INTERNET DRAFT
Expires October 1, 2002

R. Abbi
Alcatel

B. Ray
March 2002

High Capacity Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals
draft-ietf-adslmib-hc-tc-00.txt

Status of This Memo

Copyright Notice

Abstract

This document presents a set of Textual Conventions for MIB modules which extends the conventions presented in RFC2493 to 64 bit resolution using the conventions presented in RFC2856.

Table of Contents

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. The textual conventions defined in this MIB module cannot be translated to SMIv1 since the Counter64 type does not exist in SMIv1.

2. Overview

In cases where a manager must obtain performance history data about the behavior of equipment it manages several strategies can be followed in the design of a MIB that represents the managed equipment, including:

0 The agent counts events on a continuous basis and, whenever desired, the manager obtains the value of the event counter and adjusts its understanding of the history of events at the agent.

0 The agent allocates events to 'buckets' where each bucket represents an interval of time.

Telecommunications equipment often makes use of the latter strategy. For such equipment the standard practice is that history data is maintained by the agent in terms of 15-minute intervals [T1.231].

MIB modules for collecting performance history based on 15-minute intervals have been defined for the DS1/E1 [RFC2495], DS3/E3 [RFC2496], SONET/SDH [RFC2558], and ADSL [RFC2622] interface types. These MIB modules use a common set of textual conventions defined in [RFC2493]. Those textual conventions are based on the Gauge32 data type.

A need has arisen in connection with recent work on a VDSL MIB [VDSL-MIB] to define 64-bit versions of the textual conventions in [RFC2493]. Ideally, these high-capacity textual conventions would be based on a Gauge64 or Unsigned64 data type, but unfortunately no such types exist in SMIv2. The next best choice would be to base them on the CounterBasedGauge64 textual convention presented in [RFC2856], but that is not possible either since SMIv2 allows only base types to be used textual conventions. Therefore the textual conventions presented in this memo are based directly on the Counter64 type, like those in [RFC2856]. They are subject to the following limitations:

- The MAX-ACCESS of objects defined using these textual conventions must be read-only, because the MAX-ACCESS of the underlying Counter64 type is read-only.

- No sub-range can be specified in object definitions using these textual conventions, because sub-ranges are not allowed on Counter64 objects.

- No DEFVAL clause can be specified in object definitions using these textual conventions, because DEFVALs are not allowed on Counter64 objects.

- Objects defined using these textual conventions cannot be used in an INDEX clause, because there is no INDEX clause mapping defined for objects of type Counter64.

3. Definitions

```
HC-PerfHist-TC-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    Counter64,
    mib-2                FROM SNMPv2-SMI
    TEXTUAL-CONVENTION   FROM SNMPv2-TC;

hcPerfHistTCMIB MODULE-IDENTITY
   LAST-UPDATED "200203310000Z" -- March 31, 2002
   ORGANIZATION "ADSLMIB Working Group"
   CONTACT-INFO "WG-email:  adslmib@ietf.org
                Info:      https://www1.ietf.org/mailman/listinfo/adslmib

                Chair:     Mike Sneed
                           Inovia Telecoms
                Postal:    1017 Main Campus Drive
                           Raleigh NC 27606 USA
                Email:     Mike.Sneed@go.ecitele.com
                Phone:     +1 919 513 1435

                Co-editor: Rajesh Abbi
                           Alcatel USA
                Postal:    2912 Wake Forest Road
                           Raleigh, NC 27609-7860 USA
                Email:     Rajesh.Abbi@usa.alcatel.com
                Phone:     +1 919 850 6194

                Co-editor: Bob Ray
                Email:     bob_ray_99@yahoo.com
          "
    DESCRIPTION
        "This MIB Module provides Textual Conventions to be
         used by systems supporting 15 minute based performance
         history counts that require high-capacity counts."
    ::= { mib-2 xxx }  -- to be assigned by IANA

-- The Textual Conventions defined below are organized
-- alphabetically

-- Use of these TCs assumes the following:
-- 0  The agent supports 15 minute based history
--    counters.
-- 0  The agent is capable of keeping a history of n
```

```
    --    intervals of 15 minute performance data. The
    --    value of n is defined by the specific MIB
    --    module but shall be 0 < n =< 96.
```

```
    -- 0  The agent may optionally support performance
    --    data aggregating the history intervals.
    -- 0  The agent will keep separate tables for the
    --    current interval, the history intervals, and
    --    the total aggregates.
    -- 0  The agent will keep the following objects.
    --    If performance data is kept for multiple instances
    --    of a measured entity, then
    --    these objects are applied to each instance of
    --    the measured entity (e.g., interfaces).

    -- xyzTimeElapsed OBJECT-TYPE
    --       SYNTAX  INTEGER (0..899)
    --       MAX-ACCESS  read-only
    --       STATUS  current
    --       DESCRIPTION
    --       "The number of seconds that have elapsed since
    --       the beginning of the current measurement period.
    --       If, for some reason, such as an adjustment in the
    --       system's time-of-day clock, the current interval
    --       exceeds the maximum value, the agent will return
    --       the maximum value."
    --       ::= { xxx }

    -- xyzValidIntervals OBJECT-TYPE
    --       SYNTAX  INTEGER (0..<n>)
    --       MAX-ACCESS  read-only
    --       STATUS  current
    --       DESCRIPTION
    --       "The number of previous near end intervals
    --       for which data was collected.
    --          [ The overall constraint on <n> is 1 =< n =< 96; ]
    --          [ Define any additional constraints on <n> here. ]
    --       The value will be <n> unless the measurement was
    --       (re-)started within the last (<n>*15) minutes, in which
    --       case the value will be the number of complete 15
    --       minute intervals for which the agent has at least
    --       some data. In certain cases (e.g., in the case
    --       where the agent is a proxy) it is possible that some
    --       intervals are unavailable.  In this case, this
    --       interval is the maximum interval number for
    --       which data is available."
    --       ::= { xxx }

    -- xyzInvalidIntervals OBJECT-TYPE
    --     SYNTAX  INTEGER (0..<n>)
    --     MAX-ACCESS  read-only
    --     STATUS  current
    --     DESCRIPTION
```

```
    --       "The number of intervals in the range from
    --        0 to xyzValidIntervals for which no
    --        data is available. This object will typically
    --        be zero except in cases where the data for some
```

```
    --        intervals are not available (e.g., in proxy
    --        situations)."
    --       ::= { xxx }

    -- See the notes in [RFC2493] for additional information
    -- concerning the above objects.

    HCPerfCurrentCount ::= TEXTUAL-CONVENTION
        STATUS  current
        DESCRIPTION
            "A counter associated with a performance measurement in a
             current 15 minute measurement interval. The value of this
             counter starts from zero and is increased when associated
             events occur, until the end of the 15 minute interval.
             At that time the value of the counter is stored in the
             first 15 minute history interval, and the CurrentCount is
             restarted at zero. In the case where the agent has no valid
             data available for the current interval the corresponding
             object instance is not available and upon a retrieval
             request a corresponding error message shall be returned to
             indicate that this instance does not exist.

             This count represents a a non-negative integer, which
             may increase or decrease, but shall never exceed 2^64-1
             (18446744073709551615 decimal), nor fall below 0.  The
             The value of a HCPerfCurrentCount object assumes its
             maximum value whenever the underlying count exceeds
             2^64-1.  If the underlying count subsequently decreases
             below 2^64-1 (due, e.g., to a retroactive adjustment as a
             result of entering or exiting unavailable time), then the
             HCPerfCurrentCount object also decreases.

             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

    HCPerfIntervalCount ::= TEXTUAL-CONVENTION
        STATUS  current
        DESCRIPTION
            "A counter associated with a performance measurement in
```

a previous 15 minute measurement interval. In the case where the agent has no valid data available for a particular interval the corresponding object instance is not available and upon a retrieval request a corresponding

```
            error message shall be returned to indicate that this
            instance does not exist.

            In a system supporting a history of n intervals with
            IntervalCount(1) and IntervalCount(n) the most and least
            recent intervals respectively, the following applies at
            the end of a 15 minute interval:

               - discard the value of IntervalCount(n)
               - the value of IntervalCount(i) becomes that
                   of IntervalCount(i-1) for n >= i > 1
               - the value of IntervalCount(1) becomes that
                   of CurrentCount
               - the TotalCount, if supported, is adjusted.

            This count represents a a non-negative integer, which
            may increase or decrease, but shall never exceed 2^64-1
            (18446744073709551615 decimal), nor fall below 0.  The
            The value of a HCPerfIntervalCount object assumes its
            maximum value whenever the underlying count exceeds
            2^64-1.  If the underlying count subsequently decreases
            below 2^64-1 (due, e.g., to a retroactive adjustment as a
            result of entering or exiting unavailable time), then the
            HCPerfIntervalCount object also decreases.

            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

    HCPerfTotalCount ::= TEXTUAL-CONVENTION
        STATUS  current
        DESCRIPTION
            "A counter associated with a performance measurements
             aggregating the previous valid 15 minute measurement
             intervals. Intervals for which no valid data was
             available are not counted.

             This count represents a a non-negative integer, which
             may increase or decrease, but shall never exceed 2^64-1
             (18446744073709551615 decimal), nor fall below 0.  The
```

The value of a HCPerfTotalCount object assumes its maximum value whenever the underlying count exceeds 2^64-1. If the underlying count subsequently decreases below 2^64-1 (due, e.g., to a retroactive adjustment as a

```
            result of entering or exiting unavailable time), then the
            HCPerfTotalCount object also decreases.

            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
    END
```

4. Acknowledgements

This document borrows tremendously from [RFC2493] and [RFC2856]. As such, any credit for the text found within should be fully attributed to the authors of those documents.

5. References

[T1.231] American National Standard for Telecommunications - Digital Hierarchy - Layer 1 In-Service Digital Transmission Performance Monitoring, ANSI T1.231-1997, September 1997.

[RFC1155] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

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

[RFC1212] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

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

[RFC1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC1905] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network

Management Protocol (SNMPv2)", RFC 1905, January 1996.

[RFC1906] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2493] Tesink, K., "Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals", RFC 2493, January 1999.

[RFC2495] Fowler, D., "Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types", RFC 2495, January 1999.

[RFC2496] Fowler, D., "Definitions of Managed Objects for the DS3/E3 Interface Type", RFC 2496, January 1999.

[RFC2558] Tesink, K., "Definitions of Managed Objects for the SONET/SDH Interface Type", RFC 2558, March 1999.

[RFC2662] Bathrick, G., and F. Ly, "Definitions of Managed Objects for the ADSL Lines", RFC 2662, August 1999.

[RFC2570] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[RFC2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[RFC2572] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[RFC2573] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

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

[RFC2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

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

1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J.,

Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

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

[RFC2856] Bierman, A., McCloghrie, K., Presuhn, R., "Textual Conventions for Additional High Capacity Data Types", RFC2856, June 2000.

[VDSL-MIB] Abbi, R., Ray, B., work in progress.

6. Security Considerations

This module does not define any management objects. Instead, it defines a set of textual conventions which may be used by other MIB modules to define management objects.

Meaningful security considerations can only be written in the modules that define management objects.

7. Author's Address

Rajesh Abbi
Alcatel USA
2912 Wake Forest Road
Raleigh, NC 27609-7860 USA
EMail: Rajesh.Abbi@usa.alcatel.com
Phone: +1 919 850 6194

Bob Ray
EMail: bob_ray_99@yahoo.com

8. Intellectual Property

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. Full Copyright Statement