Internet Draft

Andy Bierman
Cisco Systems, Inc.
Keith McCloghrie
Cisco Systems, Inc.
12 November 2001

Remote Monitoring MIB Extensions for High Capacity Alarms

<draft-bierman-rmonmib-hc-alarm-mib-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [RFC2026].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Distribution of this document is unlimited. Please send comments to authors.

1. Copyright Notice

2. 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 for extending the alarm thresholding capabilities found in the RMON MIB [RFC2819], to provide similar threshold monitoring of objects based on the Counter64 data type.

3. Table of Contents

4. The SNMP Network 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 RFC 1155 [RFC1155], RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in RFC 2578

[RFC2578], RFC 2579 [RFC2579] and RFC 2580 [RFC2580].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in 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 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.

5. Overview

There is a need for a standardized way of providing the same type of alarm thresholding capabilities for Counter64 objects, as already exists for Counter32 objects. The RMON-1 alarmTable objects and RMON-1 notification types are specific to 32-bit objects, and cannot be used to properly monitor Counter64-based objects. Extensions to these existing constructs are needed which explicitly support Counter64-based objects. These extensions are completely independent of the existing RMON-1 alarm

mechanisms.

The usage of Counter64 objects is increasing. One of the causes for this increase is the increasing speeds of network interfaces; RFC 2863 [RFC2863] says:

> As the speed of network media increase, the minimum time in which a 32 bit counter will wrap decreases. For example, a 10Mbs stream of back-to-back, full-size packets causes ifInOctets to wrap in just over 57 minutes; at 100Mbs, the minimum wrap time is 5.7 minutes, and at 1Gbs, the minimum is 34 seconds. Requiring that interfaces be polled frequently enough not to miss a counter wrap is increasingly problematic.

and therefore requires:

> For interfaces that operate at 20,000,000 (20 million) bits per second or less, 32-bit byte and packet counters MUST be supported. For interfaces that operate faster than 20,000,000 bits/second, and slower than 650,000,000 bits/second, 32-bit packet counters MUST be supported and 64-bit octet counters MUST be supported. For interfaces that operate at 650,000,000 bits/second or faster, 64-bit packet counters AND 64-bit octet counters MUST be supported.

Of the variables on which thresholds are set using RMON-1's alarmTable, two of the most popular are: ifInOctets and ifOutOctets. Thus, the increasing usage of the 64-bit versions: ifHCInOctets and ifHCOutOctets means that there's an increasing requiurement to use RMON-1's thresholding capability for ifHCInOctets and ifHCOutOctets.

The RMON-1 Alarm Group is implemented not only by all RMON probes, but also by the SNMP agents in many other types of devices for the purpose of monitoring any of their (non-RMON) integer-valued MIB objects. The fact that it has been so widely implemented indicates its obvious value. Without this extension, that obvious value is becoming incomplete because of its lack of support for 64-bit integers. This extension is the easiest, simplest, and most compatible way for an implementation to overcome that lack of support.

5.1. Terms

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

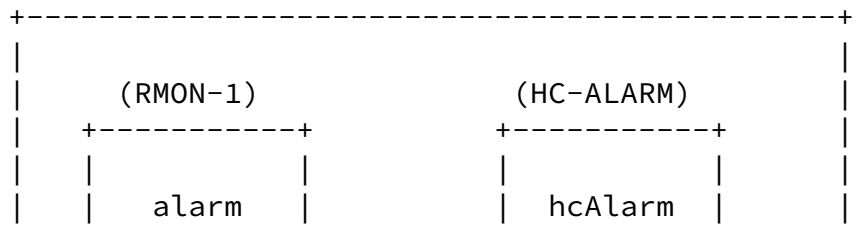
5.2. Relationship to the Remote Monitoring MIBs

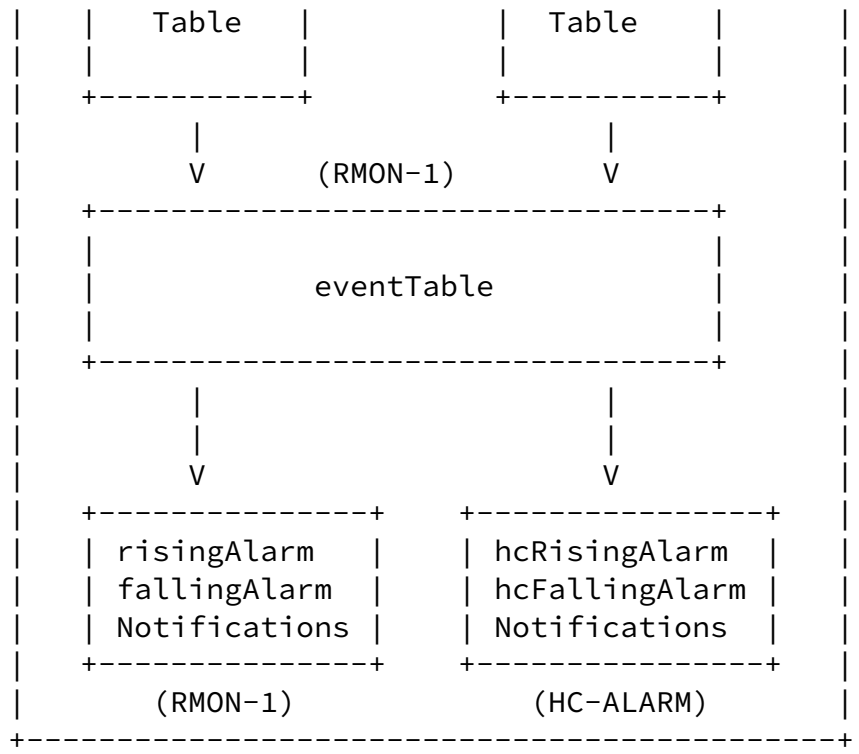
This MIB is intended to be implemented in Remote Monitoring (RMON) probes, which may also support the RMON-1 MIB [RFC2819]. Such probes may be stand-alone devices, or may be co-located with other networking devices (e.g., ethernet switches and repeaters).

The functionality of the High Capacity Alarm Group is a supserset of RMON-1's Alarm Group. Thus, one day in the distant future, it's a possibility that RMON-1's Alarm Group will be deprecated in favour of this MIB's High Capacity Alarm Group. However, that day will not come before this document, or one of its successors, reaches the same standardization state as RMON-1.

6. MIB Structure

Figure 1: HC-ALARM MIB Functional Structure





6.1. MIB Group Overview

The HC-ALARM MIB contains two MIB groups:

- hcAlarmObjects group
  Controls the configuration of alarms for high capacity MIB object instances.

- hcAlarmNotifications group
  Provide new rising and falling threshold notifications for high capacity objects.

6.1.1. High Capacity Alarm Group

This group contains one table, which is used by a management station to configure high capacity alarm entries. To configure alarm thresholding for Counter64 or CounterBasedGauge64 objects, a management application must configure the hcAlarmTable in a manner similar to how RMON-1's

alarmTable is configured.

Because the language in the some of the DESCRIPTION clauses of objects in the alarmTable is specific to the alarmTable itself, their defined semantics do not allow them to be used for this MIB also. Therefore, the following objects are essentially cloned from the alarmTable to the hcAlarmTable:

alarmTable	hcAlarmTable
alarmIndex	hcAlarmIndex
alarmInterval	hcAlarmInterval
alarmVariable	hcAlarmVariable
alarmSampleType	hcAlarmSampleType
alarmStartupAlarm	hcAlarmStartupAlarm
alarmRisingEventIndex	hcAlarmRisingEventIndex
alarmFallingEventIndex	hcAlarmFallingEventIndex
alarmOwner	hcAlarmOwner
alarmStatus	hcAlarmStatus

In addition, the following hcAlarmTable objects are used as high capacity values instead of the corresponding 32-bit version in the alarmTable.

alarmTable	hcAlarmTable
alarmValue	hcAlarmAbsValue

	hcAlarmValueStatus
alarmRisingThreshold	hcAlarmRisingThresholdAbsValue
	hcAlarmRisingThresholdValStatus
alarmFallingThreshold	hcAlarmFallingThresholdAbsValue
	hcAlarmFallingThresholdValStatus

Nevertheless, the hcAlarmTable does have a few differences from the alarmTable:

- Counter64 based objects are thresholded properly
- an entry is not destroyed if the instance identified by the hcAlarmVariable is not available during a polling interval.

- the RowStatus textual convention is used instead of EntryStatus for the hcAlarmStatus object."

6.1.2. High Capacity Alarm Notifications

This group contains two notifications, hcRisingAlarm and hcFallingAlarm. These are generated for high capacity alarms in the same manner and used to convey essentially the same information as RMON-1's risingAlarm and fallingAlarm notifications do for alarmTable-specified alarms.

7. Definitions

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

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
```

```
        Integer32, Counter64, experimental
                FROM SNMPv2-SMI
        MODULE-COMPLIANCE, OBJECT-GROUP,
        NOTIFICATION-GROUP
                FROM SNMPv2-CONF
        RowStatus, VariablePointer, TEXTUAL-CONVENTION
                FROM SNMPv2-TC
        OwnerString, rmonEventGroup
                FROM RMON-MIB;

hcAlarmMIB MODULE-IDENTITY
    LAST-UPDATED    "200111060000Z"
    ORGANIZATION    "Cisco Systems, Inc."
    CONTACT-INFO
            "        Andy Bierman
                     Cisco Systems, Inc.
                Tel: +1 408 527-3711
             E-mail: abierman@cisco.com
             Postal: 170 West Tasman Drive
                     San Jose, CA USA 95134

                     Keith McCloghrie
                     Cisco Systems, Inc.
                Tel: +1 408 526-5260
             E-mail: kzm@cisco.com
             Postal: 170 West Tasman Drive
                     San Jose, CA USA 95134
            "
    DESCRIPTION
            "This module defines Remote Monitoring MIB extensions for
            High Capacity Alarms."
    REVISION        "200111060000Z"
    DESCRIPTION
            "Initial version of the High Capacity Alarm MIB module."
    ::= { experimental 999999 } -- unassigned


hcAlarmObjects       OBJECT IDENTIFIER ::= { hcAlarmMIB 1 }
hcAlarmNotifications OBJECT IDENTIFIER ::= { hcAlarmMIB 2 }
```

```
hcAlarmConformance   OBJECT IDENTIFIER ::= { hcAlarmMIB 3 }

hcAlarmControlObjects OBJECT IDENTIFIER ::= { hcAlarmObjects 1 }


--
-- Textual Conventions
--

HcValueStatus ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "This data type indicates the validity and sign of the data
            in associated object instances of the type HcAbsoluteValue.
            Objects of this type MUST be defined together with an object
            of type HcAbsoluteValue.

            If the associated object instance of type HcAbsoluteValue
            could not be accessed during the sampling interval, and is
            therefore invalid, then the associated HcValueStatus object
            will contain the value 'valueNotAvailable(1)'.

            If the associated object instance of type HcAbsoluteValue is
            valid and actual value of the sample is greater than or
            equal to zero, then the associated HcValueStatus object will
            contain the value 'valuePositive(2)'.

            If the associated object instance of type HcAbsoluteValue is
            valid and the actual value of the sample is less than zero,
            then the associated HcValueStatus object will contain the
            value 'valueNegative(3)'.  The associated absolute value
            should be multiplied by -1 to obtain the true sample value."
    SYNTAX INTEGER {
        valueNotAvailable(1),
        valuePositive(2),
        valueNegative(3)
    }

HcAbsoluteValue ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "Objects of this data type represent a 64-bit absolute
            value.  Objects of this type MUST be defined together with
            an object of type HcValueStatus. The 'combined object'
            represents a 64-bit signed integer, or an unavailable
```

```
            number.

            This data type represents a non-negative integer, which may
            increase or decrease, but shall never exceed a maximum
            value, nor fall below a minimum value. The maximum value can
            not be greater than 2^64-1 (18446744073709551615 decimal),
            and the minimum value can not be smaller than 0.  The value
            of an HcAbsoluteValue has its maximum value whenever the
            information being modeled is greater than or equal to its
            maximum value, and has its minimum value whenever the
            information being modeled is smaller than or equal to its
            minimum value.  If the information being modeled
            subsequently decreases below (increases above) the maximum
            (minimum) value, the HcAbsoluteValue also decreases
            (increases).

            Note that this TC is not strictly supported in SMIv2,
            because the 'always increasing' and 'counter wrap' semantics
            associated with the Counter64 base type are not preserved.
            It is possible that management applications which rely
            solely upon the (Counter64) ASN.1 tag to determine object
            semantics will mistakenly operate upon objects of this type
            as they would for Counter64 objects.

            This textual convention represents a limited and short-term
            solution, and may be deprecated as a long term solution is
            defined and deployed to replace it."
    SYNTAX Counter64        -- CounterBasedGauge64

--
-- High Capacity Alarm Table
--

hcAlarmTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF HcAlarmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "A list of entries for the configuration of high capacity
            alarms."
    ::= { hcAlarmControlObjects 1 }

hcAlarmEntry OBJECT-TYPE
    SYNTAX      HcAlarmEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
            "A conceptual row in the hcAlarmTable. Entries are created
            in this table by management application action only.

            The agent SHOULD support non-volatile configuration of this
            table, and upon system initialization, the table SHOULD be
            initialized with the saved values."
    INDEX { hcAlarmIndex }
    ::= { hcAlarmTable 1 }


HcAlarmEntry ::= SEQUENCE {
    hcAlarmIndex                    Integer32,
    hcAlarmInterval                 Integer32,
    hcAlarmVariable                 VariablePointer,
    hcAlarmSampleType               INTEGER,
    hcAlarmAbsValue                 HcAbsoluteValue,
    hcAlarmValueStatus              HcValueStatus,
    hcAlarmStartupAlarm             INTEGER,
    hcAlarmRisingThresholdAbsValue  HcAbsoluteValue,
    hcAlarmRisingThresholdValStatus HcValueStatus,
    hcAlarmFallingThresholdAbsValue HcAbsoluteValue,
    hcAlarmFallingThresholdValStatus HcValueStatus,
    hcAlarmRisingEventIndex         Integer32,
    hcAlarmFallingEventIndex        Integer32,
    hcAlarmOwner                    OwnerString,
    hcAlarmStatus                   RowStatus }

hcAlarmIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An arbitrary integer index value used to uniquely identify
            this high capacity alarm entry."
    ::= { hcAlarmEntry 1 }

hcAlarmInterval OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
```

```
    UNITS      "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The interval in seconds over which the data is sampled and
```



```
            compared with the rising and falling thresholds.  When
            setting this variable, care should be taken in the case of
            deltaValue sampling - the interval should be set short
            enough that the sampled variable is very unlikely to
            increase or decrease by more than 2^63 - 1 during a single
            sampling interval.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 2 }

hcAlarmVariable OBJECT-TYPE
    SYNTAX      VariablePointer
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The object identifier of the particular variable to be
            sampled.  Only variables that resolve to an ASN.1 primitive
            type of INTEGER (INTEGER, Integer32, Counter32, Counter64,
            Gauge, or TimeTicks) may be sampled.

            Because SNMP access control is articulated entirely in terms
            of the contents of MIB views, no access control mechanism
            exists that can restrict the value of this object to
            identify only those objects that exist in a particular MIB
            view.  Because there is thus no acceptable means of
            restricting the read access that could be obtained through
            the alarm mechanism, the probe must only grant write access
            to this object in those views that have read access to all
            objects on the probe.

            During a set operation, if the supplied variable name is not
            available in the selected MIB view, a badValue error MUST be
            returned.
```

```
            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 3 }

hcAlarmSampleType OBJECT-TYPE
    SYNTAX INTEGER {
               absoluteValue(1),
               deltaValue(2)
           }
    MAX-ACCESS  read-create
```



```
    STATUS      current
    DESCRIPTION
            "The method of sampling the selected variable and
            calculating the value to be compared against the thresholds.
            If the value of this object is absoluteValue(1), the value
            of the selected variable will be compared directly with the
            thresholds at the end of the sampling interval.  If the
            value of this object is deltaValue(2), the value of the
            selected variable at the last sample will be subtracted from
            the current value, and the difference compared with the
            thresholds.

            If the associated hcAlarmVariable instance could not be
            obtained at the previous sample interval, then a delta
            sample is not possible, and the value of the associated
            hcAlarmValueStatus object for this interval will be
            valueNotAvailable(1).

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 4 }

hcAlarmAbsValue OBJECT-TYPE
    SYNTAX      HcAbsoluteValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The absolute value (i.e. unsigned value) of the
```

```
            hcAlarmVariable statistic during the last sampling period.
            The value during the current sampling period is not made
            available until the period is completed.

            To obtain the true value for this sampling interval, the
            associated instance of hcAlarmValueStatus must be checked,
            and the value of this object adjusted as necessary.

            If the MIB instance could not be accessed during the
            sampling interval, then this object will have a value of
            zero and the associated instance of hcAlarmValueStatus will
            be set to 'valueNotAvailable(1)'."
    ::= { hcAlarmEntry 5 }

hcAlarmValueStatus OBJECT-TYPE
    SYNTAX      HcValueStatus
    MAX-ACCESS  read-only
```



```
    STATUS      current
    DESCRIPTION
            "This object indicates the validity and sign of the data for
            the hcAlarmAbsValue object, as described in the
            HcValueStatus textual convention."
    ::= { hcAlarmEntry 6 }

hcAlarmStartupAlarm OBJECT-TYPE
    SYNTAX     INTEGER {
                  risingAlarm(1),
                  fallingAlarm(2),
                  risingOrFallingAlarm(3)
               }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The alarm that may be sent when this entry is first set to
            valid.  If the first sample after this entry becomes valid
            is greater than or equal to the rising threshold and this
            object is equal to risingAlarm(1) or
            risingOrFallingAlarm(3), then a single rising alarm will be
            generated.  If the first sample after this entry becomes
```

```
            valid is less than or equal to the falling threshold and
            this object is equal to fallingAlarm(2) or
            risingOrFallingAlarm(3), then a single falling alarm will be
            generated.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 7 }

hcAlarmRisingThresholdAbsValue OBJECT-TYPE
    SYNTAX      HcAbsoluteValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The absolute value for threshold for the sampled statistic.
            The actual threshold value is determined by the associated
            instance of the hcAlarmRisingThresholdValStatus object, as
            described in the HcAbsoluteValue textual convention.

            When the current sampled value is greater than or equal to
            this threshold, and the value at the last sampling interval
            was less than this threshold, a single event will be
            generated.  A single event will also be generated if the
```



```
            first sample after this entry becomes valid is greater than
            or equal to this threshold and the associated
            hcAlarmStartupAlarm is equal to risingAlarm(1) or
            risingOrFallingAlarm(3).

            After a rising event is generated, another such event will
            not be generated until the sampled value falls below this
            threshold and reaches the threshold identified by the
            hcAlarmFallingThresholdAbsValue and
            hcAlarmFallingThresholdValStatus objects.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 8 }

hcAlarmRisingThresholdValStatus OBJECT-TYPE
```

```
    SYNTAX      HcValueStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object indicates the sign of the data for the
            hcAlarmRisingThresholdAbsValue object, as described in the
            HcValueStatus textual convention.

            The enumeration 'valueNotAvailable(1)' is not allowed, and
            the associated hcAlarmStatus object cannot be equal to
            'active(1)' if this object is set to this value.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 9 }

hcAlarmFallingThresholdAbsValue OBJECT-TYPE
    SYNTAX      HcAbsoluteValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The absolute value for threshold for the sampled statistic.
            The actual threshold value is determined by the associated
            instance of the hcAlarmFallingThresholdValStatus object, as
            described in the HcAbsoluteValue textual convention.

            When the current sampled value is less than or equal to this
            threshold, and the value at the last sampling interval was
            greater than this threshold, a single event will be
```



```
            generated.  A single event will also be generated if the
            first sample after this entry becomes valid is less than or
            equal to this threshold and the associated
            hcAlarmStartupAlarm is equal to fallingAlarm(2) or
            risingOrFallingAlarm(3).

            After a falling event is generated, another such event will
            not be generated until the sampled value rises above this
            threshold and reaches the threshold identified by the
            hcAlarmRisingThresholdAbsValue and
```

```
            hcAlarmRisingThresholdValStatus objects.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 10 }

hcAlarmFallingThresholdValStatus OBJECT-TYPE
    SYNTAX      HcValueStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "This object indicates the sign of the data for the
            hcAlarmFallingThresholdAbsValue object, as described in the
            HcValueStatus textual convention.

            The enumeration 'valueNotAvailable(1)' is not allowed, and
            the associated hcAlarmStatus object cannot be equal to
            'active(1)' if this object is set to this value.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 11 }

hcAlarmRisingEventIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The index of the eventEntry that is used when a rising
            threshold is crossed.  The eventEntry identified by a
            particular value of this index is the same as identified by
            the same value of the eventIndex object.  If there is no
            corresponding entry in the eventTable, then no association
            exists.  In particular, if this value is zero, no associated
            event will be generated, as zero is not a valid event index.
```



```
            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 12 }
```

```
hcAlarmFallingEventIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The index of the eventEntry that is used when a falling
            threshold is crossed.  The eventEntry identified by a
            particular value of this index is the same as identified by
            the same value of the eventIndex object.  If there is no
            corresponding entry in the eventTable, then no association
            exists.  In particular, if this value is zero, no associated
            event will be generated, as zero is not a valid event index.

            This object may not be modified if the associated
            hcAlarmStatus object is equal to active(1)."
    ::= { hcAlarmEntry 13 }

hcAlarmOwner OBJECT-TYPE
    SYNTAX     OwnerString
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
            "The entity that configured this entry and is therefore
            using the resources assigned to it."
    ::= { hcAlarmEntry 14 }

hcAlarmStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The status of this row.

            An entry MUST NOT exist in the active state unless all
            objects in the entry have an appropriate value, as described
            in the description clause for each writable object."
    ::= { hcAlarmEntry 15 }

--
-- Notifications
--
```

```
hcAlarmNotifPrefix OBJECT IDENTIFIER
      ::= { hcAlarmNotifications 0 }

hcRisingAlarm NOTIFICATION-TYPE
    OBJECTS  { hcAlarmVariable,
               hcAlarmSampleType,
               hcAlarmAbsValue,
               hcAlarmValueStatus,
               hcAlarmRisingThresholdAbsValue,
               hcAlarmRisingThresholdValStatus,
               hcAlarmRisingEventIndex }
    STATUS   current
    DESCRIPTION
            "The SNMP notification that is generated when a high
            capacity alarm entry crosses its rising threshold and
            generates an event that is configured for sending SNMP
            traps.

            The hcAlarmEntry object instances identified in the OBJECTS
            clause are from the entry that causes this notification to
            be generated."
     ::= { hcAlarmNotifPrefix 1 }


hcFallingAlarm NOTIFICATION-TYPE
    OBJECTS { hcAlarmVariable,
              hcAlarmSampleType,
              hcAlarmAbsValue,
              hcAlarmValueStatus,
              hcAlarmFallingThresholdAbsValue,
              hcAlarmFallingThresholdValStatus,
              hcAlarmFallingEventIndex }
    STATUS    current
    DESCRIPTION
            "The SNMP notification that is generated when a high
            capacity alarm entry crosses its falling threshold and
            generates an event that is configured for sending SNMP
            traps.

            The hcAlarmEntry object instances identified in the OBJECTS
            clause are from the entry that causes this notification to
            be generated."
     ::= { hcAlarmNotifPrefix 2 }

--
```

```
-- Conformance Section
--

hcAlarmCompliances OBJECT IDENTIFIER ::= { hcAlarmConformance 1 }
hcAlarmGroups      OBJECT IDENTIFIER ::= { hcAlarmConformance 2 }

hcAlarmCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "Describes the requirements for conformance to the High
            Capacity Alarm MIB."
    MODULE  -- this module
        MANDATORY-GROUPS {
                           hcAlarmControlGroup,
                           hcAlarmNotificationsGroup
                         }

    MODULE RMON-MIB
        MANDATORY-GROUPS { rmonEventGroup }

    ::= { hcAlarmCompliances 1 }


-- Object Groups

hcAlarmControlGroup OBJECT-GROUP
    OBJECTS {
              hcAlarmInterval,
              hcAlarmVariable,
              hcAlarmSampleType,
              hcAlarmAbsValue,
              hcAlarmValueStatus,
              hcAlarmStartupAlarm,
              hcAlarmRisingThresholdAbsValue,
              hcAlarmRisingThresholdValStatus,
              hcAlarmFallingThresholdAbsValue,
              hcAlarmFallingThresholdValStatus,
              hcAlarmRisingEventIndex,
              hcAlarmFallingEventIndex,
              hcAlarmOwner,
              hcAlarmStatus
    }
    STATUS  current
```

```
    DESCRIPTION
            "A collection of objects used to configure entries for high
```



```
            capacity alarm threshold monitoring purposes."
    ::= { hcAlarmGroups 1 }

hcAlarmNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
            hcRisingAlarm,
            hcFallingAlarm
    }
    STATUS  current
    DESCRIPTION
            "A collection of notifications to deliver information
            related to a high capacity rising or falling threshold event
            to a management application."
    ::= { hcAlarmGroups 2 }

END
```

8. 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.

9. Acknowledgements

This memo is based on existing alarmTable objects in the RMON-1 MIB module [RFC2819]. In order to maintain the RMON 'look-and-feel' and semantic consistency, some of Steve Waldbusser's text from [RFC2819] has been adapted for use in this MIB.

10. References

[RFC1155]
Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.

[RFC1157]
Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[RFC1212]
Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.

[RFC1215]
M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991.

[RFC1901]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[RFC1905]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[RFC1906]
SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

[RFC2021]

S. Waldbusser, "Remote Network Monitoring MIB (RMON-2)", RFC 2021, International Network Services, January 1997.

[RFC2026]
Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, Harvard University, October, 1996.

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

[RFC2570]
Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, SNMP Research, Inc., TIS Labs at Network Associates, Inc., Ericsson, Cisco Systems, April 1999.

[RFC2571]
Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, Cabletron

Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April 1999.

[RFC2572]
Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, SNMP Research, Inc., Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April 1999.

[RFC2573]
Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, SNMP Research, Inc., Secure Computing Corporation, Cisco Systems, April 1999.

[RFC2574]
Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, IBM T. J. Watson Research, April 1999.

[RFC2575]
Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, IBM T. J. Watson Research, BMC Software, Inc., Cisco Systems, Inc., April 1999.

[RFC2578]
McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", RFC 2578, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999.

[RFC2579]
McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", RFC 2579, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999.

[RFC2580]
McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", RFC 2580, STD 58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First Virtual Holdings, International Network Services, April 1999.

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

[RFC2856]
Bierman, A., McCloghrie, K., and R. Presuhn, "Textual Conventions for Additional High Capacity Data Types", RFC 2856, Cisco Systems, Inc., BMC Software, Inc., June, 2000.

[RFC2863]
McCloghrie, K., and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, Cisco Systems, Argon Networks, June, 2000.

11. Security Considerations

There are a number of management objects defined in this MIB that have 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.

There are a number of managed objects in this MIB that may contain sensitive information. These are:

```
hcAlarmAbsValue
hcAlarmValueStatus
```

This object may expose the values of particular MIB instances, as identified by associated instances of the hcAlarmVariable object.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [RFC2574] and the View-based Access Control Model RFC 2575 [RFC2575] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

12. Authors' Addresses

Andy Bierman
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA USA 95134
Phone: +1 408-527-3711

Email: abierman@cisco.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA USA 95134
Phone: +1 408-526-5260
Email: kzm@cisco.com

13. Full Copyright Statement