Entity MIB Working Group
Internet Draft
Document: draft-ietf-entmib-state-01.txt
Category: Standards Track

Expiration Date: December 2003

S. Chisholm
Nortel Networks
D. Perkins
SNMPinfo &
RiverStone Networks

June 2003

Entity State MIB

Status of this Memo

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 state of the entity.

Table of Contents

1. 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. Entity State

The goal in adding state objects to the Entity MIB [RFC2737] is to define a useful subset of the possible state attributes that could be tracked for a given entity that both fit into the existing IETF model, as well as leveraged existing well-deployed models. The entStateTable contains state objects that are a subset of the popular ISO/OSI states that are also defined in ITU's X.731 specification [X.731]. Objects are defined to capture administrative, operational and usage states. In addition there are further state objects defined to provide additional information for these three basic states.

Administrative state indicates permission to use or prohibition against using the entity and is imposed through the management services.

Operational state indicates whether or not the entity is physically installed and working. Note that unlike the ifOperStatus [RFC2863], this operational state is independent of the administrative state.

Usage state indicates whether or not the entity is in use at a specific instance, and if so, whether or not it currently has spare capacity to serve additional users. In the context of this MIB, the user is equivalent to an entity, so this term us substituted.

Alarm state indicates whether or not there are any alarms active against the entity. In addition to those alarm status defined in X.731 [X.731], warning and indeterminate status are also defined to provide a more complete mapping to the Alarm MIB [Alarm-MIB].

Standby state indicates whether the entity is currently running as hot standby, cold standby or is currently providing service.

The terms state and status are used interchangeably in this memo.

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].

2.1 Hierarchical State Management

Physical entities exist within a containment hierarchy. This raises some interesting issues not addresses in existing work on state management [X.731].

There are two types of state for an entity:

1) The state of the entity independent of the states of its parents and children in its containment hierarchy. This is often referred to as raw state.

2) The state of the entity, as it may be influenced by the state of its parents and children. This is often referred to as computed state.

All state objects in this memo are raw state.

2.2 State Relationships

The following section outlines all of the combinations of the three basic states -administrative, operational and usage - and briefly describes what each of these combinations of states means. It also compare this combination of states to that of the ifAdminStatus and ifOperStatus objects of the Interfaces Group MIB [RFC2863] to both provide insight to those familiar with these status objects as well as to clarify the relationship between entities and interfaces, as indicated by entAliasMappingIdentifier object in the Entity MIB [RFC2737].

The Interfaces MIB [RFC2863] defines the ifAdminStatus object, which has status of up, down and testing and the ifOperStatus object, which has states of up, down, testing, unknown, dormant, notPresent and lowerLayerDown.

2.2.1 Admin State Locked, Operational State Disabled and Usage State Idle

The entity is totally inoperable, it is not servicing any entities and it is also administratively prohibited from use. To make it available for use, both management permission and some corrective action are necessary. This is similar to an ifAdminStatus of down and ifOperStatus of down.

2.2.2 Admin State Locked, Operational State Enabled and Usage State Idle

The entity is partially or fully operable, it is not servicing any entities but is administratively prohibited from use. To make it available for use, only management permission is required. This is similar to an ifAdminStatus of down and ifOperStatus of down.

2.2.3 Admin State Shutting Down, Operational State Enabled and Usage State Active

The entity is partially or fully operable and in use, but usage is administratively limited to current instances of use. For an additional entity to gain access, management permission is required. Otherwise, when all current entities have been removed from the resource, the managed object will automatically transit to the locked, enabled, and idle state. This is similar to the situation described in [RFC2863] where ifAdminStatus transitions to down, but the ifOperStatus's transition does not occur immediately, but rather after a small time lag to complete certain operations before going "down".

2.2.4 Admin State Shutting Down, Operational State Enabled and Usage State Busy

The entity is partially or fully operable and in use, but usage is administratively limited to current instances of use. In addition, it has no spare capacity to provide for additional entities. For an additional entity to gain access, besides waiting for an existing entity to be removed, management permission is also required. Otherwise, when all current entities have been removed from the resource, the managed object will automatically transit to the locked, enabled, idle state. This is similar to the situation described in [RFC2863] where ifAdminStatus transitions to down, but the ifOperStatus's transition does not occur immediately, but rather after a small time lag to complete certain operations before going "down".

2.2.5 Admin State Unlocked, Operational State Enabled and Usage State Idle

The entity is partially or fully operable, it is not actually in use and is not administratively prohibited from use. This is similar to an ifAdminStatus of up and ifOperStatus of up if the interface is able to pass packets. If the interface is found to be operable, but the interface is waiting for other, external, events to occur before it can transmit or receive packets, then this is similar to an ifAdminStatus of up and a ifOperStatus of dormant.

2.2.6 Admin State Unlocked, Operational State Enabled and Usage State Active

The entity is partially or fully operable, it is currently in use and is not

administratively prohibited from use. It has sufficient spare capacity to provide for additional entities. This is similar to an

ifAdminStatus of up and ifOperStatus of up.

2.2.7 Admin State Unlocked, Operational State Enabled and Usage State Busy

The entity is partially or fully operable, it is currently in use and it is not administratively prohibited from use. Currently it has no spare capacity to provide for additional entities. For an additional entity to gain access, it is necessary to wait for an existing entity to be removed or for some capacity increase to occur. This is similar to an ifAdminStatus of up and ifOperStatus of up.

2.2.8 Admin State Unlocked, Operational State Disabled and Usage State Idle

The entity is totally inoperable, it is servicing no entities but it is not administratively prohibited from use. To make it available for use, some corrective action is required. This is similar to an ifAdminStatus of up and ifOperStatus of down. If the cause of the interface being down is because of a lower layer being down, then this is similar to an ifAdminStatus of up and an ifOperStatus of lowerLayerDown.

2.3 Physical Classes and States

This section provides an overview of applying the states for the basic physical classes as indicated by the entPhysicalClass object in the Entity MIB [RFC2737]. The physical classes are chassis, backplane, container, powerSupply, fan, sensor, module, port and stack. All states can, in theory, be implemented for any class, but some states or some values of states make less sense than others, depending on the physical class of the entity.

2.3.1 Chassis

2.3.1.1 entStateAdmin

A value of unlocked for entStateAdmin means that this system is on. A value of shuttingDown for entStateAdmin means that this system is in the process of shutting down.

2.3.1.2 entStateOper

A value of enabled for entStateOper indicates that basic functions of this system are functioning. A value of disabled for entStateOper indicates a problem with basic functions on the system.

2.3.1.3 entStateUsage

Many chassis will come either fully populated or fully populated

with empty container entities, which can be filled independently and therefore do not affect the entStateUsage of the chassis itself. In both these cases and in the general case where the chassis can't

support any more direct child entities, entStateUsage will have a value of busy. While an empty chassis might happen much in practice, in this case the entStateUsage object would have a value of idle. Likewise, if the chassis is partially used, then entStateUsage would have a value of active.

2.3.1.4 entStateStandby

A value of hotStandby for enStateStandby indicates that the entire system contained within this chassis is running as a hot standby for another complete system, possibly contained within the same stack. A value of coldStandby for enStateStandby indicates that the entire system contained within this chassis is running as a cold standby for another complete system, possibly contained within the same stack. A value of providingService for enStateStandby indicates that the entire system contained within this chassis is currently providing service.

2.3.1.5 entStateAlarm

If this chassis is not contained within a stack, the alarm counts indicated by entStateAlarm will be those alarms that are against the general system, as appose sub-components within the containment hierarchy.

2.3.2 BackPlane

2.3.2.1 entStateAdmin

A value of unlocked for entStateAdmin means that the backplane is not administratively prevented from aggregating and forwarding network traffic. A value of shutting down for entStateAdmin means that the backplane will finish aggregating and forwarding the network traffic is currently handling, but then transition to be administratively locked. A value of locked for entStateAdmin means that backplane is administratively prohibited from aggregating and forwarding any network traffic.

2.3.2.2 entStateOper

A value of enabled for entStateOper means that the backplane is partially or fully capable of aggregating and forwarding network traffic. A value of disabled for entStateOper means that the backplane is unable to aggregate and forward any network traffic.

2.3.2.3 entStateUsage

The entStateUsage for a backplane will busy.

2.3.2.4 entStateStandby

A value of hotStandby for enStateStandby indicates that the backplane is running as a hot standby for another backplane within this system. A value of coldStandby for enStateStandby indicates that the backplane is running as a cold standby for another backplane, possibly within this system. A value of providingService for enStateStandby indicates that the backplane is currently providing service.

2.3.2.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this backplane.

2.3.3 Container

2.3.3.1 entStatAdmin

A value of unlocked for entStateAdmin means it is administratively possible to insert things into this container. A value of shuttingDown for entStateAdmin could be used to reflect that inserting objects into this container is administratively prohibited. This value could also be used for systems that do not support hot insertion of components.

2.3.3.2 entStateOper

It may not make sense for the entStateOper to have values other than enabled.

2.3.3.3 entStateUsage

The container physical class could be used to indicate, among other things, chassis slots or daughter-card holders. If the container is empty, for example it has no modules in its slots, then entStateUsage would have a value of idle. If the container is partially used, for example it has modules in some but not all of its slots, then entStateUsage would have a value of active. If the container is full, for example it has no empty slots, then entStateUsage would have a value of busy.

3.3.3.4 entStateStandby

Looking at the entStateStandby indicates whether this container is currently providing service or acting as a backup for another container.

2.3.3.5 entStateAlarm

If it is not possible to raise alarms against this chassis, the entStateAlarm will have no alarms set.

2.3.4 PowerSupply

2.3.4.1 entStateAdmin

A value of locked for entStateAdmin means that the power supply has been turned off. This only makes sense in the situation where there is a backup power supply. A value of unlocked for entStateAdmin means that the power supply is turned on.

2.3.4.2 entStateOper

A value of enabled for entStateOper means that the power supply is operational. A value of disabled for entStateOper means that the power supply is not functioning.

2.3.4.3 entStateUsage

A value of idle for entStateUsage means that the power supply is providing no power to the system. A value of busy for entStateUsage means that the power supply is providing power to the system.

2.3.3.4 entStateStandby

If this power supply is the currently providing power to the system, then entStateStandby would have a value of providing service. If this power supply is serving as a backup to a primary power supply, then entStateStandby would have a value of hotstandby.

2.3.3.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this power supply.

2.3.5 Fan

2.3.5.1 entStateAdmin

Looking at the entStateAdmin and entStateOper provide useful information to determine why a fan is not running. A value of locked for entStateAdmin means that the fan is not running because it has been administratively disabled.

2.3.5.2 entStateOper

A value of disabled for the entOperStatus indicates that the fan itself is not operational. A value of enabled for the entOperStatus indicates that the fan is working in theory and that cause of it not operator may lie elsewhere.

2.3.5.3 entStateUsage

A value of busy for entStateUsage would indicate that the fan was running. A value of idle for entStateUsage would indicate that the fan was not actually running.

2.3.5.4 entStateStandby

If this fan is serving as a backup to a primary fan, then entStateStandby would have a value of hotstandby. If this fan is the currently providing service to the system, then entStateStandby would have a value of providing service.

2.3.5.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this fan.

2.3.6 Sensor

2.3.6.1 entStateAdmin

A value of unlocked for entStateAdmin indicates that the sensor is not administratively prohibited from sensing. A value of shutting down for entStateAdmin indicates that the sensor will complete its current readings and then shut down. A value of locked for entStateAdmin indicates that the sensor is administratively prohibited from sensing.

2.3.6.2 entStateOper

A value of enabled for entStateOper indicates that the sensor is functioning properly. A value of disable for entStateOper indicates that the sensor is totally inoperable.

2.3.6.3 entStateUsage

The value of entStateUsage will be busy.

2.3.6.4 entStateStandby

Looking at the entStateStandby indicates whether this sensor is currently providing service or acting as a backup for another sensor.

2.3.6.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this sensor.

2.3.7 Module

2.3.7.1 entStateAdmin

For modules that support the functionality of being administratively disabled, entStateAdmin object indicates whether the module is administratively locked (disabled) or unlocked (enabled). Modules that do not support disabling will always have a value of unlocked for entStateAdmin.

2.3.7.2 entStateOper

A value of enabled for entStateOper indicates that this module is partially or fully operational. A value of disabled for entStateOper indicates that this module is totally inoperable.

2.3.7.3 entStateUsage

If a module comes fully populated with all possible child components, or if there is not further room for child components, then the value of entStateUsage will be busy. If it supports the concept of dynamically added child components and there is room to add more components, then entStateUsage will have a value of active. If it supports the concept of dynamically added child components and there are currently no children then entStateUsage will have a value of idle.

2.3.7.4 entStateStandby

Looking at the entStateStandby indicates whether this module is currently providing service or acting as a backup for another module.

2.3.7.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this module.

2.3.8 Port

2.3.8.1 entStateAdmin

A value of enabled for entStateAdmin means the port is not administratively prohibited from passing network traffic. A value of shutting down for entStateAdmin indicates that the port will pass its current traffic and then transition to the locked state. A value of locked for entStateAdmin indicates that the port is administratively prohibited from passing network traffic.

2.3.8.2 entStateOper

A value of enabled for entStateOper means that the port is partially or fully capable of forwarding network traffic. A value of disabled for entStateOper means that the port is totally unable to forward network traffic.

2.2.8.3 entStateUsage

A value of idle for entStateUsage indicates that the port is not currently in use. A value of busy for entStateUsage indicates that the port is in use.

2.3.8.4 entStateStandby

Looking at the entStateStandby indicates whether this port is currently providing service or acting as a backup for another port.

2.3.8.5 entStateAlarm

Looking at the entStateAlarm gives a convenient way to see if there are any alarms currently active against this port.

2.3.9 Stack

2.3.9.1 entStateAdmin

A value of unlocked for entStateAdmin means that this system is on. A value of shuttingDown for entStateAdmin means that this system is in the process of shutting down.

2.3.9.2 entStateOper

A value of enabled for entStateOper indicates that basic functions of this system are functioning. A value of disabled for entStateOper indicates a problem with basic functions on the system.

2.3.9.3 entStateUsage

If a stack comes fully populated with all possible child components, or if there is not further room for child components, then the value of entStateUsage will be busy. If it supports the concept of dynamically added child components and there is room to add more components, then entStateUsage will have a value of active. If it supports the concept of dynamically added child components and there are currently no children then entStateUsage will have a value of idle.

2.3.9.4 entStateStandby

A value of hotStandby for enStateStandby indicates that the entire system contained within this stack is running as a hot standby for

another complete system, possibly contained within the same parent stack. A value of coldStandby for enStateStandby indicates that the entire system contained within this stack is running as a cold standby for another complete system, possibly contained within the same parent stack. A value of providingService for enStateStandby indicates that the entire system contained within this chassis is currently providing service.

2.3.9.5 entStateAlarm

If this stack is not contained in within a parent stack, the alarm counts indicated by entStateAlarm will be those alarms that are against the general system, as appose sub-components within the containment hierarchy.

2.4 Relation to Alarm MIB

The entStateAlarm object indicates whether or not there are any active alarms against this entity. If there are active alarms, then the alarmActiveTable in the Alarm MIB [Alarm MIB] should be searched for alarmActiveResourceId that match this entPhysicalIndex.

2.5 Entity Redundancy

While this memo is not attempting to address the entire problem space around redundancy, the entStateStandby object provides an important piece of state information for entities, which helps identify which pieces of redundant equipment are currently providing service, and which are waiting in either hot or cold standby mode.

3. Definitions

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

  IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, mib-2
          FROM SNMPv2-SMI
      TEXTUAL-CONVENTION, DateAndTime
          FROM SNMPv2-TC
      MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
           FROM SNMPv2-CONF
      entPhysicalIndex
           FROM ENTITY-MIB;

  entityStateMIB MODULE-IDENTITY
      LAST-UPDATED "200306300000Z"
      ORGANIZATION "IETF Entity MIB Working Group"
      CONTACT-INFO
```

```
" General Discussion: entmib@ietf.org
 To Subscribe:
   http://www.ietf.org/mailman/listinfo/entmib
```

http://www.ietf.org/html.charters/entmib-charter.html

Sharon Chisholm
Nortel Networks
PO Box 3511 Station C
Ottawa, Ont. K1Y 4H7
Canada
schishol@nortelnetworks.com

David T. Perkins
Riverstone Networks
5200 Great America Parkway
Santa Clara, CA 95054
USA
dperkins@snmpinfo.com

```
            "
        DESCRIPTION
                "This MIB defines a state extension to the entity MIB.

               Copyright (C) The Internet Society 2003.  This version
               of this MIB module is part of RFC yyyy;  see the RFC
               itself for full legal notices."
          -- RFC Ed.: replace yyyy with actual RFC number & remove
          -- this note
        REVISION    "200306300000Z"
        DESCRIPTION
            "Initial version, published as RFC YYYY."
           -- RFC-Editor assigns yyyy
        ::= { mib-2 XX }  -- to be assigned by IANA


-- Textual conventions


 AdminState  ::=  TEXTUAL-CONVENTION
   STATUS         current
   DESCRIPTION
        " Represents the various possible administrative states.

          A value of locked means the resource is administratively
          prohibited from use. A value of shuttingDown means that
          usage is administratively limited to current instances of
          use. A value of unlocked means the resource is not
          administratively prohibited from use."
   REFERENCE
         "ITU Recommendation X.731, 'Information Technology - Open
             Systems Interconnection - System Management: State
             Management Function', 1992"
```

```
        SYNTAX          INTEGER
                        {
                        notSupported(1),
```

```
            locked(2),
            shuttingDown(3),
            unlocked(4)
            }

 OperState  ::=  TEXTUAL-CONVENTION
   STATUS         current
   DESCRIPTION
        " Represents the possible values of operational states.

          A value of disabled means the resource is totally
          inoperable. A value of enabled means the resource
          is partially or fully operable."
   REFERENCE
         "ITU Recommendation X.731, 'Information Technology - Open
            Systems Interconnection - System Management: State
            Management Function', 1992"
   SYNTAX         INTEGER
            {
            notSupported (1),
            disabled(2),
            enabled(3)
            }

 UsageState  ::=  TEXTUAL-CONVENTION
   STATUS         current
   DESCRIPTION
        " Represents the possible values of usage states.
          A value of idle means the resource is servicing no users.
          A value of active means the resource is currently in use
          and it has sufficient spare capacity to provide for
          additional users. A value of busy means the resource is
          currently in use, but it currently has no spare capacity
          to provide for additional users."
   REFERENCE
         "ITU Recommendation X.731, 'Information Technology - Open
            Systems Interconnection - System Management: State
            Management Function', 1992"
   SYNTAX         INTEGER
            {
            notSupported (1),
            idle(2),
            active(3),
            busy(4)
            }


AlarmStatus  ::=  TEXTUAL-CONVENTION
```

```
        STATUS         current
        DESCRIPTION
             "Represents the possible values of alarm status.
```

```
        When no bits of this attribute are set, then none of the
        status conditions described below are present. When the
        value of under repair is set, the resource is currently
        being repaired.

        When the value of critical is set, one or more critical
        alarms are active against the resource. When the value of
        major is set, one or more major alarms are active against
        the resource. When the value of minor is set, one or more
        minor alarms are active against the resource. When the
        value of warning is set, one or more warning alarms are
        active against the resource. When the value of
        indeterminate is set, one or more  alarms of indeterminate
        severity are active against the resource.

        When the value of alarm outstanding is set, one or more
        alarms is active against the resource. The fault may or may
        not be disabling. "
  REFERENCE
        "ITU Recommendation X.731, 'Information Technology - Open
            Systems Interconnection - System Management: State
            Management Function', 1992"
        SYNTAX         BITS
           {
           notSupported (0),
           underRepair(1),
           critical(2),
           major(3),
           minor(4),
           alarmOutstanding(5),
           warning (6), -- Not defined in X.731
           indeterminate (7) -- Not defined in X.731
           }


StandbyStatus  ::=  TEXTUAL-CONVENTION
  STATUS         current
  DESCRIPTION
       " Represents the possible values of standby status.

         A value of hotStandby means the resource is not providing
         service, but is will be immediately able to take over the
         role of the resource to be backed-up, without the need for
         initialization activity, and will contain the same
         information as the resource to be backed up. A value of
         coldStandy means that the resource is to back-up another
         resource, but will not be immediately able to take over
         the role of a resource to be backed up, and will require
```

```
            some initialization activity. A value of providingService
            means the resource is providing service."
        REFERENCE
              "ITU Recommendation X.731, 'Information Technology - Open
```

```
            Systems Interconnection - System Management: State
            Management Function', 1992"
        SYNTAX         INTEGER
          {
          notSupported (1),
          hotStandby(2),
          coldStandby(3),
          providingService(4)
          }

-- Entity State Objects

entStateObjects OBJECT IDENTIFIER ::= { entityStateMIB 1 }

entStateTable OBJECT-TYPE
 SYNTAX      SEQUENCE OF EntStateEntry
 MAX-ACCESS  not-accessible
 STATUS      current
 DESCRIPTION
     "A table of information about state/status of entities.
     This is a sparse augment of the entPhysicalTable.
      "
 ::= { entStateObjects 1 }


  entStateEntry OBJECT-TYPE
     SYNTAX      EntStateEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION "State information about this entity."
     INDEX       { entPhysicalIndex }
     ::= { entStateTable 1 }

  EntStateEntry ::= SEQUENCE {
      entStateLastChanged DateAndTime,
      entStateAdmin       AdminState,
      entStateOper        OperState,
      entStateUsage       UsageState,
      entStateAlarm       AlarmStatus,
      entStateStandby     StandbyStatus
     }

entStateLastChanged OBJECT-TYPE
 SYNTAX      DateAndTime
 MAX-ACCESS  read-only
 STATUS      current
 DESCRIPTION "The value of this object is the date and
             time when state/status of the component
```

```
                last changed, or zero."
      ::= { entStateEntry 1 }

   entStateAdmin OBJECT-TYPE
```

```
       SYNTAX      AdminState
       MAX-ACCESS  read-write
       STATUS      current
       DESCRIPTION
            "The administrative state for this entity."
       ::= { entStateEntry 2 }

 entStateOper OBJECT-TYPE
       SYNTAX      OperState
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
           "The operational state for this entity."
       ::= { entStateEntry 3 }

 entStateUsage OBJECT-TYPE
       SYNTAX      UsageState
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
            "The usage state for this entity."
       ::= { entStateEntry 4 }

 entStateAlarm OBJECT-TYPE
       SYNTAX      AlarmStatus
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
            "The alarm status for this entity. It does not include
            the severity of alarms raised on child components."
       ::= { entStateEntry 5 }

entStateStandby OBJECT-TYPE
       SYNTAX StandbyStatus
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
         "The standby status for this entity."
  ::= { entStateEntry 6 }

-- Notifications

entStateTraps      OBJECT IDENTIFIER ::= { entityStateMIB 2 }
entStateTrapPrefix OBJECT IDENTIFIER ::= { entStateTraps 0 }

entStateOperEnabled NOTIFICATION-TYPE
   OBJECTS { entStateAdmin,
             entStateAlarm
```

```
            }
        STATUS              current
        DESCRIPTION
                "The entity is operational. The entity this
```

```
            notification refers can be identified by
            extracting the entPhysicalIndex from one of the
            variable bindings."
   ::= { entStateTrapPrefix 1 }

entStateOperDisabled NOTIFICATION-TYPE
   OBJECTS { entStateAdmin,
             entStateAlarm }
   STATUS             current
   DESCRIPTION
            "The entity is not operational. The entity this
            notification refers can be identified by
            extracting the entPhysicalIndex from one of the
            variable bindings."
   ::= { entStateTrapPrefix 2 }

-- Conformance and Compliance

entStateConformance OBJECT IDENTIFIER ::= { entityStateMIB 3 }

entStateCompliances OBJECT IDENTIFIER
                  ::= { entStateConformance 1 }

entStateCompliance MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
          "The compliance statement for systems supporting
          the Entity State MIB."
      MODULE -- this module
          MANDATORY-GROUPS {
           entStateGroup
          }
      OBJECT entStateAdmin
       MIN-ACCESS  read-only
       DESCRIPTION
           "Write access is not required."
   ::= { entStateCompliances 1 }

entStateGroups OBJECT IDENTIFIER ::= { entStateConformance 2 }

entStateGroup OBJECT-GROUP
   OBJECTS {
            entStateLastChanged,
            entStateAdmin,
            entStateOper,
            entStateUsage,
            entStateAlarm,
            entStateStandby
```

```
        }
    STATUS   current
    DESCRIPTION
         "Standard Entity State group."
```

```
    ::= { entStateGroups 1}

entStateNotificationGroup NOTIFICATION-GROUP
   NOTIFICATIONS {
           entStateOperEnabled,
           entStateOperDisabled
           }
    STATUS   current
    DESCRIPTION
         "Standard Entity State Notification group."
    ::= { entStateGroups 2}

END
```

4. Security Considerations

There is one management object defined in this MIB that has a MAX-ACCESS clause of read-write. The object may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The following object is defined with a MAX-ACCESS clause of read-write: entStateAdmin.

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 (entities) that have legitimate rights to indeed GET or SET (change/create/delete) them.

5. Authors' Addresses

Sharon Chisholm
Nortel Networks
PO Box 3511, Station C

Ottawa, Ontario, K1Y 4H7
Canada
Email: schishol@nortelnetworks.com

David T. Perkins
Riverstone Networks
5200 Great America Parkway
Santa Clara, CA 95054
USA
Email: dperkins@snmpinfo.com

6. Acknowledgments

This document is a product of the Entity MIB Working Group.

7. References

7.1 Normative

[ALARM-MIB] Chisholm, S., Romascanu, D., "Alarm MIB", draft-ietf-disman-alarm-mib-10.txt, April 2003

[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., Bierman, A., "Entity MIB (Version 2)", December 1999.

[X.731] ITU Recommendation X.731, "Information Technology - Open Systems Interconnection - System Management: State Management Function", 1992

7.2 Informative References

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

8. Full Copyright Statement