

Entity MIB Working Group
Internet Draft
Document: [draft-ietf-entmib-state-00.txt](#)
Category: Standards Track

Expiration Date: July 2003

S. Chisholm
Nortel Networks
D. Perkins
SNMPinfo &
RiverStone Networks
January 2003

Entity State MIB

Status of this Memo

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

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. The SNMP Management Framework
2. Entity State
 - 2.1. State Relationships
 - 2.2. Physical Classes and State
 - 2.3. Relation to Alarm MIB
3. Definitions

- 4. Security Considerations
- 5. Authors' Addresses
- 6. Acknowledgements

Chisholm & Perkins

Standards Track
Entity State MIB

[Page 1]ü
January 2003

- 7. References
- 8. Full Copyright Statement

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in [RFC 2571](#) [[RFC2571](#)].
- o 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 STD 16, [RFC 1155](#) [[RFC1155](#)], STD 16, [RFC 1212](#) [[RFC1212](#)] and [RFC 1215](#) [[RFC1215](#)]. The second version, called SMIV2, is described in STD 58, [RFC 2578](#) [[RFC2578](#)], STD 58, [RFC 2579](#) [[RFC2579](#)] and STD 58, [RFC 2580](#) [[RFC2580](#)].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, [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 STD 15, [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.

Chisholm & Perkins

Standards Track
Entity State MIB

[Page 3]ü
January 2003

2. Entity State

The goal in adding state objects to the Entity MIB was 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. 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. The administrative state defined for an entity is independent of administrative states in its containment hierarchy. This means that administratively locking an entity does not automatically lock its children in the containment hierarchy.

Operational state indicates whether or not the entity is physically installed and working. The operational state defined for an entity is indirectly dependent on the operational state of the entities in which it is contained. If its parent entities in its containment hierarchy are disabled, and therefore totally inoperable, then it is unlikely that the given entity will be operable.

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.

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](#).

2.1 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.

Chisholm & Perkins

Standards Track
Entity State MIB

[Page 4]ü
January 2003

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

The entity is totally inoperable, it is not servicing any users 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.

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

The entity is partially or fully operable, it is not servicing any users 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.ü

2.1.3 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 user to gain access, management permission is required. Otherwise, when all current users have terminated their use of 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".ü

2.1.4 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 users. For an additional user to gain access, besides waiting for an existing user to terminate, management permission is also required. Otherwise, when all current users have terminated their use of 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".

2.1.5 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.

2.1.6 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 users. This is similar to an ifAdminStatus of up and ifOperStatus of up.

2.1.7 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 users. For an additional user to gain access, it is necessary to wait for an existing user to terminate or for some capacity increase to occur. This is similar to an ifAdminStatus of up and ifOperStatus of up.

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

The entity is totally inoperable, it is servicing no users 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.

2.2 Physical Classes and States

2.2.1 Chassis

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. 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.

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.

If this chassis is not contained in 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.

2.2.2 BackPlane

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. 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.

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

2.2.3 Container

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.

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 busy. If the container is full, for example it has no empty slots, then entStateUsage would have a value of busy.

If it is not possible to raise alarms against this chassis, the entStateAlarm will have no alarms set. It may not make sense for the entStateOper to have values other than enabled.

2.2.4 PowerSupply

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.

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

2.2.5 Fan

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

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

2.2.6 Sensor

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. 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.

Looking at the entStateStandby indicates whether this sensor is

currently providing service or acting as a backup for another sensor. Looking at the entStateAlarm gives a convenient way to see

if there are any alarms currently active against this sensor.

2.2.7 Module

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`. 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. A value of idle for `entStateUsage` indicates that this module is currently not performing any functions. A value of active `entStateUsage` indicates that this module is currently performing functions, but capable of performing more. A value of busy for `entStateUsage` indicates that the module is functioning at full capacity and unable to perform further functions at this current time.

Looking at the `entStateStandby` indicates whether this module is currently providing service or acting as a backup for another module. Looking at the `entStateAlarm` gives a convenient way to see if there are any alarms currently active against this module.

2.2.8 Port

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. 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. 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.

Looking at the `entStateStandby` indicates whether this port is currently providing service or acting as a backup for another port. Looking at the `entStateAlarm` gives a convenient way to see if there are any alarms currently active against this port.

2.2.9 Stack

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. 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.

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.

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.

2.3 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 should be searched for alarmActiveResourceId that match this entPhysicalIndex.

3. Definitions

ENTITY-STATE-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

entityStateMIB MODULE-IDENTITY

LAST-UPDATED "200301310000Z"

ORGANIZATION "IETF Entity MIB Working Group"

CONTACT-INFO

" Sharon Chisholm
Nortel Networks
PO Box 3511 Station C
Ottawa, Ont. K1Y 4H7

Canada
schishol@nortelnetworks.com

David T. Perkins

Chisholm & Perkins

Standards Track
Entity State MIB

[Page 10]ü
January 2003

dperkins@dperkins.com

"

DESCRIPTION

"This MIB defines a state extension to the entity MIB "

REVISION "200301310000Z"

DESCRIPTION

"Initial version, published as RFC XXXX."

::= { mib-2 XX }

-- Textual conventions

AdminState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the various possible administrative states
(ITU-T X.731).

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."

SYNTAX INTEGER

{
locked(1),
shuttingDown(2),
unlocked(3)
}

OperState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the possible values of operational states
(ITU-T X.731).

A value of disabled means the resource is totally
inoperable. A value of enabled means the resource
is partially or fully operable."

SYNTAX INTEGER

{

```
disabled(1),
enabled(2)
}
```

UsageState ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the possible values of usage states
(ITU-T X.731).

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."

SYNTAX INTEGER

```
{
idle(1),
active(2),
busy(3)
}
```

AlarmStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the possible values of alarm status
(ITU-T X.731).

When no values 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 alarm outstanding is set, one or more
alarms is active against the resource. The fault may or may
not be disabling. "

SYNTAX BITS

```
{
```

```

    underRepair(0),
    critical(1),
    major(2),
    minor(3),
    alarmOutstanding(4),
    warning (5), -- Not defined in X.731
    indeterminate (6) -- Not defined in X.731
}

```

```

StandbyStatus ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        " Represents the possible values of standby status
        (IU-T X.731).

```

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 coldStandby 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.

"

```

SYNTAX      INTEGER
{
    hotStandby(1),
    coldStandby(2),
    providingService(3)
}

```

-- 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.

```

```

        "

```

```
::= { 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 {
```

```
    entStateAdmin      AdminState,
```

```
    entStateOper       OperState,
```

```
    entStateUsage      UsageState,
```

```
    entStateAlarm      AlarmStatus,
```

```
    entStateStandby    StandbyStatus
```

```
}
```

```
entStateAdmin OBJECT-TYPE
```

```
SYNTAX      AdminState
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The administrative state for this entity."
```

```
::= { entStateEntry 1 }
```

```
entStateOper OBJECT-TYPE
```

```
SYNTAX      OperState
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The operational state for this entity."
```

```
::= { entStateEntry 2 }
```

```
entStateUsage OBJECT-TYPE
```

```
SYNTAX      UsageState
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
    "The usage state for this entity."
```

```
::= { entStateEntry 3 }
```

```
entStateAlarm OBJECT-TYPE
```

```
SYNTAX      AlarmStatus
```

```
MAX-ACCESS  read-only
```

```

STATUS      current
DESCRIPTION
    "The alarm state for this entity. It does not include
    the severity of alarms raised on child components."
::= { entStateEntry 4 }

```

```

entStateStandby OBJECT-TYPE
    SYNTAX StandbyStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The standby status for this entity."
    ::= { entStateEntry 5 }

```

-- 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 {
        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

ü

[4. Security Considerations](#)

There is one management objects defined in this MIB that has a MAX-ACCESS clause of read-write. Such objects 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.

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 implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 2574](#) [[RFC2574](#)] and the View-based Access Control Model [RFC 2575](#) [[RFC2575](#)] 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.

5. Authors' Addresses

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

David T. Perkins
Email: dperkins@dperkins.com

6. Acknowledgments

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

7. References

- [ALARM-MIB] Chisholm, S., Romascanu, D., "Alarm MIB",
[draft-ietf-disman-alarm-mib-09.txt](#), September 2002
- [RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification
of Management Information for TCP/IP-based Internets", STD
16, [RFC 1155](#), May 1990.
- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin,
"Simple Network Management Protocol", STD 15, [RFC 1157](#),
May 1990.
- [RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions",
STD 16, [RFC 1212](#), March 1991.
- [RFC1215] M. Rose, "A Convention for Defining Traps for use with the
SNMP", [RFC 1215](#), March 1991.
- [RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
"Introduction to Community-based SNMPv2", [RFC 1901](#),
January 1996.
- [RFC1905] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
"Protocol Operations for Version 2 of the Simple Network
Management Protocol (SNMPv2)", [RFC 1905](#), January 1996.
- [RFC1906] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
"Transport Mappings for Version 2 of the Simple Network
Management Protocol (SNMPv2)", [RFC 1906](#), January 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", [BCP 14](#), [RFC 2119](#), 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](#), April 1999.
- [RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An
Architecture for Describing SNMP Management Frameworks",
[RFC 2571](#), April 1999.

- [RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen,

"Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", [RFC 2572](#), April 1999.

- [RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", [RFC 2573](#), April 1999.
- [RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", [RFC 2574](#), April 1999.
- [RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", [RFC 2575](#), April 1999.
- [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.
- [X.731] ITU Recommendation X.731, "Information Technology - Open Systems Interconnection - System Management: State Management Function", 1992

8. Full Copyright Statement

Copyright (C) The Internet Society (2003). 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.

