Notification Log MIB

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for logging Simple Network Management Protocol (SNMP) Notifications.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.
1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].
A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

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

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

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

2. Overview

Systems that support SNMP often need a mechanism for recording Notification information as a hedge against lost Notifications, whether those are Traps or Informs [RFC1905] that exceed retransmission limits. This MIB therefore provides common infrastructure for other MIBs in the form of a local logging function. It is intended primarily for senders of Notifications but could be used also by receivers.

Given the Notification Log MIB, individual MIBs bear less responsibility to record the transient information associated with an event against the possibility that the Notification message is lost, and applications can poll the log to verify that they have not missed important Notifications.

2.1. Environment

The overall environmental concerns for the MIB are:

- SNMP Engines and Contexts
- Security
2.1.1. SNMP Engines and Contexts

There are two distinct information flows from multiple notification originators that one may log. The first is the notifications that are received (from one or more SNMP engines) for logging as SNMP informs and traps. The other comprises notifications delivered to an SNMP engine at the interface to the notification originator (using a notification mechanism other than SNMP informs or traps). The latter information flow (using a notification mechanism other than SNMP informs or traps) is modeled here as the SNMP engine (which maintains the log) sending a notification to itself. The remainder of this section discusses the handling of the former information flow – notifications (received in the form of SNMP informs or traps) from multiple SNMP engines.

As described in the SNMP architecture [RFC2571], a given system may support multiple SNMP engines operating independently of one another, each with its own SNMP engine identification. Furthermore, within the purview of a given engine there may be multiple named management contexts supporting overlapping or disjoint sets of MIB objects and Notifications. Thus, understanding a particular Notification requires knowing the SNMP engine and management context from whence it came.

To provide the necessary source information for a logged Notification, the MIB includes objects to record that Notification’s source SNMP engine ID and management context name.

2.1.2. Security

Security for Notifications is awkward since access control for the objects in the Notification 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 [RFC2571] 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 [RFC2573].

The Notification Log MIB has the notion of a "named log." By using log names and view-based access control [RFC2575] a network administrator can provide different access for different users. When an application creates a named log the security credentials of the creator stay associated with that log.
A managed system with fewer resources MAY disallow the creation of named logs, providing only the default, null-named log. Such a log has no implicit security credentials for Notification object access control and Notifications are put into it with no further checking.

When putting locally-generated Notifications into a named log, the managed system MUST use the security credentials associated with that log and MUST apply the same access control rules as described for a Notification Originator in [RFC2573].

The managed system SHOULD NOT apply access control when adding remotely-generated Notifications into either a named log or the default, null-named log. In those cases the security of the information in the log SHOULD be left to the normal, overall access control for the log itself.

The Notification Log MIB allows applications to set the maximum number of Notifications that can be logged, using nlmConfigGlobalEntryLimit. Similarly, an application can set the maximum age using nlmConfigGlobalAgeOut, after which older Notifications MAY be timed out. Please be aware that contention between multiple applications trying to set these objects to different values MAY affect the reliability and completeness of data seen by each application, i.e., it is possible that one application may change the value of either of these objects, resulting in some Notifications being deleted before the other applications have had a chance to see them. This could be used to orchestrate a denial-of-service attack. Methods for countering such an attack are for further study.

2.2. Structure

The MIB has the following sections:

- Configuration -- control over how much the log can hold and what Notifications are to be logged.
- Statistics -- indications of logging activity.
- Log -- the Notifications themselves.

2.2.1. Configuration

The configuration section contains objects to manage resource use by the MIB.

This section also contains a table to specify what logs exist and how they operate. Deciding which Notifications are to be logged depends
on filters defined in the snmpNotifyFilterTable in the standard
SNMP Notification MIB [RFC2573] identified by the initial index
(snmpNotifyFilterName) from that table.

2.2.2. Statistics

The statistics section contains counters for Notifications logged and
discarded, supplying a means to understand the results of log
capacity configuration and resource problems.

2.2.3. Log

The log contains the Notifications and the objects that came in their
variable binding list, indexed by an integer that reflects when the
entry was made. An application that wants to collect all logged
Notifications or to know if it may have missed any can keep track of
the highest index it has retrieved and start from there on its next
poll, checking sysUpTime for a discontinuity that would have reset
the index and perhaps have lost entries.

Variables are in a table indexed by Notification index and variable
index within that Notification. The values are kept as a
"discriminated union," with one value object per variable. Exactly
which value object is instantiated depends on the SNMP data type of
the variable, with a separate object of appropriate type for each
distinct SNMP data type.

An application can thus reconstruct the information from the
Notification PDU from what is recorded in the log.

2.3. Example

Following is an example configuration of a named log for logging only
linkUp and linkDown Notifications.

In nlmConfigLogTable:

    nlmConfigLogFilterName.5."links"    = "link-status"
    nlmConfigLogEntryLimit.5."links"    = 0
    nlmConfigLogAdminStatus.5."links"   = enabled
    nlmConfigLogOperStatus.5."links"    = operational
    nlmConfigLogStorageType.5."links"   = nonVolatile
    nlmConfigLogEntryStatus.5."links"   = active

Note that snmpTraps is:

    iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.5
3. Definitions

NOTIFICATION-LOG-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY, OBJECT-TYPE,
    Integer32, Unsigned32,
    TimeTicks, Counter32, Counter64,
    IpAddress, Opaque, mib-2 FROM SNMPv2-SMI
    TimeStamp, DateAndTime,
    StorageType, RowStatus,
    TAddress, TDomain FROM SNMPv2-TC
    SnmpAdminString, SnmpEngineID FROM SNMP-FRAMEWORK-MIB

MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF;

notificationLogMIB MODULE-IDENTITY
    LAST-UPDATED "200011270000Z" -- 27 November 2000
    ORGANIZATION "IETF Distributed Management Working Group"
    CONTACT-INFO "Ramanathan Kavasseri
        Cisco Systems, Inc.
        170 West Tasman Drive,
        San Jose CA 95134-1706.
        Phone: +1 408 527 2446
        Email: ramk@cisco.com"

    DESCRIPTION "The MIB module for logging SNMP Notifications, that is, Traps"
and Informs."

-- Revision History

REVISION "200011270000Z" -- 27 November 2000
DESCRIPTION "This is the initial version of this MIB.
Published as RFC 3014"
 ::= { mib-2 92 }

notificationLogMIBObjects OBJECT IDENTIFIER ::= { notificationLogMIB 1 }

nlmConfig OBJECT IDENTIFIER ::= { notificationLogMIBObjects 1 }
nlmStats OBJECT IDENTIFIER ::= { notificationLogMIBObjects 2 }
nlmLog OBJECT IDENTIFIER ::= { notificationLogMIBObjects 3 }

-- -- Configuration Section --

nlmConfigGlobalEntryLimit OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The maximum number of notification entries that may be held in nlmLogTable for all nlmLogNames added together. A particular setting does not guarantee that much data can be held.

If an application changes the limit while there are Notifications in the log, the oldest Notifications MUST be discarded to bring the log down to the new limit - thus the value of nlmConfigGlobalEntryLimit MUST take precedence over the values of nlmConfigGlobalAgeOut and nlmConfigLogEntryLimit, even if the Notification being discarded has been present for fewer minutes than the value of nlmConfigGlobalAgeOut, or if the named log has fewer entries than that specified in nlmConfigLogEntryLimit.

A value of 0 means no limit.

Please be aware that contention between multiple managers trying to set this object to different values MAY affect the reliability and completeness of data seen by each manager."
DEFVAL { 0 }
 ::= { nlmConfig 1 }

nlmConfigGlobalAgeOut OBJECT-TYPE
SYNTAX Unsigned32
UNITS "minutes"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The number of minutes a Notification SHOULD be kept in a log before it is automatically removed.

If an application changes the value of nlmConfigGlobalAgeOut, Notifications older than the new time MAY be discarded to meet the new time.

A value of 0 means no age out.

Please be aware that contention between multiple managers trying to set this object to different values MAY affect the reliability and completeness of data seen by each manager."
DEFVAL { 1440 } -- 24 hours
::= { nlmConfig 2 }

--
-- Basic Log Configuration Table
--

nlmConfigLogTable OBJECT-TYPE
SYNTAX SEQUENCE OF NlmConfigLogEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A table of logging control entries."
::= { nlmConfig 3 }

NlmConfigLogEntry OBJECT-TYPE
SYNTAX NlmConfigLogEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A logging control entry. Depending on the entry’s storage type entries may be supplied by the system or created and deleted by applications using nlmConfigLogEntryStatus."
INDEX { nlmLogName }
::= { nlmConfigLogTable 1 }

NlmConfigLogEntry ::= SEQUENCE {
  nlmLogName           SnmpAdminString,
  nlmConfigLogFilterName    SnmpAdminString,
  nlmConfigLogEntryLimit    Unsigned32,
  nlmConfigLogAdminStatus   INTEGER,
nlmConfigLogOperStatus INTEGER,
nlmConfigLogStorageType StorageType,
nlmConfigLogEntryStatus RowStatus
}

nlmLogName OBJECT-TYPE
SYNTAX SnmpAdminString (SIZE(0..32))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The name of the log.

An implementation may allow multiple named logs, up to some
implementation-specific limit (which may be none). A
zero-length log 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 logs."
::= { nlmConfigLogEntry 1 }

nlmConfigLogFilterName OBJECT-TYPE
SYNTAX SnmpAdminString (SIZE(0..32))
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"A value of snmpNotifyFilterProfileName as used as an index
into the snmpNotifyFilterTable in the SNMP Notification MIB,
specifying the locally or remotely originated Notifications
to be filtered out and not logged in this log.

A zero-length value or a name that does not identify an
existing entry in snmpNotifyFilterTable indicate no
Notifications are to be logged in this log."
DEFVAL { ''H }
::= { nlmConfigLogEntry 2 }

nlmConfigLogEntryLimit OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The maximum number of notification entries that can be held in
nlmLogTable for this named log. A particular setting does not
guarantee that that much data can be held.

If an application changes the limit while there are
Notifications in the log, the oldest Notifications are discarded
to bring the log down to the new limit."
A value of 0 indicates no limit.

Please be aware that contention between multiple managers trying to set this object to different values MAY affect the reliability and completeness of data seen by each manager.

DEFVAL { 0 }

::= { nlmConfigLogEntry 3 }  

nlmConfigLogAdminStatus OBJECT-TYPE
SYNTAX     INTEGER { enabled(1), disabled(2) }
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"Control to enable or disable the log without otherwise disturbing the log’s entry.

Please be aware that contention between multiple managers trying to set this object to different values MAY affect the reliability and completeness of data seen by each manager."

DEFVAL { enabled }

::= { nlmConfigLogEntry 4 }  

nlmConfigLogOperStatus OBJECT-TYPE
SYNTAX     INTEGER { disabled(1), operational(2), noFilter(3) }
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The operational status of this log:

  disabled  administratively disabled

  operational administratively enabled and working

  noFilter   administratively enabled but either
              nlmConfigLogFilterName is zero length
              or does not name an existing entry in
              snmpNotifyFilterTable"

::= { nlmConfigLogEntry 5 }  

nlmConfigLogStorageType OBJECT-TYPE
SYNTAX     StorageType
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"The storage type of this conceptual row."

::= { nlmConfigLogEntry 6 }  

nlmConfigLogEntryStatus OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"Control for creating and deleting entries. Entries may be
modified while active.

For non-null-named logs, the managed system records the security
credentials from the request that sets nlmConfigLogStatus
to ‘active’ and uses that identity to apply access control to
the objects in the Notification to decide if that Notification
may be logged."
 ::= { nlmConfigLogEntry 7 }

--

-- Statistics Section
--

nlmStatsGlobalNotificationsLogged OBJECT-TYPE
SYNTAX      Counter32
UNITS       "notifications"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of Notifications put into the nlmLogTable. This
counts a Notification once for each log entry, so a Notification
put into multiple logs is counted multiple times."
 ::= { nlmStats 1 }

nlmStatsGlobalNotificationsBumped OBJECT-TYPE
SYNTAX      Counter32
UNITS       "notifications"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of log entries discarded to make room for a new entry
due to lack of resources or the value of nlmConfigGlobalEntryLimit
or nlmConfigLogEntryLimit. This does not include entries discarded
due to the value of nlmConfigGlobalAgeOut."
 ::= { nlmStats 2 }

--

-- Log Statistics Table
--

nlmStatsLogTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NlmStatsLogEntry
MAX-ACCESS  not-accessible
STATUS  current
DESCRIPTION  "A table of Notification log statistics entries."
 ::= { nlmStats 3 }

nlmStatsLogEntry OBJECT-TYPE
SYNTAX  NlmStatsLogEntry
MAX-ACCESS  not-accessible
STATUS  current
DESCRIPTION  "A Notification log statistics entry."
AUGMENTS { nlmConfigLogEntry }
 ::= { nlmStatsLogTable 1 }

NlmStatsLogEntry ::= SEQUENCE {
   nlmStatsLogNotificationsLogged Counter32,
   nlmStatsLogNotificationsBumped Counter32
}

nlmStatsLogNotificationsLogged OBJECT-TYPE
SYNTAX  Counter32
UNITS  "notifications"
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION  "The number of Notifications put in this named log."
 ::= { nlmStatsLogEntry 1 }

nlmStatsLogNotificationsBumped OBJECT-TYPE
SYNTAX  Counter32
UNITS  "notifications"
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION  "The number of log entries discarded from this named log to make
room for a new entry due to lack of resources or the value of
nlmConfigGlobalEntryLimit or nlmConfigLogEntryLimit. This does not
include entries discarded due to the value of
nlmConfigGlobalAgeOut."
 ::= { nlmStatsLogEntry 2 }

--
-- Log Section
--

--
-- Log Table
nlmLogTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF NlmLogEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
      "A table of Notification log entries.

      It is an implementation-specific matter whether entries in this
      table are preserved across initializations of the management
      system. In general one would expect that they are not.

      Note that keeping entries across initializations of the
      management system leads to some confusion with counters and
      TimeStamps, since both of those are based on sysUpTime, which
      resets on management initialization. In this situation,
      counters apply only after the reset and nlmLogTime for entries
      made before the reset MUST be set to 0."
 ::= { nlmLog 1 }

nlmLogEntry OBJECT-TYPE
   SYNTAX      NlmLogEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
      "A Notification log entry.

      Entries appear in this table when Notifications occur and pass
      filtering by nlmConfigLogFilterName and access control. They are
      removed to make way for new entries due to lack of resources or
      the values of nlmConfigGlobalEntryLimit, nlmConfigGlobalAgeOut, or
      nlmConfigLogEntryLimit.

      If adding an entry would exceed nlmConfigGlobalEntryLimit or system
      resources in general, the oldest entry in any log SHOULD be removed
      to make room for the new one.

      If adding an entry would exceed nlmConfigLogEntryLimit the oldest
      entry in that log SHOULD be removed to make room for the new one.

      Before the managed system puts a locally-generated Notification
      into a non-null-named log it assures that the creator of the log
      has access to the information in the Notification. If not it
      does not log that Notification in that log."
   INDEX       { nlmLogName, nlmLogIndex }
 ::= { nlmLogTable 1 }
NlmLogEntry ::= SEQUENCE {
    nlmLogIndex     Unsigned32,
    nlmLogTime      TimeStamp,
    nlmLogDateAndTime DateAndTime,
    nlmLogEngineID  SnmpEngineID,
    nlmLogEngineTAddress TAddress,
    nlmLogEngineTDomain TDomain,
    nlmLogContextEngineID SnmpEngineID,
    nlmLogContextName SnmpAdminString,
    nlmLogNotificationID OBJECT IDENTIFIER
}

nlmLogIndex OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A monotonically increasing integer for the sole purpose of indexing entries within the named log. When it reaches the maximum value, an extremely unlikely event, the agent wraps the value back to 1."
 ::= { nlmLogEntry 1 }

nlmLogTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value of sysUpTime when the entry was placed in the log. If the entry occurred before the most recent management system initialization this object value MUST be set to zero."
 ::= { nlmLogEntry 2 }

nlmLogDateAndTime OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The local date and time when the entry was logged, instantiated only by systems that have date and time capability."
 ::= { nlmLogEntry 3 }

nlmLogEngineID OBJECT-TYPE
SYNTAX SnmpEngineID
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The identification of the SNMP engine at which the Notification
If the log can contain Notifications from only one engine or the Trap is in SNMPv1 format, this object is a zero-length string.

::= { nlmLogEntry 4 }

nlmLogEngineTAddress OBJECT-TYPE
SYNTAX TAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The transport service address of the SNMP engine from which the Notification was received, formatted according to the corresponding value of nlmLogEngineTDomain. This is used to identify the source of an SNMPv1 trap, since an nlmLogEngineId cannot be extracted from the SNMPv1 trap pdu.

This object MUST always be instantiated, even if the log can contain Notifications from only one engine.

Please be aware that the nlmLogEngineTAddress may not uniquely identify the SNMP engine from which the Notification was received. For example, if an SNMP engine uses DHCP or NAT to obtain ip addresses, the address it uses may be shared with other network devices, and hence will not uniquely identify the SNMP engine."

::= { nlmLogEntry 5 }

nlmLogEngineTDomain OBJECT-TYPE
SYNTAX TDomain
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Indicates the kind of transport service by which a Notification was received from an SNMP engine. nlmLogEngineTAddress contains the transport service address of the SNMP engine from which this Notification was received.

Possible values for this object are presently found in the Transport Mappings for SNMPv2 document (RFC 1906 [8])."

::= { nlmLogEntry 6 }

nlmLogContextEngineID OBJECT-TYPE
SYNTAX SnmpEngineID
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"If the Notification was received in a protocol which has a contextEngineID element like SNMPv3, this object has that value. Otherwise its value is a zero-length string."
::= { nlmLogEntry 7 }

nlmLogContextName OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The name of the SNMP MIB context from which the Notification came. For SNMPv1 Traps this is the community string from the Trap."
::= { nlmLogEntry 8 }

nlmLogNotificationID OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The NOTIFICATION-TYPE object identifier of the Notification that occurred."
::= { nlmLogEntry 9 }

--
-- Log Variable Table
--

nlmLogVariableTable OBJECT-TYPE
SYNTAX SEQUENCE OF NlmLogVariableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A table of variables to go with Notification log entries."
::= { nlmLog 2 }

nlmLogVariableEntry OBJECT-TYPE
SYNTAX NlmLogVariableEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A Notification log entry variable.
Entries appear in this table when there are variables in the varbind list of a Notification in nlmLogTable."
INDEX { nlmLogName, nlmLogIndex, nlmLogVariableIndex }
::= { nlmLogVariableTable 1 }

NlmLogVariableEntry ::= SEQUENCE {

Kavasseri Standards Track [Page 17]
nlmLogVariableIndex OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A monotonically increasing integer, starting at 1 for a given nlmLogIndex, for indexing variables within the logged Notification."
 ::= { nlmLogVariableEntry 1 }

nlmLogVariableID OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The variable’s object identifier."
 ::= { nlmLogVariableEntry 2 }

nlmLogVariableValueType 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 must be instantiated, based on this type."
 ::= { nlmLogVariableEntry 3 }

nlmLogVariableCounter32Val OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value when nlmLogVariableType is 'counter32'."
 ::= { nlmLogVariableEntry 4 }

nlmLogVariableUnsigned32Val OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value when nlmLogVariableType is 'unsigned32'."
 ::= { nlmLogVariableEntry 5 }

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

nlmLogVariableInteger32Val OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value when nlmLogVariableType is 'integer32'."
 ::= { nlmLogVariableEntry 7 }

nlmLogVariableOctetStringVal OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value when nlmLogVariableType is 'octetString'."
 ::= { nlmLogVariableEntry 8 }

nlmLogVariableIpAddressVal OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value when nlmLogVariableType is 'ipAddress'.
Although this seems to be unfriendly for IPv6, we have to recognize that there are a number of older MIBs that do contain an IPv4 format address, known as IpAddress.

IPv6 addresses are represented using TAddress or InetAddress, and so the underlying datatype is
OCTET STRING, and their value would be stored in the nlmLogVariableOctetStringVal column."
::= { nlmLogVariableEntry 9 }

nlmLogVariableOidVal OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value when nlmLogVariableType is 'objectId'".
::= { nlmLogVariableEntry 10 }

nlmLogVariableCounter64Val OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value when nlmLogVariableType is 'counter64'".
::= { nlmLogVariableEntry 11 }

nlmLogVariableOpaqueVal OBJECT-TYPE
SYNTAX Opaque
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value when nlmLogVariableType is 'opaque'".
::= { nlmLogVariableEntry 12 }

--
-- Conformance
--

notificationLogMIBConformance OBJECT IDENTIFIER ::= { notificationLogMIB 3 }
notificationLogMIBCompliances OBJECT IDENTIFIER ::= { notificationLogMIBConformance 1 }
notificationLogMIBGroups OBJECT IDENTIFIER ::= { notificationLogMIBConformance 2 }

-- Compliance

notificationLogMIBCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION "The compliance statement for entities which implement the Notification Log MIB."
MODULE -- this module
MANDATORY-GROUPS {
    notificationLogConfigGroup,
    notificationLogStatsGroup,
    notificationLogLogGroup
}

OBJECT nlmConfigGlobalEntryLimit
SYNTAX Unsigned32 (0..4294967295)
MIN-ACCESS read-only
DESCRIPTION
"Implementations may choose a limit and not allow it to be
changed or may enforce an upper or lower bound on the
limit."

OBJECT nlmConfigLogEntryLimit
SYNTAX Unsigned32 (0..4294967295)
MIN-ACCESS read-only
DESCRIPTION
"Implementations may choose a limit and not allow it to be
changed or may enforce an upper or lower bound on the
limit."

OBJECT nlmConfigLogEntryStatus
MIN-ACCESS read-only
DESCRIPTION
"Implementations may disallow the creation of named logs."

GROUP notificationLogDateGroup
DESCRIPTION
"This group is mandatory on systems that keep wall clock
date and time and should not be implemented on systems that
do not have a wall clock date."

::= { notificationLogMIBCompliances 1 }

-- Units of Conformance

notificationLogConfigGroup OBJECT-GROUP
OBJECTS {
    nlmConfigGlobalEntryLimit,
    nlmConfigGlobalAgeOut,
    nlmConfigLogFilterName,
    nlmConfigLogEntryLimit,
    nlmConfigLogAdminStatus,
    nlmConfigLogOperStatus,
    nlmConfigLogStorageType,
    nlmConfigLogEntryStatus
}
STATUS current
DESCRIPTION
"Notification log configuration management."
::= { notificationLogMIBGroups 1 }

notificationLogStatsGroup OBJECT-GROUP
OBJECTS {
  nlmStatsGlobalNotificationsLogged,
  nlmStatsGlobalNotificationsBumped,
  nlmStatsLogNotificationsLogged,
  nlmStatsLogNotificationsBumped
}
STATUS current
DESCRIPTION
"Notification log statistics."
::= { notificationLogMIBGroups 2 }

notificationLogLogGroup OBJECT-GROUP
OBJECTS {
  nlmLogTime,
  nlmLogEngineID,
  nlmLogEngineTAddress,
  nlmLogEngineTDomain,
  nlmLogContextEngineID,
  nlmLogContextName,
  nlmLogNotificationID,
  nlmLogVariableID,
  nlmLogVariableValueType,
  nlmLogVariableCounter32Val,
  nlmLogVariableUnsigned32Val,
  nlmLogVariableTimeTicksVal,
  nlmLogVariableInteger32Val,
  nlmLogVariableOctetStringVal,
  nlmLogVariableIpAddressVal,
  nlmLogVariableOidVal,
  nlmLogVariableCounter64Val,
  nlmLogVariableOpaqueVal
}
STATUS current
DESCRIPTION
"Notification log data."
::= { notificationLogMIBGroups 3 }

notificationLogDateGroup OBJECT-GROUP
OBJECTS {
  nlmLogDateAndTime
}
STATUS current
DESCRIPTION

"Conditionally mandatory notification log data. This group is mandatory on systems that keep wall clock date and time and should not be implemented on systems that do not have a wall clock date."

 ::= { notificationLogMIBGroups 4 }

END

4. Intellectual Property

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

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

5. References


6. Security Considerations

Security issues are discussed in Section 3.1.2.

7. Authors' Addresses

   Bob Stewart
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, CA 95134-1706
   U.S.A.

   Ramanathan Kavasseri
   Cisco Systems, Inc.
   170 West Tasman Drive
   San Jose, CA 95134-1706
   U.S.A.

   Phone: +1 408 527 2446
   EMail: ramk@cisco.com
8. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

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

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

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

Acknowledgement

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