

Entity State MIB

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

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 physical entities.

In addition, this memo defines a set of Textual Conventions to represent various states of an entity. The intent is that these Textual Conventions will be imported and used in MIB modules that would otherwise define their own representations.

Table of Contents

1. The Internet-Standard Management Framework	2
2. Entity State	2
2.1. Hierarchical State Management	3
2.2. Entity Redundancy	3
2.3. Physical Entity Users	3
2.4. Physical Class Behavior	4
3. Relation to Other MIBs	4
3.1. Relation to the Interfaces MIB	4
3.2. Relation to Alarm MIB	5
3.3. Relation to Bridge MIB	5
3.4. Relation to the Host Resources MIB	5
4. Textual Conventions	6
5. Definitions	9

6. Security Considerations	16
7. Acknowledgements	17
8. References	17
8.1. Normative References	17
8.2. Informative References	18

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 [RFC4133] is to define a useful subset of the possible state attributes that could be tracked for a given entity and that both fit into the state models such as those used in the Interfaces MIB [RFC2863] as well as leverage 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 more 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 usage state refers to the ability of an entity to service other entities within its containment hierarchy.

Alarm state indicates whether or not there are any alarms active against the entity. In addition to those alarm states defined in X.731 [X.731], warning and indeterminate status are also defined to provide a more complete mapping to the Alarm MIB [RFC3877].

Standby state indicates whether the entity is currently running as hot standby or 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. Physical containment is defined by the entPhysicalContainedIn object [RFC4133]. This raises some interesting issues not addressed in existing work on state management.

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. 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.

2.3. Physical Entity Users

There are three ways to define the 'user' of a physical entity

1. Direct containment in physical hierarchy
2. Anywhere in physical hierarchy

3. As defined by a means outside the scope of this MIB. This could include logical interfaces that could run on a port, software that could run on a module, etc.

Administrative, operational, alarm, and standby state use all three definitions of 'user'. Usage state supports only the concept of direct containment to simplify implementations of this object.

2.4. Physical Class Behavior

This MIB makes no effort to standardize the behaviors and characteristics of the various physical classes [RFC4133], but rather how this information is reported. In looking at real-world products, items within the same physical class vary substantially. The MIB has therefore provided guidance on how to support objects where a particular instance of a physical class cannot support part or all of a particular state.

3. Relation to Other MIBs

3.1. Relation to the Interfaces MIB

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

An ifAdminStatus of 'up' is equivalent to setting the entStateAdmin object to 'unlocked'. An ifAdminStatus of 'down' is equivalent to setting the entStateAdmin object to either 'locked' or 'shuttingDown', depending on a system's interpretation of 'down'.

An ifOperStatus of 'up' is equivalent to an entStateOper value of 'enabled'. An ifOperStatus of 'down' due to operational failure is equivalent to an entStateOper value of 'disabled'. An ifOperStatus of 'down' due to being administratively disabled is equivalent to an entStateAdmin value of 'locked' and an entStateOper value of either 'enabled' or 'disabled' depending on whether there are any known issues that would prevent the entity from becoming operational when its entStateAdmin is set to 'unlocked'. An ifOperStatus of 'unknown' is equivalent to an entStateOper value of 'unknown'. The ifOperStatus values of 'testing' and 'dormant' are not explicitly supported by this MIB, but the state objects will be able to reflect other aspects of the entities' administrative and operational state. The ifOperStatus values of 'notPresent' and 'lowerLayerDown' are in some ways computed states and so are therefore not supported in this

MIB. They can, though, be computed by examining the states of entities within this object's containment hierarchy and other available related states.

3.2. 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 [RFC3877] should be searched for rows whose `alarmActiveResourceId` matches this `entPhysicalIndex`.

Alternatively, if the `alarmActiveTable` is queried first and an active alarm with a value of `alarmActiveResourceId` that matches this `entPhysicalIndex` is found, then `entStateAlarm` can be used to quickly determine if there are additional active alarms with a different severity against this physical entity.

3.3 Relation to Bridge MIB

For entities of physical type of 'port' that support the `dot1dStpPortEnable` object in the Bridge MIB [RFC4188], a value of 'enabled' is equivalent to setting the `entStateAdmin` object to 'unlocked'. Setting `dot1dStpPortEnable` to 'disabled' is equivalent to setting the `entStateAdmin` object to 'locked'.

3.4 Relation to the Host Resources MIB

The `hrDeviceStatus` object in the Host Resources MIB [RFC2790] provides an operational state for devices. For entities that logically correspond to the concept of a device, a value of 'unknown' for `hrDeviceStatus` corresponds to an `entStateOper` value of 'unknown'. A value of 'running' corresponds to an `entStateOper` value of 'enabled'. A value of 'warning' also corresponds to an `entStateOper` value of 'enabled', but with appropriate bits set in the `entStateAlarm` object to indicate the alarms corresponding to the unusual error condition detected. A value of 'testing' or 'down' is equivalent to an `entStateOper` value of 'disabled'.

4. Textual Conventions

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

IMPORTS

MODULE-IDENTITY, mib-2 FROM SNMPv2-SMI
TEXTUAL-CONVENTION FROM SNMPv2-TC;

entityStateTc MODULE-IDENTITY

LAST-UPDATED "200511220000Z"

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@nortel.com

David T. Perkins

548 Qualbrook Ct

San Jose, CA 95110

USA

Phone: 408 394-8702

dperkins@snmpinfo.com"

DESCRIPTION

"This MIB defines state textual conventions.

Copyright (C) The Internet Society 2005. This version
of this MIB module is part of RFC 4268; see the RFC
itself for full legal notices."

REVISION "200511220000Z"

DESCRIPTION

"Initial version, published as RFC 4268."

::= { mib-2 130 }

EntityAdminState ::= 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. A value of 'unknown' means that this resource is unable to report administrative state."

```
SYNTAX          INTEGER
{
    unknown (1),
    locked (2),
    shuttingDown (3),
    unlocked (4)
}
```

EntityOperState ::= 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. A value of 'testing' means the resource is currently being tested and cannot therefore report whether it is operational or not. A value of 'unknown' means that this resource is unable to report operational state."

```
SYNTAX          INTEGER
{
    unknown (1),
    disabled (2),
    enabled (3),
    testing (4)
}
```

EntityUsageState ::= 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. A value of 'unknown' means that this resource is unable to report usage state."

```
SYNTAX          INTEGER
```

```

{
  unknown (1),
  idle (2),
  active (3),
  busy (4)
}

```

EntityAlarmStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the possible values of alarm status.
An Alarm [RFC3877] is a persistent indication
of an error or warning condition.

When no bits of this attribute are set, then no active
alarms are known against this entity and it is not under
repair.

When the 'value of underRepair' is set, the resource is
currently being repaired, which, depending on the
implementation, may make the other values in this bit
string not meaningful.

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 whose
perceived severity cannot be determined are active
against this resource.

A value of 'unknown' means that this resource is
unable to report alarm state."

SYNTAX BITS

```

{
  unknown (0),
  underRepair (1),
  critical(2),
  major(3),
  minor(4),
  -- The following are not defined in X.733
  warning (5),
  indeterminate (6)
}

```

EntityStandbyStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" Represents the possible values of standby status.

A value of 'hotStandby' means the resource is not providing service, but it 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. A value of 'unknown' means that this resource is unable to report standby state."

SYNTAX INTEGER

```
{
  unknown (1),
  hotStandby (2),
  coldStandby (3),
  providingService (4)
}
```

END

5. Definitions

ENTITY-STATE-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, mib-2
FROM SNMPv2-SMI

DateAndTime

FROM SNMPv2-TC

MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP

FROM SNMPv2-CONF

entPhysicalIndex

FROM ENTITY-MIB

EntityAdminState, EntityOperState, EntityUsageState,

EntityAlarmStatus, EntityStandbyStatus

FROM ENTITY-STATE-TC-MIB;

entityStateMIB MODULE-IDENTITY

LAST-UPDATED "200511220000Z"

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@nortel.com

David T. Perkins
548 Qualbrook Ct
San Jose, CA 95110
USA
Phone: 408 394-8702
dperkins@snmpinfo.com

"

DESCRIPTION

"This MIB defines a state extension to the Entity MIB.

Copyright (C) The Internet Society 2005. This version
of this MIB module is part of RFC 4268; see the RFC
itself for full legal notices."

REVISION "200511220000Z"

DESCRIPTION

"Initial version, published as RFC 4268."

::= { mib-2 131 }

-- 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. Entries
appear in this table for values of
entPhysicalClass [RFC4133] that in this implementation
are able to report any of the state or status stored in
this table."

```

    "
 ::= { entStateObjects 1 }

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

EntStateEntry ::= SEQUENCE {
    entStateLastChanged  DateAndTime,
    entStateAdmin        EntityAdminState,
    entStateOper         EntityOperState,
    entStateUsage        EntityUsageState,
    entStateAlarm        EntityAlarmStatus,
    entStateStandby      EntityStandbyStatus
}

entStateLastChanged OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is the date and
         time when the value of any of entStateAdmin,
         entStateOper, entStateUsage, entStateAlarm,
         or entStateStandby changed for this entity.

         If there has been no change since
         the last re-initialization of the local system,
         this object contains the date and time of
         local system initialization.  If there has been
         no change since the entity was added to the
         local system, this object contains the date and
         time of the insertion."
    ::= { entStateEntry 1 }

entStateAdmin OBJECT-TYPE
    SYNTAX      EntityAdminState
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative state for this entity.

```

This object refers to an entities administrative permission to service both other entities within its containment hierarchy as well other users of its services defined by means outside the scope of this MIB.

Setting this object to 'notSupported' will result in an 'inconsistentValue' error. For entities that do not support administrative state, all set operations will result in an 'inconsistentValue' error.

Some physical entities exhibit only a subset of the remaining administrative state values. Some entities cannot be locked, and hence this object exhibits only the 'unlocked' state. Other entities cannot be shutdown gracefully, and hence this object does not exhibit the 'shuttingDown' state. A value of 'inconsistentValue' will be returned if attempts are made to set this object to values not supported by its administrative model."

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

entStateOper OBJECT-TYPE

SYNTAX EntityOperState

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The operational state for this entity.

Note that unlike the state model used within the Interfaces MIB [RFC2863], this object does not follow the administrative state. An administrative state of down does not predict an operational state of disabled.

A value of 'testing' means that entity currently being tested and cannot therefore report whether it is operational or not.

A value of 'disabled' means that an entity is totally inoperable and unable to provide service both to entities within its containment hierarchy, or to other receivers of its service as defined in ways outside the scope of this MIB.

A value of 'enabled' means that an entity is fully or partially operable and able to provide service both to

entities within its containment hierarchy, or to other receivers of its service as defined in ways outside the scope of this MIB.

Note that some implementations may not be able to accurately report entStateOper while the entStateAdmin object has a value other than 'unlocked'. In these cases, this object MUST have a value of 'unknown'."

::= { entStateEntry 3 }

entStateUsage OBJECT-TYPE

SYNTAX EntityUsageState

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The usage state for this entity.

This object refers to an entity's ability to service more physical entities in a containment hierarchy. A value of 'idle' means this entity is able to contain other entities but that no other entity is currently contained within this entity.

A value of 'active' means that at least one entity is contained within this entity, but that it could handle more. A value of 'busy' means that the entity is unable to handle any additional entities being contained in it.

Some entities will exhibit only a subset of the usage state values. Entities that are unable to ever service any entities within a containment hierarchy will always have a usage state of 'busy'. Some entities will only ever be able to support one entity within its containment hierarchy and will therefore only exhibit values of 'idle' and 'busy'."

::= { entStateEntry 4 }

entStateAlarm OBJECT-TYPE

SYNTAX EntityAlarmStatus

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The alarm status for this entity. It does not include the alarms raised on child components within its containment hierarchy.

A value of 'unknown' means that this entity is

unable to report alarm state. Note that this differs from 'indeterminate', which means that alarm state is supported and there are alarms against this entity, but the severity of some of the alarms is not known.

If no bits are set, then this entity supports reporting of alarms, but there are currently no active alarms against this entity."

```
::= { entStateEntry 5 }
```

```
entStateStandby OBJECT-TYPE
```

```
SYNTAX EntityStateStandbyStatus
```

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

```
STATUS current
```

```
DESCRIPTION
```

"The standby status for this entity.

Some entities will exhibit only a subset of the remaining standby state values. If this entity cannot operate in a standby role, the value of this object will always be 'providingService'."

```
::= { entStateEntry 6 }
```

```
-- Notifications
```

```
entStateNotifications OBJECT IDENTIFIER ::= { entityStateMIB 0 }
```

```
entStateOperEnabled NOTIFICATION-TYPE
```

```
OBJECTS { entStateAdmin,
           entStateAlarm
        }
```

```
STATUS current
```

```
DESCRIPTION
```

"An entStateOperEnabled notification signifies that the SNMP entity, acting in an agent role, has detected that the entStateOper object for one of its entities has transitioned into the 'enabled' state.

The entity this notification refers can be identified by extracting the entPhysicalIndex from one of the variable bindings. The entStateAdmin and entStateAlarm varbinds may be examined to find out additional information on the administrative state at the time of the operation state change as well as to find out whether there were any known alarms against the entity at that time that may explain why the physical entity has become operationally disabled."

```
::= { entStateNotifications 1 }
```

entStateOperDisabled NOTIFICATION-TYPE

OBJECTS { entStateAdmin,
 entStateAlarm }
STATUS current

DESCRIPTION

"An entStateOperDisabled notification signifies that the SNMP entity, acting in an agent role, has detected that the entStateOper object for one of its entities has transitioned into the 'disabled' state.

The entity this notification refers can be identified by extracting the entPhysicalIndex from one of the variable bindings. The entStateAdmin and entStateAlarm varbinds may be examined to find out additional information on the administrative state at the time of the operation state change as well as to find out whether there were any known alarms against the entity at that time that may affect the physical entity's ability to stay operationally enabled."

::= { entStateNotifications 2 }

-- Conformance and Compliance

entStateConformance OBJECT IDENTIFIER ::= { entityStateMIB 2 }

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
}

GROUP entStateNotificationsGroup

DESCRIPTION

"This group is optional."

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}

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

END
```

6. Security Considerations

The ENTITY-STATE-TC-MIB defined in section 4 does not define any management objects. Instead, it defines a set of textual conventions that may be used by other MIB modules to define management objects. Meaningful security considerations can only be written in the MIB modules that define management objects. The ENTITY-STATE-TC-MIB has therefore no impact on the security of the Internet.

The ENTITY-STATE-MIB defined in section 5 defines one management object -- entStateAdmin -- 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.

Note that setting the entStateAdmin to 'locked' or 'shuttingDown' can cause disruption of services ranging from those running on a port to those on an entire device, depending on the type of entity. Access to this object should be properly protected.

Access to the objects defined in this MIB allows one to figure out what the active and standby resources in a network are. This information can be used to optimize attacks on networks so even read-only access to this MIB should be properly protected.

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.

7. Acknowledgements

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

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., 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.

- [RFC4133] Bierman, A. and K. McCloghrie, "Entity MIB (Version 3)", RFC 4133, August 2005.

8.2. Informative References

- [RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2863, June 2000.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3877] Chisholm, S. and D. Romascanu, "Alarm Management Information Base (MIB)", RFC 3877, September 2004.
- [RFC4188] Norseth, K. and E. Bell, "Definitions of Managed Objects for Bridges", RFC 4188, September 2005.
- [X.731] ITU Recommendation X.731, "Information Technology - Open Systems Interconnection - System Management: State Management Function", 1992.

Authors' Addresses

Sharon Chisholm
Nortel Networks
PO Box 3511, Station C
Ottawa, Ontario, K1Y 4H7
Canada

Email: schishol@nortel.com

David T. Perkins
548 Qualbrook Ct
San Jose, CA 95110
USA

Phone: 408 394-8702
Email: dperkins@snmpinfo.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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

