Network Working Group
Request for Comments: 1660
Obsoletes: 1318
Category: Standards Track

B. Stewart
Xyplex, Inc.

July 1994

Definitions of Managed Objects for Parallel-printer-like Hardware Devices using SMIv2

Status of this Memo

This document specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" 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 Parallel-printer-like 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 Parallel-printer-like Hardware Device MIB applies to interface ports that would most probably support the Character MIB. The most common example is Centronics-like printer port.

The Parallel-printer-like Hardware Device MIB is mandatory for all systems that have such a hardware port supporting services managed through some other MIB.

The Parallel-printer-like Hardware Port MIB includes Centronics-like and Data-Products-like parallel physical links with a similar set of control signals.

The MIB contains objects that relate to physical layer connections.

The MIB comprises one base object and three tables, detailed in the following sections. The tables contain objects for ports and input and output control signals.

3.1. Relationship to Interface MIB

The Parallel-printer-like MIB is one of many MIBs designed for layered use as described in the Interface MIB [5]. In most implementations where it is present, it will be in the lowest interface sublayer, that is, the Parallel-printer-like MIB represents the physical layer, providing service to higher layers such as the

Character MIB [6].

Although it is unlikely that a parallel printer port will actually be used as a network interface, which is the intent of the Interface MIB, the Parallel-printer-like MIB is closely connected to the Character MIB, which can share hardware interfaces with network operation, and relate to the RS-232 MIB [7].

The Interface MIB's ifTestTable and ifRcvAddressTable are not relevant to the Parallel-printer-like MIB.

The Parallel-printer-like MIB is relevant for ifType values para(34) and perhaps others.

The Parallel-printer-like MIB requires the conformance groups ifGeneralGroup, and ifFixedLengthGroup.

Usefulness of error counters in this MIB depends on the octet counters in ifFixedLengthGroup.

4. Definitions

```
PARALLEL-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Counter32, Integer32
        FROM SNMPv2-SMI
    InterfaceIndex
        FROM IF-MIB
    transmission
        FROM RFC1213-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;


para 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 Parallel-printer-like hardware devices."
        ::= { transmission 34 }


    -- Generic Parallel-printer-like information

    paraNumber OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The number of ports (regardless of their current
            state) in the Parallel-printer-like port table."
        ::= { para 1 }


    -- the Parallel-printer-like Port table

    paraPortTable OBJECT-TYPE
        SYNTAX SEQUENCE OF ParaPortEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "A list of port entries.  The number of entries is
            given by the value of paraNumber."
        ::= { para 2 }

    paraPortEntry OBJECT-TYPE
        SYNTAX ParaPortEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Status and parameter values for a port."
        INDEX { paraPortIndex }
        ::= { paraPortTable 1 }

    ParaPortEntry ::=
        SEQUENCE {
            paraPortIndex
                InterfaceIndex,
            paraPortType
                INTEGER,
            paraPortInSigNumber
                Integer32,
            paraPortOutSigNumber
                Integer32
        }
```

```
   paraPortIndex OBJECT-TYPE
       SYNTAX InterfaceIndex
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The value of ifIndex for the port.  By convention
           and if possible, hardware port numbers map directly
           to external connectors.  The value for each port must
           remain constant at least from one re-initialization
           of the network management agent to the next."
       ::= { paraPortEntry 1 }

   paraPortType OBJECT-TYPE
       SYNTAX INTEGER {
           other(1),
           centronics(2),
           dataproducts(3)
       }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The port's hardware type."
       ::= { paraPortEntry 2 }

   paraPortInSigNumber OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The number of input signals for the port in the
           input signal table (paraPortInSigTable).  The table
           contains entries only for those signals the software
           can detect and that are useful to observe."
       ::= { paraPortEntry 3 }

   paraPortOutSigNumber OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The number of output signals for the port in the
           output signal table (paraPortOutSigTable).  The
           table contains entries only for those signals the
           software can assert and that are useful to observe."
       ::= { paraPortEntry 4 }
```

```
   -- Parallel-printer-like Input Signal Table

   paraInSigTable OBJECT-TYPE
       SYNTAX SEQUENCE OF ParaInSigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "A list of port input control signal entries."
       ::= { para 3 }

   paraInSigEntry OBJECT-TYPE
       SYNTAX ParaInSigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "Input control signal status for a hardware port."
       INDEX { paraInSigPortIndex, paraInSigName }
       ::= { paraInSigTable 1 }

   ParaInSigEntry ::=
       SEQUENCE {
           paraInSigPortIndex
               InterfaceIndex,
           paraInSigName
               INTEGER,
           paraInSigState
               INTEGER,
           paraInSigChanges
               Counter32
       }

   paraInSigPortIndex OBJECT-TYPE
       SYNTAX InterfaceIndex
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The value of paraPortIndex for the port to which
           this entry belongs."
       ::= { paraInSigEntry 1 }

   paraInSigName OBJECT-TYPE
       SYNTAX INTEGER { power(1), online(2), busy(3),
                        paperout(4), fault(5) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "Identification of a hardware signal."
       ::= { paraInSigEntry 2 }
```

```
   paraInSigState OBJECT-TYPE
       SYNTAX INTEGER { none(1), on(2), off(3) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The current signal state."
       ::= { paraInSigEntry 3 }

   paraInSigChanges OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The number of times the signal has changed from
           'on' to 'off' or from 'off' to 'on'."
       ::= { paraInSigEntry 4 }


   -- Output Signal Table

   paraOutSigTable OBJECT-TYPE
       SYNTAX SEQUENCE OF ParaOutSigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "A list of port output control signal entries."
       ::= { para 4 }

   paraOutSigEntry OBJECT-TYPE
       SYNTAX ParaOutSigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
           "Output control signal status for a hardware port."
       INDEX { paraOutSigPortIndex, paraOutSigName }
       ::= { paraOutSigTable 1 }

   ParaOutSigEntry ::=
       SEQUENCE {
           paraOutSigPortIndex
               InterfaceIndex,
           paraOutSigName
               INTEGER,
           paraOutSigState
               INTEGER,
           paraOutSigChanges
               Counter32
       }
```

```
   paraOutSigPortIndex OBJECT-TYPE
       SYNTAX InterfaceIndex
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The value of paraPortIndex for the port to which
           this entry belongs."
       ::= { paraOutSigEntry 1 }

   paraOutSigName OBJECT-TYPE
       SYNTAX INTEGER { power(1), online(2), busy(3),
                        paperout(4), fault(5) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "Identification of a hardware signal."
       ::= { paraOutSigEntry 2 }

   paraOutSigState OBJECT-TYPE
       SYNTAX INTEGER { none(1), on(2), off(3) }
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The current signal state."
       ::= { paraOutSigEntry 3 }

   paraOutSigChanges OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The number of times the signal has changed from
           'on' to 'off' or from 'off' to 'on'."
       ::= { paraOutSigEntry 4 }


   -- conformance information

   paraConformance OBJECT IDENTIFIER ::= { para 5 }

   paraGroups      OBJECT IDENTIFIER ::= { paraConformance 1 }
   paraCompliances OBJECT IDENTIFIER ::= { paraConformance 2 }
```

```
   -- compliance statements

   paraCompliance MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for SNMPv2 entities
               which have Parallel-printer-like hardware
               interfaces."

       MODULE  -- this module
           MANDATORY-GROUPS { paraGroup }
       ::= { paraCompliances 1 }


   -- units of conformance

   paraGroup    OBJECT-GROUP
       OBJECTS { paraNumber, paraPortIndex, paraPortType,
                 paraPortInSigNumber, paraPortOutSigNumber,
                 paraInSigPortIndex, paraInSigName,
                 paraInSigState, paraInSigChanges,
                 paraOutSigPortIndex, paraOutSigName,
                 paraOutSigState, paraOutSigChanges }
       STATUS  current
       DESCRIPTION
               "A collection of objects providing information
                applicable to all Parallel-printer-like interfaces."
       ::= { paraGroups 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 Character Stream Devices using SMIv2", RFC 1658, Xyplex, Inc., July 1994.

[7] Stewart, B., "Definitions of Managed Objects for RS-232-like Devices using SMIv2", RFC 1659, 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