Network Working Group
Request for Comments: 1658
Obsoletes: 1316
Category: Standards Track

B. Stewart
Xyplex, Inc.

July 1994

Definitions of Managed Objects for Character Stream Devices using SMIv2

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.

Table of Contents

1. Introduction

This memo defines an extension to the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for the management of character stream devices.

2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- RFC 1442 [1] which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- STD 17, RFC 1213 [2] defines MIB-II, the core set of managed objects for the Internet suite of protocols.

- RFC 1445 [3] which defines the administrative and other architectural aspects of the framework.

- RFC 1448 [4] which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

3. Overview

The Character MIB applies to ports that carry a character stream, whether physical or virtual, serial or parallel, synchronous or asynchronous. The most common example of a character stream device is a hardware terminal port with an RS-232 interface. Another common hardware example is a parallel printer port, say with a Centronics interface. The concept also includes virtual terminal ports, such as a software connection point for a remote console.

The Character MIB is mandatory for all systems that offer character stream ports. This includes, for example, terminal servers, general-purpose time-sharing hosts, and even such systems as a bridge with a (virtual) console port. It may or may not include character ports that do not support network sessions, depending on the system's needs.

The Character MIB's central abstraction is a port. Physical ports have a one-to-one correspondence with hardware ports. Virtual ports are software entities analogous to physical ports, but with no hardware connector.

Each port supports one or more sessions. A session represents a virtual connection that carries characters between the port and some partner. Sessions typically operate over a stack of network protocols. A typical session, for example, uses Telnet over TCP.

The MIB comprises one base object and two tables, detailed in the following sections. The tables contain objects for ports and sessions.

The MIB intentionally contains no distinction between what is often called permanent and operational or volatile data bases. For the purposes of this MIB, handling of such distinctions is implementation specific.

3.1. Relationship to Interface MIB

The Character MIB does not relate directly to the Interface MIB [1], since it is not intrinsically a network interface. On the other hand, in most implementations where it is present, it will be above a physical sublayer interface, such as the RS-232-like [2] or Parallel-printer-like [3] MIBs. Such physical interfaces typically are represented by a row in the interface table (ifTable), identified by a value of ifIndex.

4. Definitions

```
CHARACTER-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Counter32, Integer32, Gauge32, TimeTicks
        FROM SNMPv2-SMI
    AutonomousType, InstancePointer
        FROM SNMPv2-TC
    InterfaceIndex
        FROM IF-MIB
    transmission, mib-2
        FROM RFC1213-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;


char MODULE-IDENTITY
    LAST-UPDATED "9405261700Z"
    ORGANIZATION "IETF Character MIB Working Group"
    CONTACT-INFO
            "       Bob Stewart
             Postal: Xyplex, Inc.
                     295 Foster Street
                     Littleton, MA 01460

                Tel: 508-952-4816
                Fax: 508-952-4887
```

```
                E-mail: rlstewart@eng.xyplex.com"
        DESCRIPTION
                "The MIB module for character stream devices."
        ::= { mib-2 19 }

    PortIndex ::= TEXTUAL-CONVENTION
        DISPLAY-HINT "d"
        STATUS current
        DESCRIPTION
                "A unique value, greater than zero, for each
                character port in the managed system.  It is
                recommended that values are assigned contiguously
                starting from 1.  The value for each interface sub-
                layer must remain constant at least from one re-
                initialization of the entity's network management
                system to the next re-initialization.

                In a system where the character ports are attached
                to hardware represented by an ifIndex, it is
                conventional, but not required, to make the
                character port index equal to the corresponding
                ifIndex."
        SYNTAX Integer32


    -- Generic Character information

    charNumber OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The number of entries in charPortTable, regardless
            of their current state."
        ::= { char 1 }


    -- the Character Port table

    charPortTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CharPortEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "A list of port entries.  The number of entries is
            given by the value of charNumber."
        ::= { char 2 }
```

```
   charPortEntry OBJECT-TYPE
       SYNTAX CharPortEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "Status and parameter values for a character port."
       INDEX { charPortIndex }
       ::= { charPortTable 1 }

   CharPortEntry ::=
       SEQUENCE {
           charPortIndex
               PortIndex,
           charPortName
               DisplayString,
           charPortType
               INTEGER,
           charPortHardware
               AutonomousType,
           charPortReset
               INTEGER,
           charPortAdminStatus
               INTEGER,
           charPortOperStatus
               INTEGER,
           charPortLastChange
               TimeTicks,
           charPortInFlowType
               INTEGER,
           charPortOutFlowType
               INTEGER,
           charPortInFlowState
               INTEGER,
           charPortOutFlowState
               INTEGER,
           charPortInCharacters
               Counter32,
           charPortOutCharacters
               Counter32,
           charPortAdminOrigin
               INTEGER,
           charPortSessionMaximum
               INTEGER,
           charPortSessionNumber
               Gauge32,
           charPortSessionIndex
               INTEGER,
           charPortInFlowTypes
```

```
            OCTET STRING,
        charPortOutFlowTypes
            OCTET STRING,
        charPortLowerIfIndex
            InterfaceIndex
    }

charPortIndex OBJECT-TYPE
    SYNTAX PortIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A unique value for each character port, perhaps
        corresponding to the same value of ifIndex when the
        character port is associated with a hardware port
        represented by an ifIndex."
    ::= { charPortEntry 1 }

charPortName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..32))
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "An administratively assigned name for the port,
        typically with some local significance."
    ::= { charPortEntry 2 }

charPortType OBJECT-TYPE
    SYNTAX INTEGER { physical(1), virtual(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The port's type, 'physical' if the port represents
        an external hardware connector, 'virtual' if it does
        not."
    ::= { charPortEntry 3 }

charPortHardware OBJECT-TYPE
    SYNTAX AutonomousType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A reference to hardware MIB definitions specific to
        a physical port's external connector.  For example,
        if the connector is RS-232, then the value of this
        object refers to a MIB sub-tree defining objects
        specific to RS-232.  If an agent is not configured
        to have such values, the agent returns the object
```

```
            identifier:

                nullHardware OBJECT IDENTIFIER ::= { 0 0 }
            "
        ::= { charPortEntry 4 }

    charPortReset OBJECT-TYPE
        SYNTAX INTEGER { ready(1), execute(2) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "A control to force the port into a clean, initial
            state, both hardware and software, disconnecting all
            the port's existing sessions.  In response to a
            get-request or get-next-request, the agent always
            returns 'ready' as the value.  Setting the value to
            'execute' causes a reset."
        ::= { charPortEntry 5 }

    charPortAdminStatus OBJECT-TYPE
        SYNTAX INTEGER { enabled(1), disabled(2), off(3),
                         maintenance(4) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The port's desired state, independent of flow
            control.  'enabled' indicates that the port is
            allowed to pass characters and form new sessions.
            'disabled' indicates that the port is allowed to
            pass characters but not form new sessions.  'off'
            indicates that the port is not allowed to pass
            characters or have any sessions. 'maintenance'
            indicates a maintenance mode, exclusive of normal
            operation, such as running a test.

            'enabled' corresponds to ifAdminStatus 'up'.
            'disabled' and 'off' correspond to ifAdminStatus
            'down'.  'maintenance' corresponds to ifAdminStatus
            'test'."
        ::= { charPortEntry 6 }

    charPortOperStatus OBJECT-TYPE
        SYNTAX INTEGER { up(1), down(2),
                         maintenance(3), absent(4), active(5) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The port's actual, operational state, independent
```

```
            of flow control.  'up' indicates able to function
            normally.  'down' indicates inability to function
            for administrative or operational reasons.
            'maintenance' indicates a maintenance mode,
            exclusive of normal operation, such as running a
            test.  'absent' indicates that port hardware is not
            present.  'active' indicates up with a user present
            (e.g. logged in).

            'up' and 'active' correspond to ifOperStatus 'up'.
            'down' and 'absent' correspond to ifOperStatus
            'down'.  'maintenance' corresponds to ifOperStatus
            'test'."
        ::= { charPortEntry 7 }

    charPortLastChange OBJECT-TYPE
        SYNTAX TimeTicks
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The value of sysUpTime at the time the port entered
            its current operational state.  If the current state
            was entered prior to the last reinitialization of
            the local network management subsystem, then this
            object contains a zero value."
        ::= { charPortEntry 8 }

    -- charPortInFlowType is deprecated in favor of
    -- charPortInFlowTypes

    charPortInFlowType OBJECT-TYPE
        SYNTAX INTEGER { none(1), xonXoff(2), hardware(3),
                         ctsRts(4), dsrDtr(5) }
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "The port's type of input flow control.  'none'
            indicates no flow control at this level or below.
            'xonXoff' indicates software flow control by
            recognizing XON and XOFF characters.  'hardware'
            indicates flow control delegated to the lower level,
            for example a parallel port.

            'ctsRts' and 'dsrDtr' are specific to RS-232-like
            ports.  Although not architecturally pure, they are
            included here for simplicity's sake."
        ::= { charPortEntry 9 }
```

```
   -- charPortOutFlowType is deprecated in favor of
   -- charPortOutFlowTypes

   charPortOutFlowType OBJECT-TYPE
       SYNTAX INTEGER { none(1), xonXoff(2), hardware(3),
                        ctsRts(4), dsrDtr(5) }
       MAX-ACCESS read-write
       STATUS deprecated
       DESCRIPTION
           "The port's type of output flow control.  'none'
           indicates no flow control at this level or below.
           'xonXoff' indicates software flow control by
           recognizing XON and XOFF characters.  'hardware'
           indicates flow control delegated to the lower level,
           for example a parallel port.

           'ctsRts' and 'dsrDtr' are specific to RS-232-like
           ports.  Although not architecturally pure, they are
           included here for simplicy's sake."
       ::= { charPortEntry 10 }

   charPortInFlowState OBJECT-TYPE
       SYNTAX INTEGER { none(1), unknown(2), stop(3), go(4) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The current operational state of input flow control
           on the port.  'none' indicates not applicable.
           'unknown' indicates this level does not know.
           'stop' indicates flow not allowed.  'go' indicates
           flow allowed."
       ::= { charPortEntry 11 }

   charPortOutFlowState OBJECT-TYPE
       SYNTAX INTEGER { none(1), unknown(2), stop(3), go(4) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The current operational state of output flow
           control on the port.  'none' indicates not
           applicable.  'unknown' indicates this level does not
           know.  'stop' indicates flow not allowed.  'go'
           indicates flow allowed."
       ::= { charPortEntry 12 }

   charPortInCharacters OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
```

```
        STATUS current
        DESCRIPTION
            "Total number of characters detected as input from
            the port since system re-initialization and while
            the port operational state was 'up', 'active', or
            'maintenance', including, for example, framing, flow
            control (i.e. XON and XOFF), each occurrence of a
            BREAK condition, locally-processed input, and input
            sent to all sessions."
        ::= { charPortEntry 13 }

    charPortOutCharacters OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Total number of characters detected as output to
            the port since system re-initialization and while
            the port operational state was 'up', 'active', or
            'maintenance', including, for example, framing, flow
            control (i.e. XON and XOFF), each occurrence of a
            BREAK condition, locally-created output, and output
            received from all sessions."
        ::= { charPortEntry 14 }

    charPortAdminOrigin OBJECT-TYPE
        SYNTAX INTEGER { dynamic(1), network(2), local(3),
                         none(4) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The administratively allowed origin for
            establishing session on the port.  'dynamic' allows
            'network' or 'local' session establishment. 'none'
            disallows session establishment."
        ::= { charPortEntry 15 }

    charPortSessionMaximum OBJECT-TYPE
        SYNTAX INTEGER (-1..2147483647)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The maximum number of concurrent sessions allowed
            on the port.  A value of -1 indicates no maximum.
            Setting the maximum to less than the current number
            of sessions has unspecified results."
        ::= { charPortEntry 16 }
```

```
   charPortSessionNumber OBJECT-TYPE
       SYNTAX Gauge32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The number of open sessions on the port that are in
           the connecting, connected, or disconnecting state."
       ::= { charPortEntry 17 }

   charPortSessionIndex OBJECT-TYPE
       SYNTAX INTEGER (0..2147483647)
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The value of charSessIndex for the port's first or
           only active session.  If the port has no active
           session, the agent returns the value zero."
       ::= { charPortEntry 18 }

   charPortInFlowTypes OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE (1))
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
           "The port's types of input flow control at the
           software level.  Hardware-level flow control is
           independently controlled by the appropriate
           hardware-level MIB.

           A value of zero indicates no flow control.
           Depending on the specific implementation, any or
           all combinations of flow control may be chosen by
           adding the values:

           128  xonXoff, recognizing XON and XOFF characters
           64   enqHost, ENQ/ACK to allow input to host
           32   enqTerm, ACK to allow output to port
           "
       ::= { charPortEntry 19 }

   charPortOutFlowTypes OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE (1))
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
           "The port's types of output flow control at the
           software level.  Hardware-level flow control is
           independently controlled by the appropriate
```

```
            hardware-level MIB.

            A value of zero indicates no flow control.
            Depending on the specific implementation, any or
            all combinations of flow control may be chosen by
            adding the values:

            128  xonXoff, recognizing XON and XOFF characters
            64   enqHost, ENQ/ACK to allow input to host
            32   enqTerm, ACK to allow output to port
            "
        ::= { charPortEntry 20 }

    charPortLowerIfIndex OBJECT-TYPE
        SYNTAX InterfaceIndex
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The ifIndex value of the lower level hardware supporting
            this character port, zero if none."
        ::= { charPortEntry 21 }


    -- the Character Session table

    charSessTable OBJECT-TYPE
        SYNTAX SEQUENCE OF CharSessEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "A list of port session entries."
        ::= { char 3 }

    charSessEntry OBJECT-TYPE
        SYNTAX CharSessEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Status and parameter values for a character port
            session."
        INDEX { charSessPortIndex, charSessIndex }
        ::= { charSessTable 1 }

    CharSessEntry ::=
        SEQUENCE {
            charSessPortIndex
                PortIndex,
            charSessIndex
```

```
            INTEGER,
        charSessKill
            INTEGER,
        charSessState
            INTEGER,
        charSessProtocol
            AutonomousType,
        charSessOperOrigin
            INTEGER,
        charSessInCharacters
            Counter32,
        charSessOutCharacters
            Counter32,
        charSessConnectionId
            InstancePointer,
        charSessStartTime
            TimeTicks
    }

charSessPortIndex OBJECT-TYPE
    SYNTAX PortIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of charPortIndex for the port to which
        this session belongs."
    ::= { charSessEntry 1 }

charSessIndex OBJECT-TYPE
    SYNTAX INTEGER (1..2147483647)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The session index in the context of the port, a
        non-zero positive integer.  Session indexes within a
        port need not be sequential.  Session indexes may be
        reused for different ports.  For example, port 1 and
        port 3 may both have a session 2 at the same time.
        Session indexes may have any valid integer value,
        with any meaning convenient to the agent
        implementation."
    ::= { charSessEntry 2 }

charSessKill OBJECT-TYPE
    SYNTAX INTEGER { ready(1), execute(2) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
            "A control to terminate the session.  In response to
            a get-request or get-next-request, the agent always
            returns 'ready' as the value.  Setting the value to
            'execute' causes termination."
        ::= { charSessEntry 3 }

    charSessState OBJECT-TYPE
        SYNTAX INTEGER { connecting(1), connected(2),
                         disconnecting(3) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The current operational state of the session,
            disregarding flow control.  'connected' indicates
            that character data could flow on the network side
            of session.  'connecting' indicates moving from
            nonexistent toward 'connected'.  'disconnecting'
            indicates moving from 'connected' or 'connecting' to
            nonexistent."
        ::= { charSessEntry 4 }

    charSessProtocol OBJECT-TYPE
        SYNTAX AutonomousType
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The network protocol over which the session is
            running.  Other OBJECT IDENTIFIER values may be
            defined elsewhere, in association with specific
            protocols.  However, this document assigns those of
            known interest as of this writing."
        ::= { charSessEntry 5 }

    wellKnownProtocols OBJECT IDENTIFIER ::= { char 4 }

    protocolOther  OBJECT IDENTIFIER ::= { wellKnownProtocols 1 }
    protocolTelnet OBJECT IDENTIFIER ::= { wellKnownProtocols 2 }
    protocolRlogin OBJECT IDENTIFIER ::= { wellKnownProtocols 3 }
    protocolLat    OBJECT IDENTIFIER ::= { wellKnownProtocols 4 }
    protocolX29    OBJECT IDENTIFIER ::= { wellKnownProtocols 5 }
    protocolVtp    OBJECT IDENTIFIER ::= { wellKnownProtocols 6 }


    charSessOperOrigin OBJECT-TYPE
        SYNTAX INTEGER { unknown(1), network(2), local(3) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
```

```
            "The session's source of establishment."
        ::= { charSessEntry 6 }

    charSessInCharacters OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This session's subset of charPortInCharacters."
        ::= { charSessEntry 7 }

    charSessOutCharacters OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This session's subset of charPortOutCharacters."
        ::= { charSessEntry 8 }

    charSessConnectionId OBJECT-TYPE
        SYNTAX InstancePointer
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "A reference to additional local MIB information.
            This should be the highest available related MIB,
            corresponding to charSessProtocol, such as Telnet.
            For example, the value for a TCP connection (in the
            absence of a Telnet MIB) is the object identifier of
            tcpConnState.  If an agent is not configured to have
            such values, the agent returns the object
            identifier:

                nullConnectionId OBJECT IDENTIFIER ::= { 0 0 }
            "
        ::= { charSessEntry 9 }

    charSessStartTime OBJECT-TYPE
        SYNTAX TimeTicks
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The value of sysUpTime in MIB-2 when the session
            entered connecting state."
        ::= { charSessEntry 10 }
```

```
   -- conformance information

   charConformance OBJECT IDENTIFIER ::= { char 5 }

   charGroups      OBJECT IDENTIFIER ::= { charConformance 1 }
   charCompliances OBJECT IDENTIFIER ::= { charConformance 2 }


   -- compliance statements

   charCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for SNMPv2 entities
               which have Character hardware interfaces."

       MODULE  -- this module
           MANDATORY-GROUPS { charGroup }
       ::= { charCompliances 1 }


   -- units of conformance

   charGroup    OBJECT-GROUP
       OBJECTS { charNumber, charPortIndex, charPortName,
                 charPortType, charPortHardware, charPortReset,
                 charPortAdminStatus, charPortOperStatus,
                 charPortLastChange,
                 charPortInFlowState, charPortOutFlowState,
                 charPortAdminOrigin, charPortSessionMaximum,
                 charPortInFlowTypes, charPortOutFlowTypes,
                 charPortInCharacters, charPortOutCharacters,
                 charPortSessionNumber, charPortSessionIndex,
                 charPortLowerIfIndex,
                 charSessPortIndex, charSessIndex,
                 charSessKill, charSessState,
                 charSessProtocol, charSessOperOrigin,
                 charSessInCharacters, charSessOutCharacters,
                 charSessConnectionId, charSessStartTime }
       STATUS  current
       DESCRIPTION
               "A collection of objects providing information
                applicable to all Character interfaces."
       ::= { charGroups 1 }

   END
```

5. Acknowledgements

This memo was produced by the IETF Character MIB Working Group.

6. References

[1] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research,Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[2] McCloghrie, K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.

[3] Galvin, J., and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Trusted Information Systems, Hughes LAN Systems, April 1993.

[4] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research,Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.

[5] McCloghrie, K., and F. Kastenholz, "Evolution of the Interfaces Group of MIB-II", RFC 1573, Hughes LAN Systems, FTP Software, January 1994.

[6] Stewart, B., "Definitions of Managed Objects for RS-232-like Hardware Devices using SMIv2", RFC 1659, Xyplex, Inc., July 1994.

[7] Stewart, B., "Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2", RFC 1660, Xyplex, Inc., July 1994.

7. Security Considerations

Security issues are not discussed in this memo.

8. Author's Address

Bob Stewart
Xyplex, Inc.
295 Foster Street
Littleton, MA 01460

Phone: 508-952-4816
Fax: 508-952-4887
EMail: rlstewart@eng.xyplex.com