Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [RFC2026].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Distribution of this document is unlimited. Please send comments to authors.

1. Copyright Notice
2. 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 [RFC2737] 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).

3. Table of Contents

1 Copyright Notice ................................................ 2
2 Abstract ........................................................ 2
3 Table of Contents ............................................... 2
4 The SNMP Network Management Framework ........................ 2
5 Overview ........................................................ 3
5.1 Terms ........................................................ 4
5.2 Relationship to the Entity MIB ................................. 4
5.3 Relationship to General Thresholding Mechanisms ............ 4
6 MIB Structure ................................................... 4
7 Definitions ....................................................... 5
8 Intellectual Property ............................................ 16
9 Acknowledgements ............................................... 16
10 References ...................................................... 16
11 Security Considerations ........................................ 19
12 Authors’ Addresses ............................................. 19
13 Full Copyright Statement ...................................... 20

4. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [RFC1155], RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in RFC 2578 [RFC2578], RFC 2579 [RFC2579] and RFC 2580 [RFC2580].
o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].

o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].

o A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

5. 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) augments the entPhysicalTable, but only for entries which represent physical sensors.

5.1. 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 RFC 2119. [RFC2119]

5.2. Relationship to the Entity MIB

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

5.3. 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.

6. MIB Structure

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

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

entSensorType
 This object identifies the type of data units associated with the sensor value. OPEN ISSUE: The current version of the MIB defines this object as an enum. This limits the scalability to a pre-defined set of values. A syntax of OID with a IANA-administered list of entSensorTypes would allow for the list to be extended with new types of sensors that might show up in the future. The Working
Group should decide if this extensibility is worth the price of the relative complexity of using an OID syntax. Also, an OID syntax may potentially be abused, since this syntax does not restrict the list of OID values to IANA-administered values.

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

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

entSensorValue
This object identifies the current value of the sensor.

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

entSensorUnitsDisplay
This object provides a textual description of the data units represented by the entSensorType and entSensorScale objects.

entSensorValueTimeStamp
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 entSensorValueUpdateRate object is greater than zero).

entSensorValueUpdateRate
This object indicates the nature of the agent implementation of the entSensorEntry, 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).

7. Definitions

ENTITY-SENSOR-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
entitySensorMIB MODULE-IDENTITY
LAST-UPDATED    "200201180000Z"
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 Communication
  Tel: +972-3-645-8414
  Email: dromasca@avaya.com
  Postal: Atidim technology Park, Bldg. #3
          Tel Aviv, Israel, 61131

  KC Norseth
  Enterasys Networks
  Tel: +1 801 887 9823
  Email: knorseth@enterasys.com
  Postal: 2691 South Decker Lake Lane
          Salt Lake City, Utah 84119

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."
REVISION        "2002011800000Z"
DESCRIPTION
"Initial version of the Entity Sensor MIB module."
::= ( mib-2 xxx ) -- unassigned

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 "TBD"
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

Expires July 2002
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
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)
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
--

entSensorTable OBJECT-TYPE
SYNTAX        SEQUENCE OF EntSensorEntry
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 }

entSensorEntry OBJECT-TYPE
SYNTAX        EntSensorEntry
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
::= { entSensorTable 1 }

EntSensorEntry ::= SEQUENCE {
   entSensorType            EntitySensorDataType,
   entSensorScale           EntitySensorDataScale,
   entSensorPrecision       EntitySensorPrecision,
   entSensorValue           EntitySensorValue,
   entSensorStatus          EntitySensorStatus,
   entSensorUnitsDisplay    SnmpAdminString,
   entSensorValueTimeStamp  TimeStamp,
   entSensorValueUpdateRate Unsigned32

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

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

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

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

entSensorPrecision 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 entSensorValue object.

This object SHOULD be set to '0' when the associated entSensorType 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."

::= { entSensorEntry 3 }

entSensorValue 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 entSensorType, entSensorScale, and entSensorPrecision objects must also be examined."

::= { entSensorEntry 4 }

entSensorStatus OBJECT-TYPE
SYNTAX EntitySensorStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The operational status of the sensor."

::= { entSensorEntry 5 }

entSensorUnitsDisplay 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 entSensorValue."

::= { entSensorEntry 6 }

entSensorValueTimeStamp 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."
 ::=  { entSensorEntry 7 }

entSensorValueUpdateRate  OBJECT-TYPE
SYNTAX       Unsigned32
UNITS        "milliseconds"
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION
  "An indication of the frequency that the agent updates the
  associated entSensorValue 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.

 ::=  { entSensorEntry 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 }

Expires July 2002

[Page 14]
MODULE ENTITY-MIB
  MANDATORY-GROUPS { entityPhysicalGroup }

::= { entitySensorCompliances 1 }

-- Object Groups

entitySensorValueGroup OBJECT-GROUP
  OBJECTS {
    entSensorType,
    entSensorScale,
    entSensorPrecision,
    entSensorValue,
    entSensorStatus,
    entSensorUnitsDisplay,
    entSensorValueTimeStamp,
    entSensorValueUpdateRate
  }
  STATUS   current
  DESCRIPTION
    "A collection of objects representing physical entity sensor
    information."
 ::= { entitySensorGroups 1 }

END
8. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

9. 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.

10. References

[RFC1155]

[RFC1157]

[RFC1212]
[RFC1215]
M. Rose, "A Convention for Defining Traps for use with the SNMP",

[RFC1901]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S.
Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901,
SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting,
Inc., International Network Services, January 1996.

[RFC1905]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S.
Waldbusser, "Protocol Operations for Version 2 of the Simple
Network Management Protocol (SNMPv2)", RFC 1905, SNMP Research,
Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc.,
International Network Services, January 1996.

[RFC1906]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S.
Waldbusser, "Transport Mappings for Version 2 of the Simple Network
Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc., Cisco
Systems, Inc., Dover Beach Consulting, Inc., International Network
Services, January 1996.

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

[RFC2570]
Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to
Version 3 of the Internet-standard Network Management Framework",
RFC 2570, SNMP Research, Inc., TIS Labs at Network Associates,

[RFC2571]
Describing SNMP Management Frameworks", RFC 2571, Cabletron
Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April
1999.

[RFC2572]
Processing and Dispatching for the Simple Network Management
Protocol (SNMP)", RFC 2572, SNMP Research, Inc., Cabletron Systems,
[RFC2573]

[ RFC2574]

[ RFC2575]

[ RFC2578]

[ RFC2579]

[ RFC2580]

[ RFC2737]

[ RFC2819]
11.  Security Considerations

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

    entSensorValue

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), 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.

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 [RFC 2574] and the View-based Access Control Model [RFC 2575] 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 the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

12.  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

    KC Norseth
    Enterasys Networks
    2691 South Decker Lake Lane
    Salt Lake City, Utah 84119
13. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.