Network Working Group
Request for Comments: 3877
Category: Standards Track

S. Chisholm
Nortel Networks
D. Romascanu
Avaya

September 2004

Alarm Management Information Base (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

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 management objects used for modelling and storing alarms.

Table of Contents

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

2. Introduction

In traditional SNMP management, problems are detected on an entity either through polling interesting MIB variables, waiting for the entity to send a Notification for a problem, or some combination of the two. This method is somewhat successful, but experience has shown some problems with this approach. Managers monitoring large numbers of entities cannot afford to be polling large numbers of objects on each device. Managers trying to ensure high reliability are unable to accurately determine whether any problems had occurred when they were not monitoring an entity. Finally, it can be time consuming for managers to try to understand the relationships between the various objects they poll, the Notifications they receive and the problems occurring on the entity. Even after detailed analysis they may still be left with an incomplete picture of what problems are occurring. But, it is important for an operator to be able to determine current problems on a system, so they can be fixed.

This memo describes a method of using alarm management in SNMP to address these problems. It also provides the necessary MIB objects to support this method.

Alarms and other terms related to alarm management are defined in the following sections.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

3. Alarm Management Framework

3.1. Terminology

Error
: A deviation of a system from normal operation.

Fault
: Lasting error or warning condition.

Event
: Something that happens which may be of interest. A fault, a change in status, crossing a threshold, or an external input to the system, for example.

Notification
: Unsolicited transmission of management information.

Alarm
: Persistent indication of a fault.

Alarm State
: A condition or stage in the existence of an alarm. As a minimum, alarms states are raise and clear. They could also include severity information such as defined by perceived severity in the International Telecommunications Union (ITU) model [M.3100] - cleared, indeterminate, critical, major, minor and warning.

Alarm Raise
: The initial detection of the fault indicated by an alarm or any number of alarm states later entered, except clear.

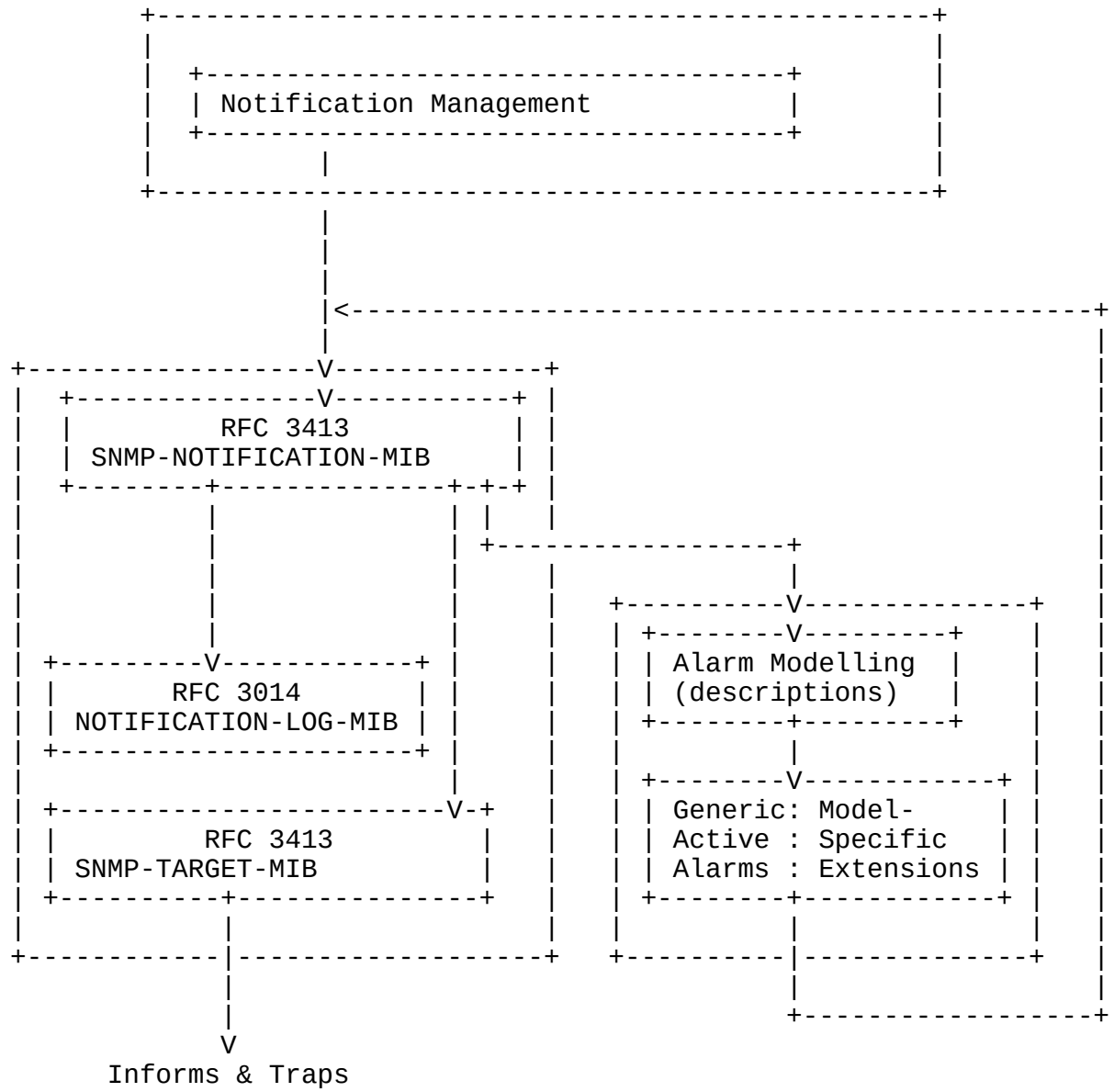
Alarm Clear
: The detection that the fault indicated by an alarm no longer exists.

Active Alarm
: An alarm which has an alarm state that has been raised, but not cleared.

Alarm Detection Point
: The entity that detected the alarm.

Perceived Severity
: The severity of the alarm as determined by the alarm detection point using the information it has available.

3.2. Alarm Management Architecture



3.3. Features of this Architecture

3.3.1. Modular Alarm Architecture

The subject of alarm management can potentially cover a large number of topics including real-time alarms, historical alarms, alarm correlation, and alarm suppression, to name a few. Within each of these topics, there are a number of established models that could be

supported. This memo focuses on a subset of this problem space, but describes a modular SNMP alarm management framework. Alarms SHOULD be modelled so Notifications are sent on alarm Clear.

The framework defines a generic Alarm MIB that can be supported on its own, or with additional alarm modelling information such as the provided ITU Alarm MIB. In addition, the active alarm tables could also be extended to support additional information about active alarm instances. This framework can also be expanded in the future to support such features as alarm correlation and alarm suppression. This modular architecture means that the cost of supporting alarm management features is proportional to the number of features an implementation supports.

3.3.2. Flexible Alarm Modelling

Alarm models document an understanding between a manager and an agent as to what problems will be reported on a system, how these problems will be reported, and what might possibly happen over the lifetime of this problem.

The alarm modelling method provided in this memo provides flexibility to support implementations with different modelling requirements. All alarms are modelled as a series of states that are related together using an alarm ID. Alarm states can be modelled using traditional Notifications, generic alarm Notifications, or without the use of Notifications.

Alarm states modelled using traditional Notifications would specify a Notification Object Identifier, and optionally an (offset, value) pair of one of the Notification varbinds to identify the state. This alarm state would be entered when the entity generated a Notification that matched this information and the alarm would be added to the active alarm table. This Notification would also get sent on the wire to any destinations, as indicated in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB [RFC3413].

Alarm states modelled using generic Notifications use the alarmActiveState or alarmClearState Notifications defined in this memo. These alarm states would be entered after being triggered by a stimulus outside the scope of this memo, the alarm would be added to the active alarm table and these generic Notifications would then be sent on the wire to any destinations, as indicated in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB [RFC3413].

Alarm states modelled without any Notifications would be triggered by some stimulus outside the scope of this memo, the alarm would be added to the active alarm table, but no Notifications would be sent to interested managers.

3.3.3. Problem Indication

The Alarm MIB provides a means to determine whether a given notification is of interest to managers for purposes of alarm management by permitting inspection of the alarm models. If no entries in the alarmModelTable could match a particular notification, then that notification is not relevant to the alarm models defined. In addition, information in the alarm model, such as the Notification ID and the description tell exactly what error or warning condition this alarm is indicating. If the ITU-ALARM-MIB is also supported, additional information is provided via the probable cause.

3.3.5. Identifying Resource under Alarm

An important goal of alarm management is to ensure that any detected problems get fixed, so it is necessary to know exactly where this problem is occurring. In addition, it is necessary to be able to tell when alarm instances are raised against the same component, as well as to be able to tell what instance of an alarm is cleared by an instance of an alarm clear.

The Alarm MIB provides a generic method for identifying the resource by extracting and building a resource ID from the Notification varbinds. It records the relevant information needed to locate the source of the alarm.

3.3.6. Means of obtaining ITU alarm information

Alarm Information, as defined in ITU alarm models [M.3100], is optionally available to implementations through the optional support of the ITU-ALARM-MIB.

3.3.7. Configuration of Alarm Models

An alarm model can be added and removed during runtime. It can be modified assuming it is not being referenced by any active alarm instance.

3.3.8. Active Alarm Management

A list of currently active alarms and supporting statistics on the SNMP entity can be obtained.

This allows the network management station to find out about any problems that may have occurred before it started managing a particular network element, or while it was out of contact with it.

3.3.9. Distributed Alarm Management

All aspects of the Alarm MIB can be supported both on the device experiencing the alarms and on any mid-level managers that might be monitoring such devices.

3.3.10. Historical Alarm Management

Some systems may have a requirement that information on alarms that are no longer active is available. This memo provides a clear table to support this requirement.

This can also be achieved through the support of the Notification Log MIB [RFC3014] to store alarm state transitions.

3.4. Security

Given the nature of VACM, security for alarms is awkward since access control for the objects in the underlying Notifications can be checked only where the Notification is created. Thus such checking is possible only for locally generated Notifications, and even then only when security credentials are available.

For the purpose of this discussion, "security credentials" means the input values for the abstract service interface function isAccessAllowed [RFC3411] and using those credentials means conceptually using that function to see that those credentials allow access to the MIB objects in question, operating as for a Notification Originator in [RFC3413].

The Alarm MIB has the notion of a named alarm list. By using alarm list names and view-based access control [RFC3415] a network administrator can provide different access for different users. When an application creates an alarm model (indexed in part by the alarm list name) the security credentials of the creator remain associated with that alarm model and constrain what information is allowed to be placed in the active alarm table, the active alarm variable table, the cleared alarm table, and the ITU alarm table.

When processing locally-generated Notifications, the managed system MUST use the security credentials associated with each alarm model respectively, and MUST apply the same access control rules as described for a Notification Originator in [RFC3413].

The managed system SHOULD NOT apply access control when processing remotely-generated Notifications using the alarm models. In those cases the security of the information in the alarm tables SHOULD be left to the normal, overall access control for those tables.

3.5. Relationship between Alarm and Notifications

It is important to understand the relationship between alarms and Notifications, as both are traditional fault management methods. This relationship is modelled using the alarmModelTable to define the alarmModelNotificationId for each alarm state.

Not all Notifications signal an alarm state transition. Some Notifications are simply informational in nature, such as those that indicate that a configuration operation has been performed on an entity. These sorts of Notifications would not be represented in the Alarm MIB.

The Alarm MIB allows the use of the Notification space as defined in [RFC2578] in order to identify the Notifications that are related with the specific alarm state transitions. However there is no assumption that the respective Notifications must be sent for all or any of the alarm state transitions. It is also possible to model alarms using no Notifications at all. This architecture allows for both the efficient exploitation of the body of defined Notification and for the use of non-Notification based systems.

3.6. Notification Varbind Storage and Reference

In SNMPv1 [RFC1157], the varbinds in the Trap-PDU sent over the wire map one to one into those varbinds listed in the SMI of the trap in the MIB in which it was defined [RFC1215]. In the case of linkDown trap, the first varbind can unambiguously be identified as ifIndex. With the introduction of the InformRequest-PDU and SNMPv2-Trap-PDU types, which send sysUptime and snmpTrapOID as the first two varbinds, while the SMI in the MIB where the Notification is defined only lists additional varbinds, the meaning of "first varbind" becomes less clear. In the case of the linkDown Notification, referring to the first varbind could potentially be interpreted as either the sysUptime or ifIndex.

The varbind storage approach taken in the Alarm MIB is that sysUptime and snmpTrapOID SHALL always be stored in the active alarm variable table as entry 1 and 2 respectively, regardless of whether the transport was the Trap-PDU, the InformRequest-PDU or the SNMPv2-Trap-PDU. If the incoming Notification is an SNMPv1 Trap-PDU then an appropriate value for sysUpTime.0 or snmpTrapOID.0 shall be determined by using the rules in section 3.1 of [RFC3584].

The varbind reference approach taken in the Alarm MIB is that, for variables such as the alarmModelVarbindIndex, the first two obligatory varbinds of the InformRequest-PDU and SNMPv2-Trap-PDU need to be considered so the index values of the Trap-PDU and the SMI need be adjusted by two. In the case of linkDown, the third varbind would always be ifIndex.

3.7. Relation to Notification Log MIB

The Alarm MIB is intended to complement the Notification Log MIB [RFC3014], but can be used independently. The alarmActiveTable is defined in manner similar to that of the nlmLogTable. This format allows for the storage of any Trap or Notification type that can be defined using the SMI, or can be carried by SNMP. Using the same format as the Notification Log MIB also simplifies operations for systems choosing to implement both MIBs.

The object alarmActiveLogPointer points, for each entry in the alarmActiveLogTable, to the log index in the Notification Log MIB, if used.

If the Notification Log MIB is supported, it can be monitored by a management system as a hedge against lost alarms. The Notification Log can also be used to support historical alarm management.

3.8. Relationship with the Event MIB

During the work and discussions in the Working Group, the issue of the relationship between the MIB modules and the Event MIB [RFC2981] was raised. There is no direct relation or dependency between the Alarm MIB and the Event MIB. Some common terms (like 'event') are being used in both MIB modules, and the user is directed to the sections that define terminology in the two documents for clarification.

4. Generic Alarm MIB

4.1. Overview

The ALARM-MIB consists of alarm models and lists of active and cleared alarms.

The alarmModelTable contains information that is applicable to all instances of an alarm. It can be populated at start-up with all alarms that could happen on a system or later configured by a management application. It contains all the alarms for a given system. If a Notification is not represented in the alarmModelTable, it is not an alarm state transition. The alarmModelTable provides a

means of defining the raise/clear and other state transition relationships between alarm states. The alarmModelIndex acts as a unique identifier for an alarm. An alarm model consists of definitions of the possible states an alarm can assume as well as the Object Identifier (OID) of the Notification associated with this alarm state. The object alarmModelState defines the states of an alarm.

The alarmActiveTable contains a list of alarms that are currently occurring on a system. It is intended that this table be queried upon device discovery and rediscovery to determine which alarms are currently active on the device.

The alarmActiveVariableTable contains the Notification variable bindings associated with the alarms in the alarmActiveTable.

The alarmActiveStatsTable contains current and total raised alarm counts as well as the time of the last alarm raise and alarm clears per named alarm list.

The alarmClearTable contains recently cleared alarms. It contains up to alarmClearMaximum cleared alarms.

The MIB also defines generic alarm Notifications that can be used when there is not an existing applicable Notification to signal the alarm state transition - alarmActiveState and alarmClearState.

4.1.1. Extensibility

The relationship between the Alarm MIB and the other alarm model MIB modules is expressed by the following: The alarmModelTable has a corresponding table in the specific MIB. For each row in the specific MIB alarm model table there is one row in the alarmModelTable. The alarmActiveTable has a corresponding table in the specific MIBs. For each row in the specific MIB active alarm table, there is one row in the alarmActiveTable. The alarmModelSpecificPointer object in the alarmModelTable points to the specific model entry in an extended alarm model table corresponding to this particular alarm. The alarmActiveSpecificPointer object in the alarmActiveTable points to the specific active alarm entry in an extended active alarm table corresponding to this particular alarm instance.

Additional extensions can be defined by defining an AUGMENTATION of either the Alarm or ITU Alarm tables. As the alarm model table only provides a mechanism to point at one specific alarm model, additional specific models SHOULD define another mechanism to map from the generic alarm model to the additional model.

4.1.2. Problem Indication

The problem that each alarm indicates is identified through the Object Identifier of the NotificationId of the state transition, and, optionally, the ITU parameters. alarmModelDescription provides a description of the alarm state suitable for displaying to an operator.

4.1.3. Alarm State Transition Notification

The SNMP-TARGET-MIB [RFC3413] provides the ability to specify which managers, if any, receive Notifications of problems. Solutions can therefore use the features of this MIB to change the Notification behaviour of their implementations. Specifying target hosts in this MIB along with specifying notifications in the alarmModelNotificationId would allow Notifications to be logged and sent out to management stations in an architecture as described in section 3.2. Specifying no target hosts in this MIB along with specifying notifications in the alarmModelNotificationId would allow Notifications to be logged but not sent out to management stations in an architecture as described in section 3.2. Regardless of what is defined in the SNMP-TARGET-MIB, specifying { 0 0 } in the alarmModelNotificationId would result in no notifications being logged or sent to management stations as a consequence of this particular alarm state transition.

Alarms are modelled by defining all possible states in the alarmModelTable, as well as defining alarmModelNotificationId, alarmModelVarbindIndex, and alarmModelVarbindValue for each of the possible alarm states. Optionally, ituAlarmPerceivedSeverity models the states in terms of ITU perceived severity.

4.1.4. Active Alarm Resource Identifier

Resources under alarm can be identified using the alarmActiveResourceId. This OBJECT IDENTIFIER points to an appropriate object to identify the given resource, depending on the type of the resource.

The consumer of the alarmActiveResourceId does not necessarily need to know the type of the resource in the resource ID, but if they want to know this, examining the content of the resource ID can derive it - 1.3.6.1.2.1.2.2.1.1.something is an interface, for example. It is therefore good practice to use resource IDs that can be consistently used across technologies, such as ifIndex, entPhysicalIndex or sysApplRunIndex, to minimize the number of resource prefixes a manager interested in a resource type needs to learn.

Resource ID can be calculated using the alarmModelResourcePrefix, alarmModelVarbindSubtree and the Notification varbinds. This allows for both the managed element to be able to compute and populate the alarmActiveResourceId object and for the manager to be able to determine when two separate alarm instances are referring to the same resource.

If alarmModelResourcePrefix has a value of 0.0, then alarmActiveResourceId is simply the variable identifier of the first Notification varbind that matches the prefix defined in alarmModelVarbindSubtree. Otherwise, alarmActiveResourceId is calculated by appending the instance information from the first Notification varbind that matches alarmModelVarbindSubtree to the prefix defined in alarmModelResourcePrefix. The instance information is the portion of the variable identifier following the part that matched alarmModelVarbindSubtree. If no match is found, then alarmActiveResourceId is simply the value of alarmModelResourcePrefix.

In addition to this, the variable bindings from the Notifications that signal the alarm state transitions are stored in the active alarm variable table. This allows for implementations familiar with the particular Notifications to implement other forms of resource identification.

For Example:

A) Consider an alarm modelled using the authenticationFailure [RFC3418] Notification.

```
authenticationFailure NOTIFICATION-TYPE
 STATUS  current
 DESCRIPTION
      "An authenticationFailure trap signifies that the SNMPv2
      entity, acting in an agent role, has received a protocol
      message that is not properly authenticated.  While all
      implementations of the SNMPv2 must be capable of generating
      this trap, the snmpEnableAuthenTraps object indicates
      whether this trap will be generated."
 ::= { snmpTraps 5 }
```

To set the resource ID to be usmStats, 1.3.6.1.6.3.15.1.1, configure as follows:

```
alarmModelVarbindSubtree = 0.0
alarmModelResourcePrefix = usmStats (1.3.6.1.6.3.15.1.1)
```

B) Consider an alarm modelled using linkDown [RFC2863]

```
  linkDown NOTIFICATION-TYPE
          OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
          STATUS  current
          DESCRIPTION
              ""
      ::= { snmpTraps 3 }
```

To set the resource Id to be the ifIndex, configure as follows:

```
      alarmModelVarbindSubtree = ifIndex (1.3.6.1.2.1.2.2.1.1)
      alarmModelResourcePrefix = 0.0
```

Alternatively, since ifIndex is the first varbind, the following would also work, but might be less meaningful to a human reader of the MIB table:

```
      alarmModelVarbindSubtree = 0.0
      alarmModelResourcePrefix = 0.0
```

C) Consider an alarm modelled using the bgpBackwardTransition [RFC1657] Notification.

```
  bgpBackwardTransition NOTIFICATION-TYPE
          OBJECTS { bgpPeerLastError,
                    bgpPeerState      }
          STATUS  current
          DESCRIPTION
                "The BGPBackwardTransition Event is generated
                when the BGP FSM moves from a higher numbered
                state to a lower numbered state."
          ::= { bgpTraps 2 }
```

To set the resource Id to be the bgpPeerRemoteAddr, the index to the bgpTable, where bgpPeerState resides, configure as follows:

```
      alarmModelVarbindSubtree = bgpPeerState
                                          (1.3.6.1.2.1.15.3.1.2)
      alarmModelResourcePrefix = bgpPeerRemoteAddr
        (1.3.6.1.2.1.15.3.1.7)
```

4.1.5. Configurable Alarm Models

The alarm model table SHOULD be initially populated by the system. The objects in alarmModelTable and ituAlarmTable have a MAX-ACCESS of read-create, which allows managers to modify the alarm models to suit their requirements.

4.1.6. Active Alarm Management

Lists of alarms currently active on an SNMP entity are stored in the alarmActiveTable and, optionally, a model specific alarmTable, e.g., the ituAlarmActiveTable.

4.1.7. Distributed Alarm Management

Distributed alarm management can be achieved by support of the Alarm MIB on both the alarm detection point and on the mid-level manager. This is facilitated by the ability to be able to store different named alarm lists. A mid-level manager could create an alarmListName for each of the devices it manages and therefore store separate lists for each device. In addition, the context and IP addresses of the alarm detection point are stored in the alarmActiveTable.

4.2. Definitions

```
ALARM-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   Integer32, Unsigned32, Gauge32,
   TimeTicks, Counter32, Counter64,
   IpAddress, Opaque, mib-2,
   zeroDotZero
       FROM SNMPv2-SMI                 -- [RFC2578]
   DateAndTime,
   RowStatus, RowPointer,
   TEXTUAL-CONVENTION
       FROM SNMPv2-TC                  -- [RFC2579]
   SnmpAdminString
       FROM SNMP-FRAMEWORK-MIB         -- [RFC3411]
   InetAddressType, InetAddress
       FROM INET-ADDRESS-MIB           -- [RFC3291]
   MODULE-COMPLIANCE, OBJECT-GROUP,
   NOTIFICATION-GROUP
       FROM SNMPv2-CONF                -- [RFC2580]
   ZeroBasedCounter32
       FROM RMON2-MIB;                 -- [RFC2021]

  alarmMIB MODULE-IDENTITY
      LAST-UPDATED "200409090000Z"  -- September 09, 2004
      ORGANIZATION "IETF Distributed Management Working Group"
      CONTACT-INFO
           "WG EMail: disman@ietf.org
           Subscribe: disman-request@ietf.org
           http://www.ietf.org/html.charters/disman-charter.html
```

```
            Chair:     Randy Presuhn
                       randy_presuhn@mindspring.com

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

                       Dan Romascanu
                       Avaya
                       Atidim Technology Park, Bldg. #3
                       Tel Aviv, 61131
                       Israel
                       Tel: +972-3-645-8414
                       Email: dromasca@avaya.com"
       DESCRIPTION
            "The MIB module describes a generic solution
            to model alarms and to store the current list
            of active alarms.

            Copyright (C) The Internet Society (2004).  The
            initial version of this MIB module was published
            in RFC 3877.  For full legal notices see the RFC
            itself.  Supplementary information may be available on:
            http://www.ietf.org/copyrights/ianamib.html"
       REVISION    "200409090000Z"  -- September 09, 2004
       DESCRIPTION
           "Initial version, published as RFC 3877."
       ::= { mib-2 118 }

alarmObjects OBJECT IDENTIFIER ::= { alarmMIB 1 }

alarmNotifications OBJECT IDENTIFIER ::= { alarmMIB 0 }

alarmModel OBJECT IDENTIFIER ::= { alarmObjects 1 }

alarmActive  OBJECT IDENTIFIER ::= { alarmObjects 2 }

alarmClear OBJECT IDENTIFIER ::= { alarmObjects 3 }

-- Textual Conventions

 -- ResourceId is intended to be a general textual convention
 -- that can be used outside of the set of MIBs related to
 -- Alarm Management.
```

```
ResourceId ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "A unique identifier for this resource.

            The type of the resource can be determined by looking
            at the OID that describes the resource.

            Resources must be identified in a consistent manner.
            For example, if this resource is an interface, this
            object MUST point to an ifIndex and if this resource
            is a physical entity [RFC2737], then this MUST point
            to an entPhysicalDescr, given that entPhysicalIndex
            is not accessible.  In general, the value is the
            name of the instance of the first accessible columnar
            object in the conceptual row of a table that is
            meaningful for this resource type, which SHOULD
            be defined in an IETF standard MIB."
    SYNTAX         OBJECT IDENTIFIER

 -- LocalSnmpEngineOrZeroLenStr is intended to be a general
 -- textual convention that can be used outside of the set of
 -- MIBs related to Alarm Management.

  LocalSnmpEngineOrZeroLenStr ::= TEXTUAL-CONVENTION
      STATUS current
      DESCRIPTION
          "An SNMP Engine ID or a zero-length string.  The
           instantiation of this textual convention will provide
           guidance on when this will be an SNMP Engine ID and
           when it will be a zero lengths string"
      SYNTAX         OCTET STRING (SIZE(0 | 5..32))

-- Alarm Model

alarmModelLastChanged  OBJECT-TYPE
      SYNTAX      TimeTicks
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The value of sysUpTime at the time of the last
         creation, deletion or modification of an entry in
         the alarmModelTable.

         If the number and content of entries has been unchanged
         since the last re-initialization of the local network
         management subsystem, then the value of this object
         MUST be zero."
```

```
      ::= { alarmModel 1 }

alarmModelTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF AlarmModelEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "A table of information about possible alarms on the system,
        and how they have been modelled."
   ::= { alarmModel 2 }

alarmModelEntry OBJECT-TYPE
   SYNTAX      AlarmModelEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "Entries appear in this table for each possible alarm state.
       This table MUST be persistent across system reboots."
   INDEX       { alarmListName, alarmModelIndex, alarmModelState }
   ::= { alarmModelTable 1 }

AlarmModelEntry ::= SEQUENCE {
   alarmModelIndex                 Unsigned32,
   alarmModelState                 Unsigned32,
   alarmModelNotificationId        OBJECT IDENTIFIER,
   alarmModelVarbindIndex          Unsigned32,
   alarmModelVarbindValue          Integer32,
   alarmModelDescription           SnmpAdminString,
   alarmModelSpecificPointer       RowPointer,
   alarmModelVarbindSubtree        OBJECT IDENTIFIER,
   alarmModelResourcePrefix        OBJECT IDENTIFIER,
   alarmModelRowStatus             RowStatus
   }

alarmModelIndex OBJECT-TYPE
   SYNTAX     Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An integer that acts as an alarm Id
       to uniquely identify each alarm
       within the named alarm list. "
   ::= { alarmModelEntry 1 }

alarmModelState OBJECT-TYPE
   SYNTAX  Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS       current
```

```
   DESCRIPTION
        "A value of 1 MUST indicate a clear alarm state.
        The value of this object MUST be less than the
        alarmModelState of more severe alarm states for
        this alarm.  The value of this object MUST be more
        than the alarmModelState of less severe alarm states
        for this alarm."
    ::= { alarmModelEntry 2 }

alarmModelNotificationId OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-create
   STATUS      current
   DESCRIPTION
       "The NOTIFICATION-TYPE object identifier of this alarm
       state transition.  If there is no notification associated
       with this alarm state, the value of this object MUST be
       '0.0'"
   DEFVAL { zeroDotZero }
   ::= { alarmModelEntry 3 }

alarmModelVarbindIndex  OBJECT-TYPE
   SYNTAX  Unsigned32
   MAX-ACCESS   read-create
   STATUS       current
   DESCRIPTION
     "The index into the varbind listing of the notification
     indicated by alarmModelNotificationId which helps
     signal that the given alarm has changed state.
     If there is no applicable varbind, the value of this
     object MUST be zero.

     Note that the value of alarmModelVarbindIndex acknowledges
     the existence of the first two obligatory varbinds in
     the InformRequest-PDU and SNMPv2-Trap-PDU (sysUpTime.0
     and snmpTrapOID.0).  That is, a value of 2 refers to
     the snmpTrapOID.0.

     If the incoming notification is instead an SNMPv1 Trap-PDU,
     then an appropriate value for sysUpTime.0 or snmpTrapOID.0
     shall be determined by using the rules in section 3.1 of
     [RFC3584]"
     DEFVAL { 0 }
    ::= { alarmModelEntry 4 }

alarmModelVarbindValue OBJECT-TYPE
   SYNTAX  Integer32
   MAX-ACCESS   read-create
```

```
   STATUS       current
   DESCRIPTION
     "The value that the varbind indicated by
     alarmModelVarbindIndex takes to indicate
     that the alarm has entered this state.

     If alarmModelVarbindIndex has a value of 0, so
     MUST alarmModelVarbindValue.
     "
     DEFVAL { 0 }
    ::= { alarmModelEntry 5 }

alarmModelDescription OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
      "A brief description of this alarm and state suitable
      to display to operators."
   DEFVAL { "" }
   ::= { alarmModelEntry 6 }

alarmModelSpecificPointer OBJECT-TYPE
   SYNTAX     RowPointer
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
     "If no additional, model-specific Alarm MIB is supported by
      the system the value of this object is '0.0'and attempts
      to set it to any other value MUST be rejected appropriately.

      When a model-specific Alarm MIB is supported, this object
      MUST refer to the first accessible object in a corresponding
      row of the model definition in one of these model-specific
      MIB and attempts to set this object to { 0 0 } or any other
      value MUST be rejected appropriately."
   DEFVAL { zeroDotZero }
   ::= { alarmModelEntry 7 }

  alarmModelVarbindSubtree  OBJECT-TYPE
     SYNTAX  OBJECT IDENTIFIER
     MAX-ACCESS   read-create
     STATUS       current
     DESCRIPTION
       "The name portion of each VarBind in the notification,
        in order, is compared to the value of this object.
        If the name is equal to or a subtree of the value
        of this object, for purposes of computing the value
```

```
        of AlarmActiveResourceID the 'prefix' will be the
        matching portion, and the 'indexes' will be any
        remainder.  The examination of varbinds ends with
        the first match.  If the value of this object is 0.0,
        then the first varbind, or in the case of v2, the
        first varbind after the timestamp and the trap
        OID, will always be matched.
       "
      DEFVAL { zeroDotZero }
     ::= { alarmModelEntry 8 }

  alarmModelResourcePrefix  OBJECT-TYPE
     SYNTAX  OBJECT IDENTIFIER
     MAX-ACCESS   read-create
     STATUS       current
     DESCRIPTION
       "The value of AlarmActiveResourceId is computed
        by appending any indexes extracted in accordance
        with the description of alarmModelVarbindSubtree
        onto the value of this object.  If this object's
        value is 0.0, then the 'prefix' extracted is used
        instead.
       "
     DEFVAL { zeroDotZero }
     ::= { alarmModelEntry 9 }

alarmModelRowStatus OBJECT-TYPE
   SYNTAX     RowStatus
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
    "Control for creating and deleting entries.  Entries may be
    modified while active.  Alarms whose alarmModelRowStatus is
    not active will not appear in either the alarmActiveTable
    or the alarmClearTable.  Setting this object to notInService
    cannot be used as an alarm suppression mechanism.  Entries
    that are notInService will disappear as described in RFC2579.

    This row can not be modified while it is being
    referenced by a value of alarmActiveModelPointer.  In these
    cases, an error of 'inconsistentValue' will be returned to
    the manager.

    This entry may be deleted while it is being
    referenced by a value of alarmActiveModelPointer.  This results
    in the deletion of this entry and entries in the active alarms
    referencing this entry via an alarmActiveModelPointer.
```

```
    As all read-create objects in this table have a DEFVAL clause,
    there is no requirement that any object be explicitly set
    before this row can become active.  Note that a row consisting
    only of default values is not very meaningful."
   ::= { alarmModelEntry 10 }

-- Active Alarm Table --

alarmActiveLastChanged  OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The value of sysUpTime at the time of the last
       creation or deletion of an entry in the alarmActiveTable.
       If the number of entries has been unchanged since the
       last re-initialization of the local network management
       subsystem, then this object contains a zero value."
   ::= { alarmActive 1 }

 alarmActiveOverflow  OBJECT-TYPE
     SYNTAX      Counter32
     UNITS       "active alarms"
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
        "The number of active alarms that have not been put into
         the alarmActiveTable since system restart as a result
         of extreme resource constraints."
     ::= { alarmActive 5 }

alarmActiveTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF AlarmActiveEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "A table of Active Alarms entries."
   ::= { alarmActive 2 }

alarmActiveEntry OBJECT-TYPE
   SYNTAX      AlarmActiveEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "Entries appear in this table when alarms are raised.  They
        are removed when the alarm is cleared.

        If under extreme resource constraint the system is unable to
```

```
        add any more entries into this table, then the
        alarmActiveOverflow statistic will be increased by one."
   INDEX       { alarmListName, alarmActiveDateAndTime,
                 alarmActiveIndex }
   ::= { alarmActiveTable 1 }

AlarmActiveEntry ::= SEQUENCE {
   alarmListName                    SnmpAdminString,
   alarmActiveDateAndTime           DateAndTime,
   alarmActiveIndex                 Unsigned32,
   alarmActiveEngineID              LocalSnmpEngineOrZeroLenStr,
   alarmActiveEngineAddressType     InetAddressType,
   alarmActiveEngineAddress         InetAddress,
   alarmActiveContextName           SnmpAdminString,
   alarmActiveVariables             Unsigned32,
   alarmActiveNotificationID        OBJECT IDENTIFIER,
   alarmActiveResourceId            ResourceId,
   alarmActiveDescription           SnmpAdminString,
   alarmActiveLogPointer            RowPointer,
   alarmActiveModelPointer          RowPointer,
   alarmActiveSpecificPointer       RowPointer }

alarmListName OBJECT-TYPE
   SYNTAX     SnmpAdminString (SIZE(0..32))
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
    "The name of the list of alarms.  This SHOULD be the same as
    nlmLogName if the Notification Log MIB [RFC3014] is supported.
    This SHOULD be the same as, or contain as a prefix, the
    applicable snmpNotifyFilterProfileName if the
    SNMP-NOTIFICATION-MIB DEFINITIONS [RFC3413] is supported.

    An implementation may allow multiple named alarm lists, up to
    some implementation-specific limit (which may be none).  A
    zero-length list name is reserved for creation and deletion
    by the managed system, and MUST be used as the default log
    name by systems that do not support named alarm lists."
   ::= { alarmActiveEntry 1 }

alarmActiveDateAndTime OBJECT-TYPE
   SYNTAX      DateAndTime
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "The local date and time when the error occurred.

       This object facilitates retrieving all instances of
```

```
        alarms that have been raised or have changed state
        since a given point in time.

        Implementations MUST include the offset from UTC,
        if available.  Implementation in environments in which
        the UTC offset is not available is NOT RECOMMENDED."
    ::= { alarmActiveEntry 2 }

alarmActiveIndex OBJECT-TYPE
    SYNTAX     Unsigned32 (1..4294967295)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "A strictly monotonically increasing integer which
        acts as the index of entries within the named alarm
        list.  It wraps back to 1 after it reaches its
        maximum value."
    ::= { alarmActiveEntry 3 }

alarmActiveEngineID OBJECT-TYPE
    SYNTAX      LocalSnmpEngineOrZeroLenStr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The identification of the SNMP engine at which the alarm
         originated.  If the alarm is from an SNMPv1 system this
         object is a zero length string."
    ::= { alarmActiveEntry 4 }

alarmActiveEngineAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
     "This object indicates what type of address is stored in
     the alarmActiveEngineAddress object - IPv4, IPv6, DNS, etc."
    ::= { alarmActiveEntry 5 }

alarmActiveEngineAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
     "The address of the SNMP engine on which the alarm is
     occurring.

     This object MUST always be instantiated, even if the list
     can contain alarms from only one engine."
```

```
   ::= { alarmActiveEntry 6 }

alarmActiveContextName OBJECT-TYPE
   SYNTAX      SnmpAdminString (SIZE(0..32))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The name of the SNMP MIB context from which the alarm came.
        For SNMPv1 alarms this is the community string from the Trap.
        Note that care MUST be taken when selecting community
        strings to ensure that these can be represented as a
        well-formed SnmpAdminString.  Community or Context names
        that are not well-formed SnmpAdminStrings will be mapped
        to zero length strings.

        If the alarm's source SNMP engine is known not to support
        multiple contexts, this object is a zero length string."
   ::= { alarmActiveEntry 7 }

alarmActiveVariables OBJECT-TYPE
   SYNTAX      Unsigned32
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The number of variables in alarmActiveVariableTable for this
       alarm."
   ::= { alarmActiveEntry 8 }

alarmActiveNotificationID OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The NOTIFICATION-TYPE object identifier of the alarm
       state transition that is occurring."
   ::= { alarmActiveEntry 9 }

alarmActiveResourceId    OBJECT-TYPE
   SYNTAX      ResourceId
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "This object identifies the resource under alarm.

      If there is no corresponding resource, then
      the value of this object MUST be 0.0."
   ::= { alarmActiveEntry 10 }
```

```
alarmActiveDescription    OBJECT-TYPE
   SYNTAX      SnmpAdminString
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "This object provides a textual description of the
      active alarm.  This text is generated dynamically by the
      notification generator to provide useful information
      to the human operator.  This information SHOULD
      provide information allowing the operator to locate
      the resource for which this alarm is being generated.
      This information is not intended for consumption by
      automated tools."
   ::= { alarmActiveEntry 11 }

alarmActiveLogPointer OBJECT-TYPE
   SYNTAX     RowPointer
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "A pointer to the corresponding row in a
       notification logging MIB where the state change
       notification for this active alarm is logged.
       If no log entry applies to this active alarm,
       then this object MUST have the value of 0.0"
   ::= { alarmActiveEntry 12 }

alarmActiveModelPointer OBJECT-TYPE
   SYNTAX     RowPointer
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "A pointer to the corresponding row in the
       alarmModelTable for this active alarm.  This
       points not only to the alarm model being
       instantiated, but also to the specific alarm
       state that is active."
   ::= { alarmActiveEntry 13 }

alarmActiveSpecificPointer OBJECT-TYPE
   SYNTAX     RowPointer
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
     "If no additional, model-specific, Alarm MIB is supported by
     the system this object is '0.0'.  When a model-specific Alarm
     MIB is supported, this object is the instance pointer to the
     specific model-specific active alarm list."
```

```
   ::= { alarmActiveEntry 14 }

-- Active Alarm Variable Table --

alarmActiveVariableTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF AlarmActiveVariableEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "A table of variables to go with active alarm entries."
   ::= { alarmActive 3 }

alarmActiveVariableEntry OBJECT-TYPE
   SYNTAX      AlarmActiveVariableEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
       "Entries appear in this table when there are variables in
       the varbind list of a corresponding alarm in
       alarmActiveTable.

       Entries appear in this table as though
       the trap/notification had been transported using a
       SNMPv2-Trap-PDU, as defined in [RFC3416] - i.e., the
       alarmActiveVariableIndex 1 will always be sysUpTime
       and alarmActiveVariableIndex 2 will always be
       snmpTrapOID.

       If the incoming notification is instead an SNMPv1 Trap-PDU and
       the value of alarmModelVarbindIndex is 1 or 2, an appropriate
       value for sysUpTime.0 or snmpTrapOID.0 shall be determined
       by using the rules in section 3.1 of [RFC3584]."
   INDEX   {  alarmListName, alarmActiveIndex,
              alarmActiveVariableIndex }
   ::= { alarmActiveVariableTable 1 }

AlarmActiveVariableEntry ::= SEQUENCE {
   alarmActiveVariableIndex                Unsigned32,
   alarmActiveVariableID                   OBJECT IDENTIFIER,
   alarmActiveVariableValueType            INTEGER,
   alarmActiveVariableCounter32Val         Counter32,
   alarmActiveVariableUnsigned32Val        Unsigned32,
   alarmActiveVariableTimeTicksVal         TimeTicks,
   alarmActiveVariableInteger32Val         Integer32,
   alarmActiveVariableOctetStringVal       OCTET STRING,
   alarmActiveVariableIpAddressVal         IpAddress,
   alarmActiveVariableOidVal               OBJECT IDENTIFIER,
   alarmActiveVariableCounter64Val         Counter64,
```

```
   alarmActiveVariableOpaqueVal             Opaque }

alarmActiveVariableIndex OBJECT-TYPE
   SYNTAX     Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "A strictly monotonically increasing integer, starting at
       1 for a given alarmActiveIndex, for indexing variables
       within the active alarm variable list. "
   ::= { alarmActiveVariableEntry 1 }

alarmActiveVariableID OBJECT-TYPE
   SYNTAX     OBJECT IDENTIFIER
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The alarm variable's object identifier."
   ::= { alarmActiveVariableEntry 2 }

alarmActiveVariableValueType OBJECT-TYPE
   SYNTAX      INTEGER {
         counter32(1),
         unsigned32(2),
         timeTicks(3),
         integer32(4),
         ipAddress(5),
         octetString(6),
         objectId(7),
         counter64(8),
         opaque(9)
         }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The type of the value.  One and only one of the value
       objects that follow is used for a given row in this table,
       based on this type."
   ::= { alarmActiveVariableEntry 3 }

alarmActiveVariableCounter32Val OBJECT-TYPE
   SYNTAX      Counter32
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'counter32'."
   ::= { alarmActiveVariableEntry 4 }
```

```
alarmActiveVariableUnsigned32Val OBJECT-TYPE
   SYNTAX      Unsigned32
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'unsigned32'."
   ::= { alarmActiveVariableEntry 5 }

alarmActiveVariableTimeTicksVal OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'timeTicks'."
   ::= { alarmActiveVariableEntry 6 }

alarmActiveVariableInteger32Val OBJECT-TYPE
   SYNTAX      Integer32
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'integer32'."
   ::= { alarmActiveVariableEntry 7 }

alarmActiveVariableOctetStringVal OBJECT-TYPE
   SYNTAX      OCTET STRING (SIZE(0..65535))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'octetString'."
   ::= { alarmActiveVariableEntry 8 }

alarmActiveVariableIpAddressVal OBJECT-TYPE
   SYNTAX      IpAddress
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'ipAddress'."
   ::= { alarmActiveVariableEntry 9 }

alarmActiveVariableOidVal OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'objectId'."
   ::= { alarmActiveVariableEntry 10 }
```

```
alarmActiveVariableCounter64Val OBJECT-TYPE
   SYNTAX      Counter64
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'counter64'."
   ::= { alarmActiveVariableEntry 11 }

alarmActiveVariableOpaqueVal OBJECT-TYPE
   SYNTAX      Opaque (SIZE(0..65535))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The value when alarmActiveVariableType is 'opaque'.

       Note that although RFC2578 [RFC2578] forbids the use
       of Opaque in 'standard' MIB modules, this particular
       usage is driven by the need to be able to accurately
       represent any well-formed notification, and justified
       by the need for backward compatibility."
   ::= { alarmActiveVariableEntry 12 }

-- Statistics --

alarmActiveStatsTable  OBJECT-TYPE
      SYNTAX  SEQUENCE OF AlarmActiveStatsEntry
      MAX-ACCESS  not-accessible
      STATUS  current
      DESCRIPTION
         "This table represents the alarm statistics
         information."
  ::= { alarmActive 4 }

alarmActiveStatsEntry OBJECT-TYPE
      SYNTAX  AlarmActiveStatsEntry
      MAX-ACCESS  not-accessible
      STATUS  current
      DESCRIPTION
         "Statistics on the current active alarms."
      INDEX   { alarmListName }

  ::= {  alarmActiveStatsTable 1 }

AlarmActiveStatsEntry ::=
      SEQUENCE {
           alarmActiveStatsActiveCurrent  Gauge32,
           alarmActiveStatsActives        ZeroBasedCounter32,
           alarmActiveStatsLastRaise      TimeTicks,
```

```
           alarmActiveStatsLastClear      TimeTicks
                }

alarmActiveStatsActiveCurrent OBJECT-TYPE
      SYNTAX Gauge32
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The total number of currently active alarms on the system."
       ::= { alarmActiveStatsEntry 1 }

alarmActiveStatsActives OBJECT-TYPE
      SYNTAX ZeroBasedCounter32
      MAX-ACCESS read-only
      STATUS  current
      DESCRIPTION
         "The total number of active alarms since system restarted."
       ::= { alarmActiveStatsEntry 2 }

alarmActiveStatsLastRaise  OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The value of sysUpTime at the time of the last
       alarm raise for this alarm list.
       If no alarm raises have occurred since the
       last re-initialization of the local network management
       subsystem, then this object contains a zero value."
 ::= { alarmActiveStatsEntry 3 }

alarmActiveStatsLastClear  OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "The value of sysUpTime at the time of the last
       alarm clear for this alarm list.
       If no alarm clears have occurred since the
       last re-initialization of the local network management
       subsystem, then this object contains a zero value."
 ::= { alarmActiveStatsEntry 4 }

-- Alarm Clear

alarmClearMaximum OBJECT-TYPE
 SYNTAX Unsigned32
 MAX-ACCESS read-write
```

```
 STATUS current
 DESCRIPTION
   "This object specifies the maximum number of cleared
   alarms to store in the alarmClearTable.  When this
   number is reached, the cleared alarms with the
   earliest clear time will be removed from the table."
 ::= { alarmClear 1 }

alarmClearTable  OBJECT-TYPE
      SYNTAX  SEQUENCE OF AlarmClearEntry
      MAX-ACCESS  not-accessible
      STATUS  current
      DESCRIPTION
         "This table contains information on
         cleared alarms."
  ::= { alarmClear 2 }

alarmClearEntry OBJECT-TYPE
      SYNTAX  AlarmClearEntry
      MAX-ACCESS  not-accessible
      STATUS  current
      DESCRIPTION
         "Information on a cleared alarm."
      INDEX   { alarmListName, alarmClearDateAndTime,
alarmClearIndex }

  ::= {  alarmClearTable 1 }

AlarmClearEntry ::=
      SEQUENCE {
   alarmClearIndex                Unsigned32,
   alarmClearDateAndTime          DateAndTime,
   alarmClearEngineID             LocalSnmpEngineOrZeroLenStr,
   alarmClearEngineAddressType    InetAddressType,
   alarmClearEngineAddress        InetAddress,
   alarmClearContextName          SnmpAdminString,
   alarmClearNotificationID       OBJECT IDENTIFIER,
   alarmClearResourceId           ResourceId,
   alarmClearLogIndex             Unsigned32,
   alarmClearModelPointer         RowPointer
   }

alarmClearIndex OBJECT-TYPE
   SYNTAX     Unsigned32 (1..4294967295)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An integer which acts as the index of entries within
```

```
        the named alarm list.  It wraps back to 1 after it
        reaches its maximum value.

        This object has the same value as the alarmActiveIndex that
        this alarm instance had when it was active."
    ::= { alarmClearEntry 1 }

alarmClearDateAndTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The local date and time when the alarm cleared.

        This object facilitates retrieving all instances of
        alarms that have been cleared since a given point in time.

        Implementations MUST include the offset from UTC,
        if available.  Implementation in environments in which
        the UTC offset is not available is NOT RECOMMENDED."
    ::= { alarmClearEntry 2 }

alarmClearEngineID OBJECT-TYPE
    SYNTAX      LocalSnmpEngineOrZeroLenStr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The identification of the SNMP engine at which the alarm
         originated.  If the alarm is from an SNMPv1 system this
         object is a zero length string."
    ::= { alarmClearEntry 3 }

alarmClearEngineAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
     "This object indicates what type of address is stored in
     the alarmActiveEngineAddress object - IPv4, IPv6, DNS, etc."
    ::= { alarmClearEntry 4 }

alarmClearEngineAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
     "The Address of the SNMP engine on which the alarm was
     occurring.  This is used to identify the source of an SNMPv1
```

```
    trap, since an alarmActiveEngineId cannot be extracted from the
    SNMPv1 trap PDU.

    This object MUST always be instantiated, even if the list
    can contain alarms from only one engine."
   ::= { alarmClearEntry 5 }

alarmClearContextName OBJECT-TYPE
   SYNTAX      SnmpAdminString (SIZE(0..32))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The name of the SNMP MIB context from which the alarm came.
       For SNMPv1 traps this is the community string from the Trap.
       Note that care needs to be taken when selecting community
       strings to ensure that these can be represented as a
       well-formed SnmpAdminString.  Community or Context names
       that are not well-formed SnmpAdminStrings will be mapped
       to zero length strings.

       If the alarm's source SNMP engine is known not to support
       multiple contexts, this object is a zero length string."
   ::= { alarmClearEntry 6 }

alarmClearNotificationID OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "The NOTIFICATION-TYPE object identifier of the alarm
       clear."
   ::= { alarmClearEntry 7 }

alarmClearResourceId    OBJECT-TYPE
   SYNTAX      ResourceId
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
      "This object identifies the resource that was under alarm.

      If there is no corresponding resource, then
      the value of this object MUST be 0.0."
   ::= { alarmClearEntry 8 }

alarmClearLogIndex OBJECT-TYPE
   SYNTAX     Unsigned32 (0..4294967295)
   MAX-ACCESS read-only
   STATUS     current
```

```
   DESCRIPTION
       "This number MUST be the same as the log index of the
       applicable row in the notification log MIB, if it exists.
       If no log index applies to the trap, then this object
       MUST have the value of 0."
   ::= { alarmClearEntry 9 }

alarmClearModelPointer OBJECT-TYPE
   SYNTAX     RowPointer
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "A pointer to the corresponding row in the
       alarmModelTable for this cleared alarm."
   ::= { alarmClearEntry 10 }

-- Notifications

alarmActiveState NOTIFICATION-TYPE
 OBJECTS     { alarmActiveModelPointer,
               alarmActiveResourceId }
 STATUS      current
 DESCRIPTION
    "An instance of the alarm indicated by
    alarmActiveModelPointer has been raised
    against the entity indicated by
    alarmActiveResourceId.

    The agent must throttle the generation of
    consecutive alarmActiveState traps so that there is at
    least a two-second gap between traps of this
    type against the same alarmActiveModelPointer and
    alarmActiveResourceId.  When traps are throttled,
    they are dropped, not queued for sending at a future time.

    A management application should periodically check
    the value of alarmActiveLastChanged to detect any
    missed alarmActiveState notification-events, e.g.,
    due to throttling or transmission loss."
 ::= { alarmNotifications 2 }

alarmClearState NOTIFICATION-TYPE
   OBJECTS     { alarmActiveModelPointer,
                 alarmActiveResourceId }
   STATUS      current
   DESCRIPTION
     "An instance of the alarm indicated by
     alarmActiveModelPointer has been cleared against
```

```
     the entity indicated by alarmActiveResourceId.

    The agent must throttle the generation of
    consecutive alarmActiveClear traps so that there is at
    least a two-second gap between traps of this
    type against the same alarmActiveModelPointer and
    alarmActiveResourceId.  When traps are throttled,
    they are dropped, not queued for sending at a future time.

    A management application should periodically check
    the value of alarmActiveLastChanged to detect any
    missed alarmClearState notification-events, e.g.,
    due to throttling or transmission loss."
   ::= { alarmNotifications 3 }

-- Conformance

alarmConformance OBJECT IDENTIFIER ::= { alarmMIB 2 }

alarmCompliances OBJECT IDENTIFIER ::= { alarmConformance 1 }

alarmCompliance MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
          "The compliance statement for systems supporting
          the Alarm MIB."
      MODULE -- this module
          MANDATORY-GROUPS {
           alarmActiveGroup,
           alarmModelGroup
          }
      GROUP       alarmActiveStatsGroup
       DESCRIPTION
           "This group is optional."
      GROUP       alarmClearGroup
       DESCRIPTION
           "This group is optional."
      GROUP       alarmNotificationsGroup
       DESCRIPTION
           "This group is optional."
   ::= { alarmCompliances 1 }

alarmGroups OBJECT IDENTIFIER ::= { alarmConformance 2 }

alarmModelGroup OBJECT-GROUP
   OBJECTS {
       alarmModelLastChanged,
       alarmModelNotificationId,
```

```
       alarmModelVarbindIndex,
       alarmModelVarbindValue,
       alarmModelDescription,
       alarmModelSpecificPointer,
       alarmModelVarbindSubtree,
       alarmModelResourcePrefix,
       alarmModelRowStatus
      }
    STATUS   current
    DESCRIPTION
               "Alarm model group."
    ::= { alarmGroups 1}

alarmActiveGroup OBJECT-GROUP
        OBJECTS {
           alarmActiveLastChanged,
           alarmActiveOverflow,
           alarmActiveEngineID,
           alarmActiveEngineAddressType,
           alarmActiveEngineAddress,
           alarmActiveContextName,
           alarmActiveVariables,
           alarmActiveNotificationID,
           alarmActiveResourceId,
           alarmActiveDescription,
           alarmActiveLogPointer,
           alarmActiveModelPointer,
           alarmActiveSpecificPointer,
           alarmActiveVariableID,
           alarmActiveVariableValueType,
           alarmActiveVariableCounter32Val,
           alarmActiveVariableUnsigned32Val,
           alarmActiveVariableTimeTicksVal,
           alarmActiveVariableInteger32Val,
           alarmActiveVariableOctetStringVal,
           alarmActiveVariableIpAddressVal,
           alarmActiveVariableOidVal,
           alarmActiveVariableCounter64Val,
           alarmActiveVariableOpaqueVal
          }
          STATUS   current
          DESCRIPTION
               "Active Alarm list group."
          ::= { alarmGroups 2}

    alarmActiveStatsGroup  OBJECT-GROUP
          OBJECTS  {
                   alarmActiveStatsActives,
```

```
                    alarmActiveStatsActiveCurrent,
                    alarmActiveStatsLastRaise,
                    alarmActiveStatsLastClear
                     }
          STATUS   current
          DESCRIPTION
               "Active alarm summary group."
          ::= { alarmGroups 3}

alarmClearGroup  OBJECT-GROUP
          OBJECTS  {
   alarmClearMaximum,
   alarmClearEngineID,
   alarmClearEngineAddressType,
   alarmClearEngineAddress,
   alarmClearContextName,
   alarmClearNotificationID,
   alarmClearResourceId,
   alarmClearLogIndex,
   alarmClearModelPointer
                    }
          STATUS   current
          DESCRIPTION
               "Cleared alarm group."
          ::= { alarmGroups 4}

alarmNotificationsGroup NOTIFICATION-GROUP
   NOTIFICATIONS { alarmActiveState, alarmClearState }
   STATUS        current
   DESCRIPTION
           "The collection of notifications that can be used to
           model alarms for faults lacking pre-existing
           notification definitions."
   ::= { alarmGroups 6 }

END
```

5. ITU Alarm

5.1. Overview

This MIB module defines alarm information specific to the alarm model defined in ITU M.3100 [M.3100], X.733 [X.733], and X.736 [X.736]. This MIB module follows the modular architecture defined by the Alarm MIB, in which the generic Alarm MIB can be augmented by other alarm information defined according to more specific models that define additional behaviour and characteristics.

The ituAlarmTable contains information from the ITU Alarm Model about possible alarms in the system.

The ituAlarmActiveTable contains information from the ITU Alarm Model about alarms modelled using the ituAlarmTable that are currently occurring on the system.

The ituAlarmActiveStatsTable provides statistics on current and total alarms.

5.2. IANA Considerations

Over time, there will be a need to add new IANAITUEventType and IANAItuProbableCause enumerated values. The Internet Assigned Number Authority (IANA) is responsible for the assignment of the enumerations in these TCs.

IANAItuProbableCause value of 0 is reserved for special purposes and MUST NOT be assigned. Values of IANAItuProbableCause in the range 1 to 1023 are reserved for causes that correspond to ITU-T probable cause. All other requests for new causes will be handled on a first-come basis, with 1025.

Request should come in the form of well-formed SMI [RFC2578] for enumeration names that are unique and sufficiently descriptive.

While some effort will be taken to ensure that new enumerations do not conceptually duplicate existing enumerations it is acknowledged that the existence of conceptual duplicates in the starting probable cause list is an known industry reality.

To aid IANA in the administration of probable cause names and values, the OPS Area Director will appoint one or more experts to help review requests.

See http://www.iana.org

The following shall be used as the initial values, but the latest values for these textual conventions should be obtained from IANA:

```
IANA-ITU-ALARM-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, mib-2
       FROM SNMPv2-SMI          -- [RFC2578]
   TEXTUAL-CONVENTION
       FROM SNMPv2-TC;          -- [RFC2579]
```

```
 ianaItuAlarmNumbers MODULE-IDENTITY
     LAST-UPDATED "200409090000Z"  -- September 09, 2004
     ORGANIZATION "IANA"
     CONTACT-INFO
         "Postal:    Internet Assigned Numbers Authority
                     Internet Corporation for Assigned Names
                     and Numbers
                     4676 Admiralty Way, Suite 330
                     Marina del Rey, CA 90292-6601
                     USA

         Tel:    +1  310-823-9358
         E-Mail: iana@iana.org"
     DESCRIPTION
         "The MIB module defines the ITU Alarm
         textual convention for objects expected to require
         regular extension.

         Copyright (C) The Internet Society (2004).  The
         initial version of this MIB module was published
         in RFC 3877.  For full legal notices see the RFC
         itself.  Supplementary information may be available on:
         http://www.ietf.org/copyrights/ianamib.html"
      REVISION    "200409090000Z"  -- September 09, 2004
      DESCRIPTION
          "Initial version, published as RFC 3877."
     ::= { mib-2 119 }

IANAItuProbableCause ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "ITU-T probable cause values.  Duplicate values defined in
         X.733 are appended with X733 to ensure syntactic uniqueness.
         Probable cause value 0 is reserved for special purposes.

         The Internet Assigned Number Authority (IANA) is responsible
         for the assignment of the enumerations in this TC.
         IANAItuProbableCause value of 0 is reserved for special
         purposes and MUST NOT be assigned.

         Values of IANAItuProbableCause in the range 1 to 1023 are
         reserved for causes that correspond to ITU-T probable cause.

         All other requests for new causes will be handled on a
         first-come, first served basis and will be assigned
         enumeration values starting with 1025.

         Request should  come in the form of well-formed
```

```
        SMI [RFC2578] for enumeration names that are unique and
        sufficiently descriptive.

        While some effort will be taken to ensure that new probable
        causes do not conceptually duplicate existing probable
        causes it is acknowledged that the existence of conceptual
        duplicates in the starting probable cause list is an known
        industry reality.

        To aid IANA in the administration of probable cause names
        and values, the OPS Area Director will appoint one or more
        experts to help review requests.

        See http://www.iana.org"
    REFERENCE
        "ITU Recommendation M.3100, 'Generic Network Information
           Model', 1995
        ITU Recommendation X.733, 'Information Technology - Open
           Systems Interconnection - System Management: Alarm
           Reporting Function', 1992
        ITU Recommendation X.736, 'Information Technology - Open
           Systems Interconnection - System Management: Security
           Alarm Reporting Function', 1992"

    SYNTAX          INTEGER
            {
            -- The following probable causes were defined in M.3100
             aIS  (1),
             callSetUpFailure  (2),
             degradedSignal  (3),
             farEndReceiverFailure  (4),
             framingError  (5),
             lossOfFrame (6),
             lossOfPointer  (7),
             lossOfSignal  (8),
             payloadTypeMismatch (9),
             transmissionError (10),
             remoteAlarmInterface (11),
             excessiveBER  (12),
             pathTraceMismatch  (13),
             unavailable  (14),
             signalLabelMismatch (15),
             lossOfMultiFrame (16),
             receiveFailure (17),
             transmitFailure (18),
             modulationFailure (19),
             demodulationFailure (20),
             broadcastChannelFailure (21),
```

```
            connectionEstablishmentError (22),
            invalidMessageReceived (23),
            localNodeTransmissionError (24),
            remoteNodeTransmissionError (25),
            routingFailure (26),

 --Values 27-50 are reserved for communications alarm related
 --probable causes
 -- The following are used with equipment alarm.

            backplaneFailure (51),
            dataSetProblem  (52),
            equipmentIdentifierDuplication  (53),
            externalIFDeviceProblem  (54),
            lineCardProblem (55),
            multiplexerProblem  (56),
            nEIdentifierDuplication  (57),
            powerProblem  (58),
            processorProblem  (59),
            protectionPathFailure  (60),
            receiverFailure  (61),
            replaceableUnitMissing  (62),
            replaceableUnitTypeMismatch (63),
            synchronizationSourceMismatch  (64),
            terminalProblem   (65),
            timingProblem   (66),
            transmitterFailure  (67),
            trunkCardProblem  (68),
            replaceableUnitProblem  (69),
            realTimeClockFailure (70),
 --An equipment alarm to be issued if the system detects that the
 --real time clock has failed
            antennaFailure (71),
            batteryChargingFailure (72),
            diskFailure (73),
            frequencyHoppingFailure (74),
            iODeviceError (75),
            lossOfSynchronisation (76),
            lossOfRedundancy (77),
            powerSupplyFailure (78),
            signalQualityEvaluationFailure (79),
            tranceiverFailure (80),
            protectionMechanismFailure (81),
            protectingResourceFailure (82),
 -- Values 83-100 are reserved for equipment alarm related probable
 -- causes
 -- The following are used with environmental alarm.
            airCompressorFailure  (101),
```

```
            airConditioningFailure  (102),
            airDryerFailure   (103),
            batteryDischarging  (104),
            batteryFailure   (105),
            commercialPowerFailure  (106),
            coolingFanFailure  (107),
            engineFailure  (108),
            fireDetectorFailure  (109),
            fuseFailure  (110),
            generatorFailure  (111),
            lowBatteryThreshold (112),
            pumpFailure  (113),
            rectifierFailure  (114),
            rectifierHighVoltage  (115),
            rectifierLowFVoltage  (116),
            ventilationsSystemFailure  (117),
            enclosureDoorOpen  (118),
            explosiveGas  (119),
            fire (120),
            flood   (121),
            highHumidity  (122),
            highTemperature  (123),
            highWind  (124),
            iceBuildUp  (125),
            intrusionDetection  (126),
            lowFuel  (127),
            lowHumidity  (128),
            lowCablePressure  (129),
            lowTemperatue  (130),
            lowWater  (131),
            smoke  (132),
            toxicGas  (133),
            coolingSystemFailure (134),
            externalEquipmentFailure (135),
            externalPointFailure (136),
 -- Values 137-150 are reserved for environmental alarm related
 -- probable causes
 -- The following are used with Processing error alarm.
            storageCapacityProblem (151),
            memoryMismatch  (152),
            corruptData  (153),
            outOfCPUCycles   (154),
            sfwrEnvironmentProblem  (155),
            sfwrDownloadFailure  (156),
            lossOfRealTimel (157),
 --A processing error alarm to be issued after the system has
 --reinitialised.  This will indicate
 --to the management systems that the view they have of the managed
```

```
 --system may no longer
 --be valid.  Usage example: The managed
 --system issues this alarm after a reinitialization with severity
 --warning to inform the
 --management system about the event.  No clearing notification will
 --be sent.
             applicationSubsystemFailure (158),
             configurationOrCustomisationError (159),
             databaseInconsistency (160),
             fileError (161),
             outOfMemory (162),
             softwareError (163),
             timeoutExpired (164),
             underlayingResourceUnavailable (165),
             versionMismatch (166),
 --Values 168-200 are reserved for processing error alarm related
 -- probable causes.
             bandwidthReduced (201),
             congestion (202),
             excessiveErrorRate (203),
             excessiveResponseTime (204),
             excessiveRetransmissionRate (205),
             reducedLoggingCapability (206),
             systemResourcesOverload (207 ),
             -- The following were defined X.733
             adapterError (500),
             applicationSubsystemFailture (501),
             bandwidthReducedX733 (502),
             callEstablishmentError (503),
             communicationsProtocolError (504),
             communicationsSubsystemFailure (505),
             configurationOrCustomizationError (506),
             congestionX733 (507),
             coruptData (508),
             cpuCyclesLimitExceeded (509),
             dataSetOrModemError (510),
             degradedSignalX733 (511),
             dteDceInterfaceError (512),
             enclosureDoorOpenX733 (513),
             equipmentMalfunction (514),
             excessiveVibration (515),
             fileErrorX733 (516),
             fireDetected (517),
             framingErrorX733 (518),
             heatingVentCoolingSystemProblem (519),
             humidityUnacceptable (520),
             inputOutputDeviceError (521),
             inputDeviceError (522),
```

```
            lanError (523),
            leakDetected (524),
            localNodeTransmissionErrorX733 (525),
            lossOfFrameX733 (526),
            lossOfSignalX733 (527),
            materialSupplyExhausted (528),
            multiplexerProblemX733 (529),
            outOfMemoryX733 (530),
            ouputDeviceError (531),
            performanceDegraded (532),
            powerProblems (533),
            pressureUnacceptable (534),
            processorProblems (535),
            pumpFailureX733 (536),
            queueSizeExceeded (537),
            receiveFailureX733 (538),
            receiverFailureX733 (539),
            remoteNodeTransmissionErrorX733 (540),
            resourceAtOrNearingCapacity (541),
            responseTimeExecessive (542),
            retransmissionRateExcessive (543),
            softwareErrorX733 (544),
            softwareProgramAbnormallyTerminated (545),
            softwareProgramError (546),
            storageCapacityProblemX733 (547),
            temperatureUnacceptable (548),
            thresholdCrossed (549),
            timingProblemX733 (550),
            toxicLeakDetected (551),
            transmitFailureX733 (552),
            transmiterFailure (553),
            underlyingResourceUnavailable (554),
            versionMismatchX733 (555),
            -- The following are defined in X.736
            authenticationFailure (600),
            breachOfConfidentiality (601),
            cableTamper (602),
            delayedInformation (603),
            denialOfService (604),
            duplicateInformation (605),
            informationMissing (606),
            informationModificationDetected (607),
            informationOutOfSequence (608),
            keyExpired (609),
            nonRepudiationFailure (610),
            outOfHoursActivity (611),
            outOfService (612),
            proceduralError (613),
```

```
             unauthorizedAccessAttempt (614),
             unexpectedInformation (615),

             other (1024)
             }

IANAItuEventType ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "The ITU event Type values.

            The Internet Assigned Number Authority (IANA) is
            responsible for the assignment of the enumerations
            in this TC.

            Request should  come in the form of well-formed
            SMI [RFC2578] for enumeration names that are unique
            and sufficiently descriptive.

            See http://www.iana.org "
    REFERENCE
           "ITU Recommendation X.736, 'Information Technology - Open
            Systems Interconnection - System Management: Security
            Alarm Reporting Function', 1992"
    SYNTAX         INTEGER
           {
           other (1),
           communicationsAlarm (2),
           qualityOfServiceAlarm (3),
           processingErrorAlarm (4),
           equipmentAlarm (5),
           environmentalAlarm (6),
           integrityViolation (7),
           operationalViolation (8),
           physicalViolation (9),
           securityServiceOrMechanismViolation (10),
           timeDomainViolation (11)
           }

END
```

5.3. Textual Conventions

```
ITU-ALARM-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, mib-2
       FROM SNMPv2-SMI         -- [RFC2578]
   TEXTUAL-CONVENTION
       FROM SNMPv2-TC;         -- [RFC2579]

  ituAlarmTc MODULE-IDENTITY
      LAST-UPDATED "200409090000Z"  -- September 09, 2004
      ORGANIZATION "IETF Distributed Management Working Group"
      CONTACT-INFO
         " WG EMail: disman@ietf.org
           Subscribe: disman-request@ietf.org
           http://www.ietf.org/html.charters/disman-charter.html

           Chair:     Randy Presuhn
                      randy_presuhn@mindspring.com

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

                      Dan Romascanu
                      Avaya
                      Atidim Technology Park, Bldg. #3
                      Tel Aviv, 61131
                      Israel
                      Tel: +972-3-645-8414
                      Email: dromasca@avaya.com"
      DESCRIPTION
         "This MIB module defines the ITU Alarm
         textual convention for objects not expected to require
         regular extension.

         Copyright (C) The Internet Society (2004).  The
         initial version of this MIB module was published
         in RFC 3877.  For full legal notices see the RFC
         itself.  Supplementary information may be available on:
         http://www.ietf.org/copyrights/ianamib.html"
      REVISION    "200409090000Z"  -- September 09, 2004
      DESCRIPTION
          "Initial version, published as RFC 3877."
```

```
     ::= { mib-2 120 }

ItuPerceivedSeverity ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "ITU perceived severity values"
    REFERENCE
           "ITU Recommendation M.3100, 'Generic Network Information
            Model', 1995
            ITU Recommendation X.733, 'Information Technology - Open
            Systems Interconnection - System Management: Alarm
            Reporting Function', 1992"
    SYNTAX         INTEGER
           {
           cleared (1),
           indeterminate (2),
           critical (3),
           major (4),
           minor (5),
           warning (6)
           }

ItuTrendIndication ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "ITU trend indication values for alarms."
    REFERENCE
           "ITU Recommendation M.3100, 'Generic Network Information
            Model', 1995
            ITU Recommendation X.733, 'Information Technology - Open
            Systems Interconnection - System Management: Alarm
            Reporting Function', 1992"
    SYNTAX         INTEGER
      {
      moreSevere (1),
      noChange (2),
      lessSevere (3)
      }

END
```

5.4. Definitions

```
   ITU-ALARM-MIB DEFINITIONS ::= BEGIN

   IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE,
      Gauge32, mib-2
          FROM SNMPv2-SMI                -- [RFC2578]
      AutonomousType, RowPointer
          FROM SNMPv2-TC                 -- [RFC2579]
      SnmpAdminString
          FROM SNMP-FRAMEWORK-MIB        -- [RFC3411]
      alarmListName, alarmModelIndex,
      alarmActiveDateAndTime, alarmActiveIndex
          FROM ALARM-MIB                 -- [RFC3877]
      ItuPerceivedSeverity,
      ItuTrendIndication
          FROM ITU-ALARM-TC-MIB          -- [RFC3877]
      IANAItuProbableCause,
      IANAItuEventType
          FROM IANA-ITU-ALARM-TC-MIB     -- [RFC3877]
      MODULE-COMPLIANCE, OBJECT-GROUP
          FROM SNMPv2-CONF               -- [RFC2580]
      ZeroBasedCounter32
          FROM RMON2-MIB;                -- [RFC2021]

     ituAlarmMIB MODULE-IDENTITY
         LAST-UPDATED "200409090000Z"  -- September 09, 2004
         ORGANIZATION "IETF Distributed Management Working Group"
         CONTACT-INFO
              "WG EMail: disman@ietf.org
              Subscribe: disman-request@ietf.org
              http://www.ietf.org/html.charters/disman-charter.html

              Chair:     Randy Presuhn
                         randy_presuhn@mindspring.com

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

                         Dan Romascanu
                         Avaya
                         Atidim Technology Park, Bldg. #3
                         Tel Aviv, 61131
```

```
                        Israel
                        Tel: +972-3-645-8414
                        Email: dromasca@avaya.com"
        DESCRIPTION
                "The MIB module describes ITU Alarm information
                as defined in ITU Recommendation M.3100 [M.3100],
                X.733 [X.733] and X.736 [X.736].

                Copyright (C) The Internet Society (2004).  The
                initial version of this MIB module was published
                in RFC 3877.  For full legal notices see the RFC
                itself.  Supplementary information may be available on:
                http://www.ietf.org/copyrights/ianamib.html"
        REVISION    "200409090000Z"  -- September 09, 2004
        DESCRIPTION
            "Initial version, published as RFC 3877."
        ::= { mib-2 121 }

   ituAlarmObjects OBJECT IDENTIFIER ::= { ituAlarmMIB 1 }

   ituAlarmModel OBJECT IDENTIFIER ::= { ituAlarmObjects 1 }

   ituAlarmActive  OBJECT IDENTIFIER ::= { ituAlarmObjects 2 }

   ituAlarmTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF ItuAlarmEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A table of ITU Alarm information for possible alarms
          on the system."
      ::= { ituAlarmModel 1 }

   ituAlarmEntry OBJECT-TYPE
      SYNTAX      ItuAlarmEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Entries appear in this table whenever an entry is created
           in the alarmModelTable with a value of alarmModelState in
           the range from 1 to 6.  Entries disappear from this table
           whenever the corresponding entries are deleted from the
           alarmModelTable, including in cases where those entries
           have been deleted due to local system action.  The value of
           alarmModelSpecificPointer has no effect on the creation
           or deletion of entries in this table.  Values of
           alarmModelState map to values of ituAlarmPerceivedSeverity
           as follows:
```

```
             alarmModelState -> ituAlarmPerceivedSeverity
                    1        ->         clear (1)
                    2        ->         indeterminate (2)
                    3        ->         warning (6)
                    4        ->         minor (5)
                    5        ->         major (4)
                    6        ->         critical (3)

          All other values of alarmModelState MUST NOT appear
          in this table.

          This table MUST be persistent across system reboots."
      INDEX       { alarmListName, alarmModelIndex,
                    ituAlarmPerceivedSeverity }
      ::= { ituAlarmTable 1 }

   ItuAlarmEntry ::= SEQUENCE {
      ituAlarmPerceivedSeverity     ItuPerceivedSeverity,
      ituAlarmEventType             IANAItuEventType,
      ituAlarmProbableCause         IANAItuProbableCause,
      ituAlarmAdditionalText        SnmpAdminString,
      ituAlarmGenericModel          RowPointer }

   ituAlarmPerceivedSeverity OBJECT-TYPE
      SYNTAX  ItuPerceivedSeverity
      MAX-ACCESS   not-accessible
      STATUS       current
      DESCRIPTION
               "ITU perceived severity values."
       REFERENCE
           "ITU Recommendation M.3100, 'Generic Network Information
              Model', 1995
            ITU Recommendation X.733, 'Information Technology - Open
              Systems Interconnection - System Management: Alarm
              Reporting Function', 1992"
       ::= { ituAlarmEntry 1 }

   ituAlarmEventType OBJECT-TYPE
      SYNTAX       IANAItuEventType
      MAX-ACCESS   read-write
      STATUS       current
      DESCRIPTION
               "Represents the event type values for the alarms"
       REFERENCE
           "ITU Recommendation M.3100, 'Generic Network Information
              Model', 1995
            ITU Recommendation X.733, 'Information Technology - Open
              Systems Interconnection - System Management: Alarm
```

```
               Reporting Function', 1992
             ITU Recommendation X.736, 'Information Technology - Open
               Systems Interconnection - System Management: Security
               Alarm Reporting Function', 1992"
        ::= { ituAlarmEntry 2 }

    ituAlarmProbableCause OBJECT-TYPE
       SYNTAX      IANAItuProbableCause
       MAX-ACCESS  read-write
       STATUS       current
       DESCRIPTION
                "ITU probable cause values."
        REFERENCE
            "ITU Recommendation M.3100, 'Generic Network Information
               Model', 1995
             ITU Recommendation X.733, 'Information Technology - Open
               Systems Interconnection - System Management: Alarm
               Reporting Function', 1992
             ITU Recommendation X.736, 'Information Technology - Open
               Systems Interconnection - System Management: Security
               Alarm Reporting Function', 1992"
        ::= { ituAlarmEntry 3 }

    ituAlarmAdditionalText OBJECT-TYPE
       SYNTAX  SnmpAdminString
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
                "Represents the additional text field for the alarm."
        REFERENCE
            "ITU Recommendation M.3100, 'Generic Network Information
               Model', 1995
             ITU Recommendation X.733, 'Information Technology - Open
               Systems Interconnection - System Management: Alarm
               Reporting Function', 1992"
        ::= { ituAlarmEntry 4}

    ituAlarmGenericModel OBJECT-TYPE
       SYNTAX     RowPointer
       MAX-ACCESS read-write
       STATUS     current
       DESCRIPTION
       "This object points to the corresponding
        row in the alarmModelTable for this alarm severity.

        This corresponding entry to alarmModelTable could also
        be derived by performing the reverse of the mapping
        from alarmModelState to ituAlarmPerceivedSeverity defined
```

```
       in the description of ituAlarmEntry to determine the
       appropriate { alarmListName, alarmModelIndex, alarmModelState }
       for this { alarmListName, alarmModelIndex,
       ituAlarmPerceivedSeverity }."
      ::= { ituAlarmEntry 5 }

   -- ITU Active Alarm Table --

   ituAlarmActiveTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF ItuAlarmActiveEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A table of ITU information for active alarms entries."
      ::= { ituAlarmActive 1 }

   ituAlarmActiveEntry OBJECT-TYPE
      SYNTAX      ItuAlarmActiveEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Entries appear in this table when alarms are active.  They
          are removed when the alarm is no longer occurring."
      INDEX       { alarmListName, alarmActiveDateAndTime,
                    alarmActiveIndex }
      ::= { ituAlarmActiveTable 1 }

   ItuAlarmActiveEntry ::= SEQUENCE {
       ituAlarmActiveTrendIndication       ItuTrendIndication,
       ituAlarmActiveDetector              AutonomousType,
       ituAlarmActiveServiceProvider       AutonomousType,
       ituAlarmActiveServiceUser           AutonomousType
       }

   ituAlarmActiveTrendIndication OBJECT-TYPE
      SYNTAX      ItuTrendIndication
      MAX-ACCESS  read-only
      STATUS       current
      DESCRIPTION
               "Represents the trend indication values for the alarms."
       REFERENCE
           "ITU Recommendation M.3100, 'Generic Network Information
              Model', 1995
            ITU Recommendation X.733, 'Information Technology - Open
              Systems Interconnection - System Management: Alarm
              Reporting Function', 1992"
       ::= { ituAlarmActiveEntry 1 }
```

```
   ituAlarmActiveDetector OBJECT-TYPE
      SYNTAX AutonomousType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
         "Represents the SecurityAlarmDetector object."
       REFERENCE
           "ITU Recommendation X.736, 'Information Technology - Open
               Systems Interconnection - System Management: Security
               Alarm Reporting Function', 1992"
      ::= { ituAlarmActiveEntry 2 }

   ituAlarmActiveServiceProvider OBJECT-TYPE
      SYNTAX AutonomousType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
         "Represents the ServiceProvider object."
       REFERENCE
           "ITU Recommendation X.736, 'Information Technology - Open
               Systems Interconnection - System Management: Security
               Alarm Reporting Function', 1992"
      ::= { ituAlarmActiveEntry 3 }

   ituAlarmActiveServiceUser OBJECT-TYPE
      SYNTAX AutonomousType
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
         "Represents the ServiceUser object."
       REFERENCE
           "ITU Recommendation X.736, 'Information Technology - Open
               Systems Interconnection - System Management: Security
               Alarm Reporting Function', 1992"
      ::= { ituAlarmActiveEntry 4 }

   -- Statistics and Counters

   ituAlarmActiveStatsTable  OBJECT-TYPE
         SYNTAX  SEQUENCE OF ituAlarmActiveStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
         DESCRIPTION
            "This table represents the ITU alarm statistics
            information."
     ::= { ituAlarmActive 2 }
```

```
   ituAlarmActiveStatsEntry OBJECT-TYPE
         SYNTAX  ItuAlarmActiveStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
         DESCRIPTION
            "Statistics on the current active ITU alarms."
         INDEX   { alarmListName }

     ::= {  ituAlarmActiveStatsTable 1 }

   ItuAlarmActiveStatsEntry ::=
    SEQUENCE {
      ituAlarmActiveStatsIndeterminateCurrent Gauge32,
      ituAlarmActiveStatsCriticalCurrent      Gauge32,
      ituAlarmActiveStatsMajorCurrent         Gauge32,
      ituAlarmActiveStatsMinorCurrent         Gauge32,
      ituAlarmActiveStatsWarningCurrent       Gauge32,
      ituAlarmActiveStatsIndeterminates       ZeroBasedCounter32,
      ituAlarmActiveStatsCriticals            ZeroBasedCounter32,
      ituAlarmActiveStatsMajors               ZeroBasedCounter32,
      ituAlarmActiveStatsMinors               ZeroBasedCounter32,
      ituAlarmActiveStatsWarnings             ZeroBasedCounter32
    }

   ituAlarmActiveStatsIndeterminateCurrent OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the current number of active alarms with a
           ituAlarmPerceivedSeverity of indeterminate."
      ::= { ituAlarmActiveStatsEntry 1 }

   ituAlarmActiveStatsCriticalCurrent OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the current number of active alarms with a
           ituAlarmPerceivedSeverity of critical."
      ::= { ituAlarmActiveStatsEntry 2 }

   ituAlarmActiveStatsMajorCurrent OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the current number of active alarms with a
```

```
            ituAlarmPerceivedSeverity of major."
      ::= { ituAlarmActiveStatsEntry 3 }

   ituAlarmActiveStatsMinorCurrent OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the current number of active alarms with a
           ituAlarmPerceivedSeverity of minor."
      ::= { ituAlarmActiveStatsEntry 4 }

   ituAlarmActiveStatsWarningCurrent OBJECT-TYPE
      SYNTAX      Gauge32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the current number of active alarms with a
           ituAlarmPerceivedSeverity of warning."
      ::= { ituAlarmActiveStatsEntry 5 }

   ituAlarmActiveStatsIndeterminates OBJECT-TYPE
      SYNTAX      ZeroBasedCounter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the total number of active alarms with a
           ituAlarmPerceivedSeverity of indeterminate since system
           restart."
      ::= { ituAlarmActiveStatsEntry 6 }

   ituAlarmActiveStatsCriticals OBJECT-TYPE
      SYNTAX      ZeroBasedCounter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the total number of active alarms with a
           ituAlarmPerceivedSeverity of critical since system restart."
      ::= { ituAlarmActiveStatsEntry 7 }

   ituAlarmActiveStatsMajors OBJECT-TYPE
      SYNTAX      ZeroBasedCounter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the total number of active alarms with a
           ituAlarmPerceivedSeverity of major since system restart."
      ::= { ituAlarmActiveStatsEntry 8 }
```

```
   ituAlarmActiveStatsMinors OBJECT-TYPE
      SYNTAX      ZeroBasedCounter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the total number of active alarms with a
           ituAlarmPerceivedSeverity of minor since system restart."
      ::= { ituAlarmActiveStatsEntry 9 }

   ituAlarmActiveStatsWarnings OBJECT-TYPE
      SYNTAX      ZeroBasedCounter32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A count of the total number of active alarms with a
           ituAlarmPerceivedSeverity of warning since system restart."
      ::= { ituAlarmActiveStatsEntry 10 }

   -- Conformance

   ituAlarmConformance OBJECT IDENTIFIER ::= { ituAlarmMIB 2 }
   ituAlarmCompliances  OBJECT IDENTIFIER ::= { ituAlarmConformance 1 }

   ituAlarmCompliance MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
             "The compliance statement for systems supporting
             the ITU Alarm MIB."
      MODULE -- this module
          MANDATORY-GROUPS {
              ituAlarmGroup
              }
      GROUP       ituAlarmServiceUserGroup
          DESCRIPTION
              "This group is optional."
      GROUP       ituAlarmSecurityGroup
          DESCRIPTION
              "This group is optional."
      GROUP       ituAlarmStatisticsGroup
          DESCRIPTION
              "This group is optional."
     ::= { ituAlarmCompliances 1 }

   ituAlarmGroups OBJECT IDENTIFIER ::= { ituAlarmConformance 2 }

   ituAlarmGroup OBJECT-GROUP
    OBJECTS {
              ituAlarmEventType,
```

```
            ituAlarmProbableCause,
            ituAlarmGenericModel
          }
    STATUS   current
    DESCRIPTION
                "ITU alarm details list group."
    ::= { ituAlarmGroups 1}

   ituAlarmServiceUserGroup OBJECT-GROUP
    OBJECTS {
            ituAlarmAdditionalText,
            ituAlarmActiveTrendIndication
          }
    STATUS current
    DESCRIPTION
          "The use of these parameters is a service-user option."
    ::= { ituAlarmGroups 2 }

   ituAlarmSecurityGroup OBJECT-GROUP
     OBJECTS {
            ituAlarmActiveDetector,
            ituAlarmActiveServiceProvider,
            ituAlarmActiveServiceUser
          }
     STATUS current
     DESCRIPTION
          "Security Alarm Reporting Function"
       REFERENCE
         "ITU Recommendation X.736, 'Information Technology - Open
             Systems Interconnection - System Management: Security
             Alarm Reporting Function', 1992"
     ::= { ituAlarmGroups 3 }

   ituAlarmStatisticsGroup OBJECT-GROUP
     OBJECTS {
           ituAlarmActiveStatsIndeterminateCurrent,
           ituAlarmActiveStatsCriticalCurrent,
           ituAlarmActiveStatsMajorCurrent,
           ituAlarmActiveStatsMinorCurrent,
           ituAlarmActiveStatsWarningCurrent,
           ituAlarmActiveStatsIndeterminates,
           ituAlarmActiveStatsCriticals,
           ituAlarmActiveStatsMajors,
           ituAlarmActiveStatsMinors,
           ituAlarmActiveStatsWarnings
            }
     STATUS current
     DESCRIPTION
```

```
       "ITU Active Alarm Statistics."
     ::= { ituAlarmGroups 4 }

   END
```

6. Examples

6.1. Alarms Based on linkUp/linkDown Notifications

This example demonstrates an interface-based alarm that goes into a state of "warning" when a linkDown Notification [RFC2863] occurs but the ifAdminStatus indicates the interface was taken down administratively. If IfAdminStatus is "up" when the linkDown Notification occurs, then there is a problem, so the state of the alarm is critical. A linkUp alarm clears the alarm.

```
linkDown NOTIFICATION-TYPE
        OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
        STATUS  current
        DESCRIPTION
            ""
    ::= { snmpTraps 3 }

linkUp NOTIFICATION-TYPE
        OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
        STATUS  current
        DESCRIPTION
            ""
    ::= { snmpTraps 4 }
```

```
 alarmModelIndex                3
 alarmModelState                1
 alarmModelNotificationId       linkUp
 alarmModelVarbindIndex         0
 alarmModelVarbindValue         0
 alarmModelDescription          "linkUp"
 alarmModelSpecificPointer      ituAlarmEntry.3.1
 alarmModelVarbindSubtree       ifIndex (1.3.6.1.2.1.2.2.1.1)
 alarmModelResourcePrefix       0.0
 alarmModelRowStatus            active (1)
 ituAlarmEventType              communicationsAlarm (2)
 ituAlarmPerceivedSeverity      cleared (1)
 ituAlarmGenericModel           alarmModelEntry.3.1

 alarmModelIndex                3
 alarmModelState                2
 alarmModelNotificationId       linkDown
 alarmModelVarbindIndex         2
```

```
alarmModelVarbindValue          down (2)
alarmModelDescription           "linkDown administratively"
alarmModelSpecificPointer       ituAlarmEntry.3.6
alarmModelVarbindSubtree        ifIndex (1.3.6.1.2.1.2.2.1.1)
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               communicationsAlarm (2)
ituAlarmPerceivedSeverity       warning (6)
ituAlarmGenericModel            alarmModelEntry.3.2

alarmModelIndex                 3
alarmModelState                 3
alarmModelNotificationId        linkDown
alarmModelVarbindIndex          2
alarmModelVarbindValue          up (1)
alarmModelDescription           "linkDown - confirmed problem"
alarmModelSpecificPointer       ituAlarmEntry.3.3
alarmModelVarbindSubtree        ifIndex (1.3.6.1.2.1.2.2.1.1)
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               communicationsAlarm (2)
ituAlarmPerceivedSeverity       critical (3)
ituAlarmGenericModel            alarmModelEntry.3.3

alarmActiveIndex                1
alarmActiveDateAndTime          2342464573
alarmActiveDateAndTime          DateAndTime,
alarmActiveEngineID             SnmpEngineID,
alarmActiveEngineAddressType    ipV4
alarmActiveEngineAddress        10.10.10.10
alarmActiveContextName          SnmpAdminString,
alarmActiveVariables            3
alarmActiveNotificationID       1.3.6.1.6.3.1.1.5.3
alarmActiveResourceId           1.3.6.1.2.1.2.2.1.1.346
alarmActiveLogPointer           0.0
alarmActiveModelPointer         alarmModelEntry.3.3
alarmActiveSpecificPointer      ituAlarmActiveEntry.1.3
ituAlarmActiveTrendIndication   moreSevere (1)
ituAlarmDetector                  0.0
ituAlarmServiceProvider           0.0
ituAlarmServiceUser               0.0

alarmActiveVariableIndex                1
alarmActiveVariableID                   sysUpTime.0
alarmActiveVariableValueType            timeTicks(3)
alarmActiveVariableCounter32Val         0
alarmActiveVariableUnsigned32Val        0
alarmActiveVariableTimeTicksVal         46754
```

```
 alarmActiveVariableInteger32Val          0
 alarmActiveVariableOctetStringVal        ""
 alarmActiveVariableIpAddressVal          0
 alarmActiveVariableOidVal                0.0
 alarmActiveVariableCounter64Val          0
 alarmActiveVariableIndex                 2
 alarmActiveVariableID                    snmpTrapOID.0
 alarmActiveVariableValueType             objectId(7)
 alarmActiveVariableCounter32Val          0
 alarmActiveVariableUnsigned32Val         0
 alarmActiveVariableTimeTicksVal          0
 alarmActiveVariableInteger32Val          0
 alarmActiveVariableOctetStringVal        ""
 alarmActiveVariableIpAddressVal          0
 alarmActiveVariableOidVal                1.3.6.1.6.3.1.1.5.3
 alarmActiveVariableCounter64Val          0
 alarmActiveVariableIndex                 3
 alarmActiveVariableID                    ifIndex
 alarmActiveVariableValueType             integer32(4)
 alarmActiveVariableCounter32Val          0
 alarmActiveVariableUnsigned32Val         0
 alarmActiveVariableTimeTicksVal          0
 alarmActiveVariableInteger32Val          346
 alarmActiveVariableOctetStringVal        ""
 alarmActiveVariableIpAddressVal          0
 alarmActiveVariableOidVal                0.0
 alarmActiveVariableCounter64Val          0
 alarmActiveVariableIndex                 4
 alarmActiveVariableID                    ifAdminStatus
 alarmActiveVariableValueType             integer32(4)
 alarmActiveVariableCounter32Val          0
 alarmActiveVariableUnsigned32Val         0
 alarmActiveVariableTimeTicksVal          0
 alarmActiveVariableInteger32Val          up (1)
 alarmActiveVariableOctetStringVal        ""
 alarmActiveVariableIpAddressVal          0
 alarmActiveVariableOidVal                0.0
 alarmActiveVariableCounter64Val          0
 alarmActiveVariableIndex                 5
 alarmActiveVariableID                    ifOperStatus
 alarmActiveVariableValueType             integer32(4)
 alarmActiveVariableCounter32Val          0
 alarmActiveVariableUnsigned32Val         0
 alarmActiveVariableTimeTicksVal          0
 alarmActiveVariableInteger32Val          down(2)
 alarmActiveVariableOctetStringVal        ""
 alarmActiveVariableIpAddressVal          0
 alarmActiveVariableOidVal                0.0
```

```
 alarmActiveVariableCounter64Val          0
 alarmActiveVariableOpaqueVal
```

6.2. Temperature Alarms Using Generic Notifications

Consider a system able to detect four different temperature states for a widget - normal, minor, major, critical. The system does not have any Notification definitions for these alarm states. A temperature alarm can be modelled using the generic alarm Notifications of alarmClearState and alarmActive.

```
alarmModelIndex                 5
alarmModelState                 1
alarmModelNotificationId        alarmClearState
alarmModelVarbindIndex          2
alarmModelVarbindValue          cleared (1)
alarmModelDescription           "Acme Widget Temperature Normal"
alarmModelSpecificPointer       ituAlarmEntry.5.1
alarmModelVarbindSubtree        alarmActiveResourceId
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            cleared (1)
ituAlarmGenericModel            alarmModelEntry.5.1

alarmModelIndex                 5
alarmModelState                 2
alarmModelNotificationId        alarmActiveState
alarmModelVarbindIndex          2
alarmModelVarbindValue          minor (5)
alarmModelDescription           "Acme Widget Temperature Minor"
alarmModelSpecificPointer       ituAlarmEntry.5.5
alarmModelVarbindSubtree        alarmActiveResourceId
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventState              environmentalAlarm (6)
ituPerceivedSeverity            minor (5)
ituAlarmGenericModel            alarmModelEntry.5.2

alarmModelIndex                 5
alarmModelState                 3
alarmModelNotificationId        alarmActiveState
alarmModelVarbindIndex          2
alarmModelVarbindValue          major (4)
alarmModelDescription           "Acme Widget Temperature Major"
alarmModelSpecificPointer       ituAlarmEntry.5.4
alarmModelVarbindSubtree        alarmActiveResourceId
alarmModelResourcePrefix        0.0
```

```
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            major (4)
ituAlarmGenericModel            alarmModelEntry.5.3
alarmModelIndex                 5
alarmModelState                 4
alarmModelNotificationId        alarmActiveState
alarmModelVarbindIndex          2
alarmModelVarbindValue          critical (3)
alarmModelDescription           "Acme Widget Temperature Critical"
alarmModelSpecificPointer       ituAlarmEntry.5.3
alarmModelVarbindSubtree        alarmActiveResourceId
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            critical (3)
ituAlarmGenericModel            alarmModelEntry.5.4
```

6.3. Temperature Alarms Without Notifications

Consider a system able to detect four different temperature states for a widget - normal, minor, major, critical. The system does not have any Notification definitions for these alarm states. A temperature alarm can be modelled without specifying any Notifications in the alarm model. When a temperature state other than normal is detected, an instance of this alarm would be added to the active alarm table, but no Notifications would be sent out.

This could alternatively be accomplished using the models from example 6.2 and by not specifying any target managers in the SNMP-TARGET-MIB, which would allow the alarm state Notifications to be logged in the Notification Log while still preventing Notifications from being transmitted on the wire.

```
alarmModelIndex                 6
alarmModelState                 1
alarmModelNotificationId        0.0
alarmModelVarbindIndex          0
alarmModelVarbindValue          0
alarmModelDescription           "Widget Temperature"
alarmModelSpecificPointer       ituAlarmEntry.6.1
alarmModelVarbindSubtree        0.0
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            cleared (1)
ituAlarmGenericModel            alarmModelEntry.6.1
```

```
alarmModelIndex                 6
alarmModelState                 2
alarmModelNotificationId        0.0
alarmModelVarbindIndex          0
alarmModelVarbindValue          0
alarmModelDescription           "Widget Temperature"
alarmModelSpecificPointer       ituAlarmEntry.6.5
alarmModelVarbindSubtree        0.0
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventState              environmentalAlarm (6)
ituAlarmPerceivedSeverity       minor (5)
ituAlarmGenericModel            alarmModelEntry.6.2

alarmModelIndex                 6
alarmModelState                 3
alarmModelNotificationId        0.0
alarmModelVarbindIndex          0
alarmModelVarbindValue          0
alarmModelDescription           "Widget Temperature"
alarmModelSpecificPointer       ituAlarmEntry.6.4
alarmModelVarbindSubtree        0.0
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            major (4)
ituAlarmGenericModel            alarmModelEntry.6.3

alarmModelIndex                 6
alarmModelState                 4
alarmModelNotificationId        0.0
alarmModelVarbindIndex          0
alarmModelVarbindValue          0
alarmModelDescription           "Widget Temperature Severe"
alarmModelSpecificPointer       ituAlarmEntry.6.3
alarmModelVarbindSubtree        0.0
alarmModelResourcePrefix        0.0
alarmModelRowStatus             active (1)
ituAlarmEventType               environmentalAlarm (6)
ituPerceivedSeverity            critical (3)
ituAlarmGenericModel            alarmModelEntry.6.4
```

6.4. Printer MIB Alarm Example

Consider the following Notifications defined in the printer MIB [RFC3805]:

```
prtAlertSeverityLevel OBJECT-TYPE
 -- This value is a type 1 enumeration
 SYNTAX     INTEGER {
              other(1),
              critical(3),
              warning(4)
            }
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
   "The level of severity of this alert table entry.  The printer
   determines the severity level assigned to each entry into the
   table."
 ::= { prtAlertEntry 2 }

printerV2Alert NOTIFICATION-TYPE
 OBJECTS { prtAlertIndex, prtAlertSeverityLevel, prtAlertGroup,
         prtAlertGroupIndex, prtAlertLocation, prtAlertCode }
 STATUS  current
 DESCRIPTION
   "This trap is sent whenever a critical event is added to the
   prtAlertTable."
 ::= { printerV2AlertPrefix 1 }
```

These Notifications can be used to model a printer alarm as follows:

```
alarmModelIndex                  9 alarmModelState                1
alarmModelNotificationId         alarmClearState
alarmModelVarbindIndex           0 alarmModelVarbindValue         0
alarmModelDescription            "Printer Alarm"
alarmModelSpecificPointer        0.0 alarmModelVarbindSubtree
prtAlertGroup alarmModelResourcePrefix         0.0
alarmModelRowStatus              active (1)

alarmModelIndex                  9 alarmModelState                2
alarmModelNotificationId         printerV2Alert
alarmModelVarbindIndex           2 alarmModelVarbindValue
warning (4) alarmModelDescription            "Printer Alarm"
alarmModelSpecificPointer        0.0 alarmModelVarbindSubtree
prtAlertGroup alarmModelResourcePrefix         0.0
alarmModelRowStatus              active (1)

alarmModelIndex                  9 alarmModelState                3
```

```
alarmModelNotificationId         printerV2Alert
alarmModelVarbindIndex           2 alarmModelVarbindValue
other (1) alarmModelDescription            "Printer Alarm - unknown
severity" alarmModelSpecificPointer        0.0
alarmModelVarbindSubtree         prtAlertGroup
alarmModelResourcePrefix         0.0 alarmModelRowStatus
active (1)

alarmModelIndex                  9 alarmModelState                  4
alarmModelNotificationId         printerV2Alert
alarmModelVarbindIndex           2 alarmModelVarbindValue
critical (3) alarmModelDescription            "Printer Alarm"
alarmModelSpecificPointer        0.0 alarmModelVarbindSubtree
prtAlertGroup alarmModelResourcePrefix         0.0
alarmModelRowStatus              active (1)
```

6.5. RMON Alarm Example

The RMON MIB [RFC2819] defines a mechanism for generating threshold alarms. When the thresholds are crossed, RisingAlarm and FallingAlarm Notifications are generated as appropriate. These Notifications can be used to model an upper threshold alarm as follows:

```
alarmModelIndex                  6
alarmModelState                  1
alarmModelNotificationId         FallingAlarm
alarmModelVarbindIndex           0
alarmModelVarbindValue           0
alarmModelDescription            "RMON Rising Clear Alarm"
alarmModelSpecificPointer        0.0
alarmModelVarbindSubtree         alarmIndex
alarmModelResourcePrefix         0.0
alarmModelRowStatus              active (1)

alarmModelIndex                  6
alarmModelState                  2
alarmModelNotificationId         RisingAlarm
alarmModelVarbindIndex           0
alarmModelVarbindValue           0
alarmModelDescription            "RMON Rising Alarm"
alarmModelSpecificPointer        0.0
alarmModelVarbindSubtree         alarmIndex
alarmModelResourcePrefix         0.0
alarmModelRowStatus              active (1)
```

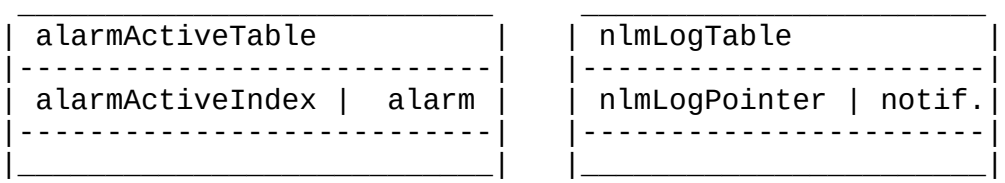
6.6. The Lifetime of an Alarm

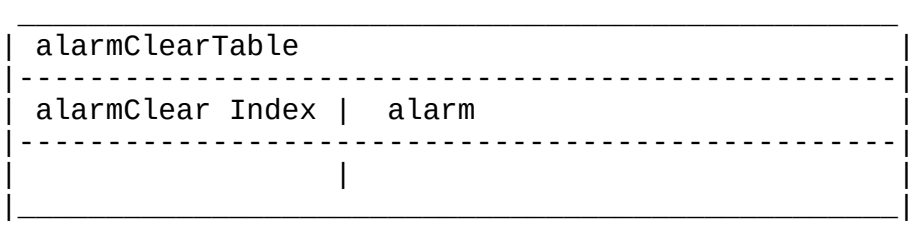
The following example demonstrates the relationship between the active alarm table, the clear alarm table and the Notification Log MIB.

Consider a system with alarms modelled as in example 1 and which also supports the informational Notification dsx3LineStatusChange.

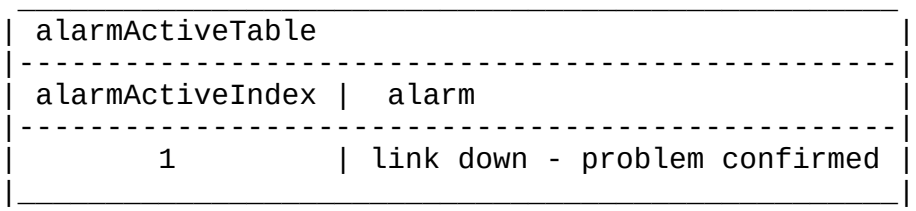
```
dsx3LineStatusChange NOTIFICATION-TYPE
    OBJECTS { dsx3LineStatus,
              dsx3LineStatusLastChange }
    STATUS  current
    DESCRIPTION
            "A dsx3LineStatusChange trap is sent when the
            value of an instance of dsx3LineStatus changes.  It
            can be utilized by an NMS to trigger polls.  When
            the line status change results in a lower level
            line status change (i.e., ds1), then no traps for
            the lower level are sent."
               ::= { ds3Traps 0 1 }
```

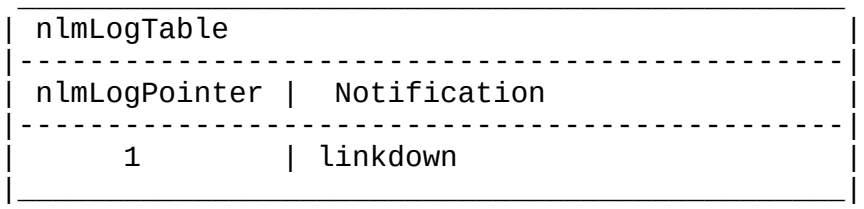
0. At system start, the active alarm table, alarm clear table and the Notification Log are all empty.

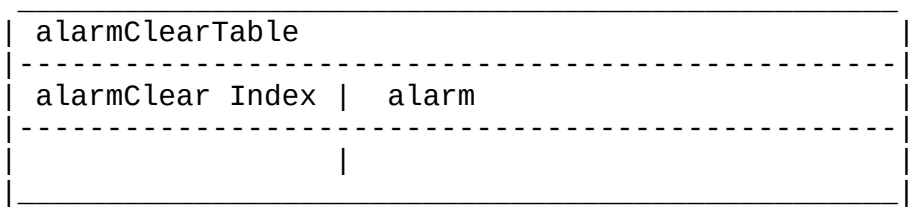




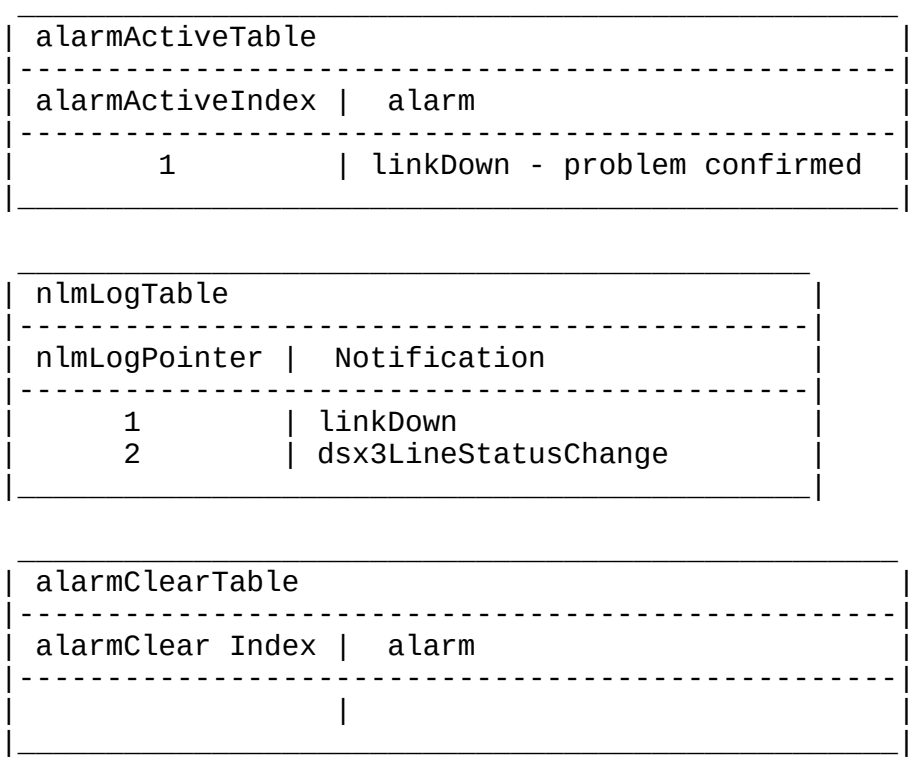
1. Some time later, a link goes down generating a linkDown Notification, which is sent out and logged in the Notification Log. As this Notification is modelled as an alarm state, an entry is added to the active alarm table.



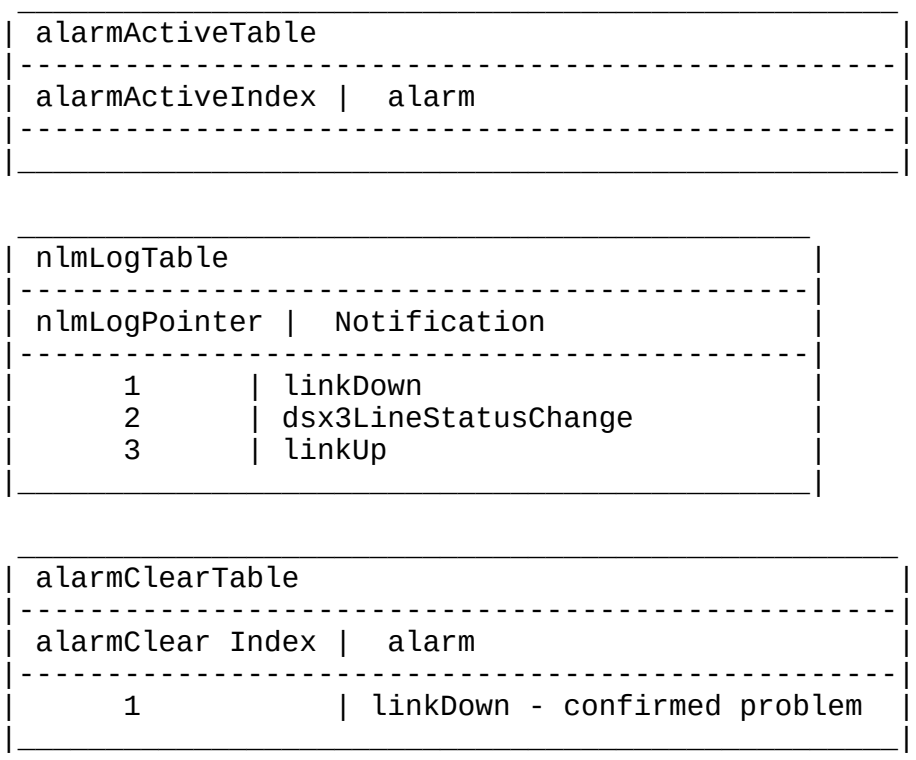




2. Some time later, the value of an instance of dsx3LineStatus changes. This Notification is sent out and logged. As this is not modelled into an alarm state, the active alarm table remains unchanged.



3. Some time later, the link goes back up. A linkUp Notification is sent out and logged. As this Notification models the clear alarm for this alarm, the alarm entry is remove from the active alarm table. An entry is added to the clear alarm table.



alarmActiveTable	
alarmActiveIndex	alarm

nlmLogTable	
nlmLogPointer	Notification
1	linkDown
2	dsx3LineStatusChange
3	linkUp

alarmClearTable	
alarmClear Index	alarm
1	linkDown - confirmed problem

7. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. 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 objects are defined with a MAX-ACCESS clause of read-write or read-create: alarmModelNotificationId, alarmModelVarbindIndex, alarmModelVarbindValue, alarmModelDescription, alarmModelSpecificPointer, alarmModelVarbindSubtree, alarmModelResourcePrefix, alarmModelRowStatus, alarmClearMaximum, ituAlarmEventType, ituAlarmProbableCause, ituAlarmAdditionalText, and ituAlarmGenericModel.

Note that setting the value of alarmClearMaximum too low may result in security related alarms history being prematurely lost.

Changing values of alarmModelRowStatus as part of creating and deleting rows in the alarmModelTable result in adding new alarm models to the system or taking them out respectively. These operations need to be carefully planned. Adding a new model should be made in a consistent manner to avoid the system overflow with alarms. Taking out a model should result in the deletion of all this model's related alarms in the system.

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

Note that the alarm throttling mechanism associated with the alarmActiveState and alarmActiveClear notifications only applies to a given alarm. Defining multiple alarms from the same internal stimulus may then still result in a flood of alarms into the network.

Although the use of community strings in SNMPv1 is not considered an effective means of providing security, security administrators SHOULD consider whether the fact that alarmActiveContextName can reveal community string values would make this object sensitive in their environment.

This MIB module can provide access to information that may also be accessed through manipulation of the SNMP-NOTIFICATION-MIB and the NOTIFICATION-LOG-MIB. This is expressed in part through the common indexing structure of nlmLogName [RFC3014], snmpNotifyFilterProfileName [RFC3413], and alarmListName. Consequently, it is RECOMMENDED that security administrators take care to configure a coherent VACM security policy. The objects

alarmActiveLogPointer, alarmActiveModelPointer, alarmActiveSpecificPointer, and alarmClearModelPointer are object identifiers that reference information to which a particular user might not be given direct access. The structure of these object identifiers does not permit the extraction of any sensitive information. Two other objects, alarmClearResourceId, and alarmActiveResourceId, are also syntactically object identifiers, but their structure could provide a user with potentially useful information to which he or she might not otherwise be granted access, such as the existence of a particular resource.

For further discussion of security, see section 3.4.

8. Acknowledgements

This document is a product of the DISMAN Working Group.

9. References

9.1. Normative References

[M.3100] ITU Recommendation M.3100, "Generic Network Information Model", 1995

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol (SNMP)", STD 15, RFC 1157, May 1990.

[RFC1215] Rose, M., "A Convention for defining traps for use with the SNMP", RFC 1215, March 1991.

[RFC2021] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 using SMIv2", January 1997.

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

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

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

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

[RFC3291] Daniele, M., Haberman, B., Routhier, S. and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 3291, May 2002.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3413] Levi, D., Meyer, P. and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3414, December 2002.

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

[RFC3416] Presuhn, R., Ed., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC3584] Frye, R., Levi, D., Routhier, S. and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework", BCP 74, RFC 3584, August 2003.

[X.733] ITU Recommendation X.733, "Information Technology - Open Systems Interconnection - System Management: Alarm Reporting Function", 1992.

[X.736] ITU Recommendation X.736, "Information Technology - Open Systems Interconnection - System Management: Security Alarm Reporting Function", 1992.

9.2 Informative References

[RFC1657] Willis, S., Burruss, J. and J. Chu, Ed., "Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2", RFC 1657, July 1994.

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

[RFC2819] Waldbusser, S. "Remote Network Monitoring Management Information Base", STD 59, RFC 2819, May 2000.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIv2", RFC 2863, June 2000.

[RFC2981] Kavasseri, R., Ed., "Event MIB", RFC 2981, October 2000.

[RFC3014] Kavasseri, R., "Notification Log MIB", RFC 3014, November 2000.

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

[RFC3418] Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC3805] Bergman, R., Lewis, H. and I. McDonald, "Printer MIB v2", RFC 3805, June 2004.

10. Authors' Addresses

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

EMail: schishol@nortelnetworks.com

Dan Romascanu
Avaya
Atidim Technology Park, Bldg. #3
Tel Aviv, 61131
Israel

Phone: +972-3-645-8414
EMail: dromasca@avaya.com

11. Full Copyright Statement

Intellectual Property

Acknowledgement

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