTime at the time the agent last updated the information in the entry. This object is only relevant if the agent uses a polling implementation strategy, (i.e., the associated entPhySensorValueUpdateRate object is greater than zero).

entPhySensorValueUpdateRate
This object indicates the nature of the agent implementation of the entPhySensorEntry, and contains the (possibly estimated) number of milliseconds that elapse between polling updates of the information in the associated entry. The value zero indicates that the agent always return current data for the entry (as opposed to the data as it was at the last polling interval).

4. Definitions

```
ENTITY-SENSOR-MIB DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        Integer32, Unsigned32, mib-2
                FROM SNMPv2-SMI
        MODULE-COMPLIANCE, OBJECT-GROUP
                FROM SNMPv2-CONF
        TEXTUAL-CONVENTION, TimeStamp
                FROM SNMPv2-TC
        entPhysicalIndex, entityPhysicalGroup
                FROM ENTITY-MIB
        SnmpAdminString
                FROM SNMP-FRAMEWORK-MIB;

entitySensorMIB MODULE-IDENTITY
    LAST-UPDATED    "200212160000Z"
    ORGANIZATION    "IETF Entity MIB Working Group"
    CONTACT-INFO
            "        Andy Bierman
                     Cisco Systems, Inc.
                Tel: +1 408-527-3711
             E-mail: abierman@cisco.com
             Postal: 170 West Tasman Drive
                     San Jose, CA USA 95134

                     Dan Romascanu
                     Avaya Inc.
                Tel: +972-3-645-8414
              Email: dromasca@avaya.com
             Postal: Atidim technology Park, Bldg. #3
                     Tel Aviv, Israel, 61131

                     K.C. Norseth
                     L-3 Communications
                Tel: +1 801-594-2809
              Email: kenyon.c.norseth@L-3com.com
             Postal: 640 N. 2200 West.
```

```
                    Salt Lake City, Utah 84116-0850

              Send comments to <entmib@ietf.org>
              Mailing list subscription info:
                http://www.ietf.org/mailman/listinfo/entmib "
    DESCRIPTION
            "This module defines Entity MIB extensions for physical
             sensors.

             Copyright (C) The Internet Society (2002). This version
             of this MIB module is part of RFC 3433; see the RFC
             itself for full legal notices."

    REVISION        "200212160000Z"
    DESCRIPTION
            "Initial version of the Entity Sensor MIB module, published
             as RFC 3433."
    ::= { mib-2 99 }

entitySensorObjects              OBJECT IDENTIFIER
     ::= { entitySensorMIB 1 }

-- entitySensorNotifications        OBJECT IDENTIFIER
--     ::= { entitySensorMIB 2 }
entitySensorConformance          OBJECT IDENTIFIER
     ::= { entitySensorMIB 3 }

--
-- Textual Conventions
--

EntitySensorDataType ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "An object using this data type represents the Entity Sensor
            measurement data type associated with a physical sensor
            value. The actual data units are determined by examining an
            object of this type together with the associated
            EntitySensorDataScale object.

            An object of this type SHOULD be defined together with
            objects of type EntitySensorDataScale and
            EntitySensorPrecision.  Together, associated objects of
            these three types are used to identify the semantics of an
            object of type EntitySensorValue.
```

```
            Valid values are:

               other(1):        a measure other than those listed below
               unknown(2):      unknown measurement, or arbitrary,
                                relative numbers
               voltsAC(3):      electric potential
               voltsDC(4):      electric potential
               amperes(5):      electric current
               watts(6):        power
               hertz(7):        frequency
               celsius(8):      temperature
               percentRH(9):    percent relative humidity
               rpm(10):         shaft revolutions per minute
               cmm(11),:        cubic meters per minute (airflow)
               truthvalue(12):  value takes { true(1), false(2) }

            "
        SYNTAX INTEGER {
            other(1),
            unknown(2),
            voltsAC(3),
            voltsDC(4),
            amperes(5),
            watts(6),
            hertz(7),
            celsius(8),
            percentRH(9),
            rpm(10),
            cmm(11),
            truthvalue(12)
        }

    EntitySensorDataScale ::= TEXTUAL-CONVENTION
        STATUS       current
        DESCRIPTION
                "An object using this data type represents a data scaling
                factor, represented with an International System of Units
                (SI) prefix.  The actual data units are determined by
                examining an object of this type together with the
                associated EntitySensorDataType object.

                An object of this type SHOULD be defined together with
                objects of type EntitySensorDataType and
                EntitySensorPrecision.  Together, associated objects of
                these three types are used to identify the semantics of an
                object of type EntitySensorValue."
        REFERENCE
                "The International System of Units (SI),
```

```
            National Institute of Standards and Technology,
            Spec. Publ. 330, August 1991."
        SYNTAX INTEGER {
            yocto(1),   -- 10^-24
            zepto(2),   -- 10^-21
            atto(3),    -- 10^-18
            femto(4),   -- 10^-15
            pico(5),    -- 10^-12
            nano(6),    -- 10^-9
            micro(7),   -- 10^-6
            milli(8),   -- 10^-3
            units(9),   -- 10^0
            kilo(10),   -- 10^3
            mega(11),   -- 10^6
            giga(12),   -- 10^9
            tera(13),   -- 10^12
            exa(14),    -- 10^15
            peta(15),   -- 10^18
            zetta(16),  -- 10^21
            yotta(17)   -- 10^24
        }

    EntitySensorPrecision ::= TEXTUAL-CONVENTION
        STATUS       current
        DESCRIPTION
                "An object using this data type represents a sensor
                precision range.

                An object of this type SHOULD be defined together with
                objects of type EntitySensorDataType and
                EntitySensorDataScale.  Together, associated objects of
                these three types are used to identify the semantics of an
                object of type EntitySensorValue.

                If an object of this type contains a value in the range 1 to
                9, it represents the number of decimal places in the
                fractional part of an associated EntitySensorValue fixed-
                point number.

                If an object of this type contains a value in the range -8
                to -1, it represents the number of accurate digits in the
                associated EntitySensorValue fixed-point number.

                The value zero indicates the associated EntitySensorValue
                object is not a fixed-point number.

                Agent implementors must choose a value for the associated
                EntitySensorPrecision object so that the precision and
```

```
            accuracy of the associated EntitySensorValue object is
            correctly indicated.

            For example, a physical entity representing a temperature
            sensor that can measure 0 degrees to 100 degrees C in 0.1
            degree increments, +/- 0.05 degrees, would have an
            EntitySensorPrecision value of '1', an EntitySensorDataScale
            value of 'units(9)', and an EntitySensorValue ranging from
            '0' to '1000'.  The EntitySensorValue would be interpreted
            as 'degrees C * 10'."
        SYNTAX Integer32 (-8..9)

EntitySensorValue ::= TEXTUAL-CONVENTION
        STATUS       current
        DESCRIPTION
                "An object using this data type represents an Entity Sensor
                value.
                An object of this type SHOULD be defined together with
                objects of type EntitySensorDataType, EntitySensorDataScale
                and EntitySensorPrecision.  Together, associated objects of
                those three types are used to identify the semantics of an
                object of this data type.

                The semantics of an object using this data type are
                determined by the value of the associated
                EntitySensorDataType object.

                If the associated EntitySensorDataType object is equal to
                'voltsAC(3)', 'voltsDC(4)', 'amperes(5)', 'watts(6),
                'hertz(7)', 'celsius(8)', or 'cmm(11)', then an object of
                this type MUST contain a fixed point number ranging from
                -999,999,999 to +999,999,999.  The value -1000000000
                indicates an underflow error. The value +1000000000
                indicates an overflow error.  The EntitySensorPrecision
                indicates how many fractional digits are represented in the
                associated EntitySensorValue object.

                If the associated EntitySensorDataType object is equal to
                'percentRH(9)', then an object of this type MUST contain a
                number ranging from 0 to 100.

                If the associated EntitySensorDataType object is equal to
                'rpm(10)', then an object of this type MUST contain a number
                ranging from -999,999,999 to +999,999,999.

                If the associated EntitySensorDataType object is equal to
                'truthvalue(12)', then an object of this type MUST contain
                either the value 'true(1)' or the value 'false(2)'.
```

```
            If the associated EntitySensorDataType object is equal to
            'other(1)' or unknown(2)', then an object of this type MUST
            contain a number ranging from -1000000000 to 1000000000."
    SYNTAX Integer32 (-1000000000..1000000000)

EntitySensorStatus ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
            "An object using this data type represents the operational
            status of a physical sensor.

            The value 'ok(1)' indicates that the agent can obtain the
            sensor value.

            The value 'unavailable(2)' indicates that the agent
            presently cannot obtain the sensor value.

            The value 'nonoperational(3)' indicates that the agent
            believes the sensor is broken.  The sensor could have a hard
            failure (disconnected wire), or a soft failure such as out-
            of-range, jittery, or wildly fluctuating readings."
    SYNTAX INTEGER {
        ok(1),
        unavailable(2),
        nonoperational(3)
    }

--
-- Entity Sensor Table
--

entPhySensorTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF EntPhySensorEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
            "This table contains one row per physical sensor represented
            by an associated row in the entPhysicalTable."
    ::= { entitySensorObjects 1 }

entPhySensorEntry OBJECT-TYPE
    SYNTAX        EntPhySensorEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
            "Information about a particular physical sensor.
```

```
            An entry in this table describes the present reading of a
            sensor, the measurement units and scale, and sensor
            operational status.

            Entries are created in this table by the agent.  An entry
            for each physical sensor SHOULD be created at the same time
            as the associated entPhysicalEntry.  An entry SHOULD be
            destroyed if the associated entPhysicalEntry is destroyed."
    INDEX  { entPhysicalIndex }    -- SPARSE-AUGMENTS
    ::= { entPhySensorTable 1 }

EntPhySensorEntry ::= SEQUENCE {
        entPhySensorType            EntitySensorDataType,
        entPhySensorScale           EntitySensorDataScale,
        entPhySensorPrecision       EntitySensorPrecision,
        entPhySensorValue           EntitySensorValue,
        entPhySensorOperStatus      EntitySensorStatus,
        entPhySensorUnitsDisplay    SnmpAdminString,
        entPhySensorValueTimeStamp  TimeStamp,
        entPhySensorValueUpdateRate Unsigned32
}

entPhySensorType OBJECT-TYPE
    SYNTAX        EntitySensorDataType
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The type of data returned by the associated
            entPhySensorValue object.

            This object SHOULD be set by the agent during entry
            creation, and the value SHOULD NOT change during operation."
    ::= { entPhySensorEntry 1 }

entPhySensorScale OBJECT-TYPE
    SYNTAX        EntitySensorDataScale
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The exponent to apply to values returned by the associated
            entPhySensorValue object.

            This object SHOULD be set by the agent during entry
            creation, and the value SHOULD NOT change during operation."
    ::= { entPhySensorEntry 2 }
```

```
entPhySensorPrecision OBJECT-TYPE
    SYNTAX        EntitySensorPrecision
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The number of decimal places of precision in fixed-point
            sensor values returned by the associated entPhySensorValue
            object.

            This object SHOULD be set to '0' when the associated
            entPhySensorType value is not a fixed-point type: e.g.,
            'percentRH(9)', 'rpm(10)', 'cmm(11)', or 'truthvalue(12)'.

            This object SHOULD be set by the agent during entry
            creation, and the value SHOULD NOT change during operation."
    ::= { entPhySensorEntry 3 }

entPhySensorValue OBJECT-TYPE
    SYNTAX        EntitySensorValue
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The most recent measurement obtained by the agent for this
            sensor.

            To correctly interpret the value of this object, the
            associated entPhySensorType, entPhySensorScale, and
            entPhySensorPrecision objects must also be examined."
    ::= { entPhySensorEntry 4 }

entPhySensorOperStatus OBJECT-TYPE
    SYNTAX        EntitySensorStatus
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The operational status of the sensor."
    ::= { entPhySensorEntry 5 }

entPhySensorUnitsDisplay OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "A textual description of the data units that should be used
            in the display of entPhySensorValue."
    ::= { entPhySensorEntry 6 }
```

```
entPhySensorValueTimeStamp OBJECT-TYPE
    SYNTAX        TimeStamp
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "The value of sysUpTime at the time the status and/or value
            of this sensor was last obtained by the agent."
    ::= { entPhySensorEntry 7 }

entPhySensorValueUpdateRate  OBJECT-TYPE
    SYNTAX        Unsigned32
    UNITS         "milliseconds"
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
            "An indication of the frequency that the agent updates the
            associated entPhySensorValue object, representing in
            milliseconds.

            The value zero indicates:

                - the sensor value is updated on demand (e.g.,
                  when polled by the agent for a get-request),
                - the sensor value is updated when the sensor
                  value changes (event-driven),
                - the agent does not know the update rate.

            "
    ::= { entPhySensorEntry 8 }

--
-- Conformance Section
--

entitySensorCompliances OBJECT IDENTIFIER
    ::= { entitySensorConformance 1 }
entitySensorGroups      OBJECT IDENTIFIER
    ::= { entitySensorConformance 2 }

entitySensorCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "Describes the requirements for conformance to the Entity
            Sensor MIB module."
    MODULE  -- this module
        MANDATORY-GROUPS { entitySensorValueGroup }
```

```
    MODULE ENTITY-MIB
        MANDATORY-GROUPS { entityPhysicalGroup }

    ::= { entitySensorCompliances 1 }

-- Object Groups

entitySensorValueGroup OBJECT-GROUP
    OBJECTS {
             entPhySensorType,
             entPhySensorScale,
             entPhySensorPrecision,
             entPhySensorValue,
             entPhySensorOperStatus,
             entPhySensorUnitsDisplay,
             entPhySensorValueTimeStamp,
             entPhySensorValueUpdateRate
    }
    STATUS  current
    DESCRIPTION
            "A collection of objects representing physical entity sensor
            information."
    ::= { entitySensorGroups 1 }

END
```

5. Intellectual Property

6. Acknowledgements

This memo is a product of the Entity MIB working group. It is based on an existing proprietary MIB module written by Cliff Sojourner.

7. Normative References

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

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

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

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

[RFC2737] McCloghrie, K. and A. Bierman, "Entity MIB (Version 2)", RFC 2737, December 1999.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

8. Informative References

[RFC2819] Waldbusser, S., "Remote network Monitoring Management Information Base", RFC 2819, May 2000.

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

9. Security Considerations

There is one managed object in this MIB that may contain sensitive information. This is:

entPhySensorValue

This object may expose the values of particular physical sensors for a device.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), 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.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model STD 62, RFC 3414 [RFC3414] and the View-based Access Control Model STD 62, RFC 3415 [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to only the objects, and those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

10. Authors' Addresses

Andy Bierman
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA USA 95134
Phone: +1 408-527-3711
EMail: abierman@cisco.com

Dan Romascanu
Avaya Inc.
Atidim Technology Park, Bldg. #3
Tel Aviv, 61131, Israel
Phone: +972-3-545-8414
EMail: dromasca@avaya.com

K.C. Norseth
L-3 Communications
640 N. 2200 West.
Salt Lake City, Utah 84116-0850
Phone: +1 801-594-2809
EMail: kenyon.c.norseth@L-3com.com

11. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.