Definitions of Managed Objects for DDS Interface Types

Mon Apr 8 9:00:00 EST 1996
Expires: October 13, 1996

draft-ietf-trunkmib-dds-mib-00.txt

Mark A. Cotton
Eastern Research, Inc.
mcotton@erinc.com

Status of this Memo

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing Digital Data System (DDS) interfaces. Along with the Definition of Managed Objects for RS-232-like Hardware Devices using SMIv2 [TBD], this memo also provides basic management capabilities for DDS DSU/CSU-like interfaces.

This memo specifies a MIB module in a manner that is both compliant to the SNMPv2 SMI, and semantically identical to the peer SNMPv1

definitions.

Introduction

This document reflects work being done by the trunk-mib working group (trunk-mib@cisco.com). This document defines a MIB that allows DDS-type interfaces to be managed via SNMP. This is an attempt to ensure that SNMP compliant DDS devices have a common MIB. An attempt has been made to include devices which support DDS secondary channel capability. This document is intended to allow for the SNMP managment of the basic DDS CSU/DSU, with and without rate adaption.

Table of Contents

1. 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. Objects

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) [6] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. 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 OBJECT DESCRIPTOR, to also refer to the object type.

2.1. Format of Definitions

Section 4 contains contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions that are specified in RFC1442 [1], RFC1445 [3], and RFC1448 [4].

3. Overview

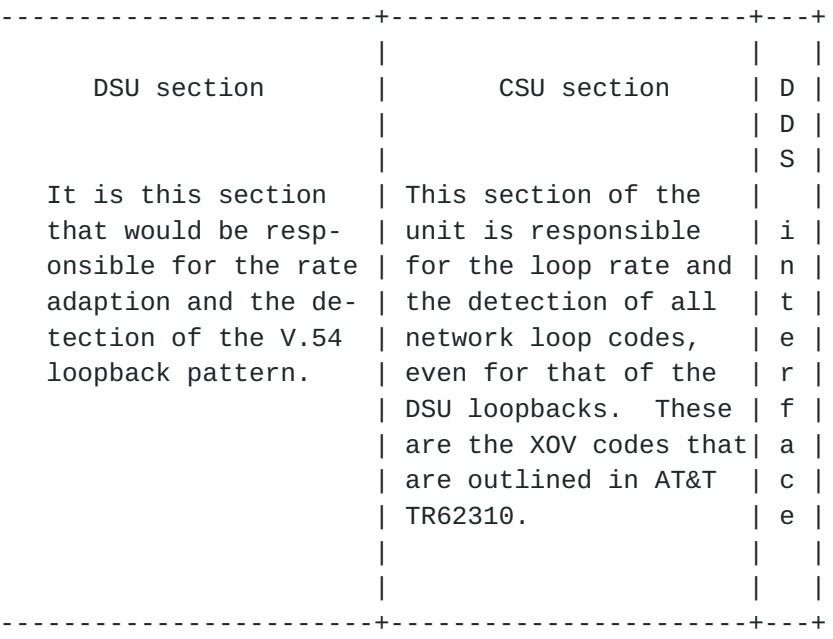
This document defines managed objects for a Digital Dataphone Service (DDS) interface. At present these objects apply to an interface

with an ifType value dds (TBD).

Editors note: The ifType value dds needs to be allocated by the IANA prior to this memo's publication.

The following diagram shows the various internal organization of

a typical DDS DSU/CSU.



The MIB contained herein describes objects for the management of configuration, diagnostics, alarms, and statistics related to DDS DSU/CSUs. These definitions are based on the AT&T DS0 Local Digital Channel Description and Interface Specification (TR 62310 [8]).

The objects defined in this memo provide instrumentation for the Network Interface. The instrumentation for the Data Interface, is provided by the managed objects for the given type of interface, e.g. RFC 1695 [TBD] for the case where the Data Interface is an RS-232-like interface or RFC TBD [TBD] for the case where the Data interface is a DS0 on a DS0-like interface.

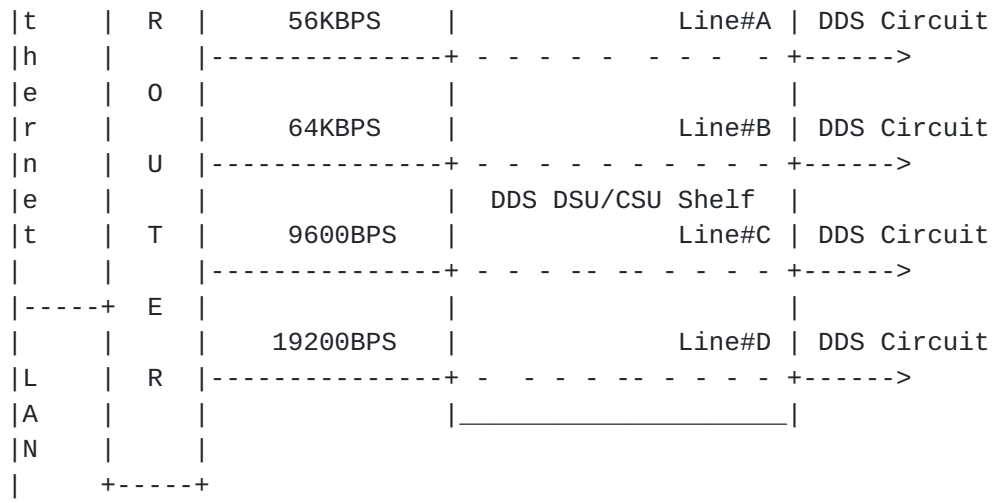
A interface with the ifType of dds(TBD) implements the ifGeneral group defined in the IF-MIB [TBD] and uses the standard interfaces

objects as follows:

ifTable Object	Use for DDS Interfaces
ifIndex	Interface index.
ifDescr	As per the interfaces MIB [TBD]
ifType	dds(TBD)
ifSpeed	The speed at which the DDS interface is operating. This is the same as the value in ddsPrimaryChannelSpeed except when it has a value of 64 kbits/s.

	In that case, ifSpeed will have a value of 72 kbits/s
ifPhysAddress	The value of the Circuit Identifier assigned by the service provider. If there is no value assigned, this object should have a length of zero..
ifAdminStatus	As per the Interfaces MIB [TBD]
ifOperStatus	As per the Interfaces MIB [TBD]
ifLastChange	As per the Interfaces MIB [TBD]
ifName	As per the interfaces MIB [TBD].
ifLinkUpDownTrapEnable	Set to enabled(1).
ifHighSpeed	Zero (the maximum line speed is 72 kbits/s).
ifConnectorPresent	true(1)

The following diagram demonstrates how an SNMP managable DDS DSU/CSU shelf could be connected to a router to allow the router WANs access to the DDS circuits.



The assignment of the index values could for example be:

ifIndex ifType Description

1	rs232(33)	Line A - Data Interface
2	rs232(33)	Line B - Data Interface
3	dds (TBD)	Line A - Network Interface
4	dds (TBD)	Line B - Network Interface
5	rs232(33)	Line C - Data Interface
6	dds (TBD)	Line C - Network Interface
7	rs232(33)	Line D - Data Interface
8	dds (TBD)	Line D - Network Interface
9	rs232(33)	Router
10	rs232(33)	Router
11	rs232(33)	Router
12	rs232(33)	Router
13	ethernetCsmaCd	Ethernet port

For this example, ifNumber is equal to 13. Note the following description of ifIndex: it describes ports on the managed device. In this example, some ports are RS-232-like interfaces, some are DDS-like interfaces, some are RS-232-like interfaces that are not associated with the DDS DSU/CSU lines, and there is one port which is the ethernet port on the device.

The relationship between the Data Interface and the corresponding Network Interface is captured in the ifStackTable.

ifStackTable Entries for proxy-managed DSU/CSU shelf:

Higher Layer	Lower Layer
1	3
2	4
5	6
7	8
9	1
10	2
11	5
12	7

If the DSU/CSU shelf is managed by itself by a local SNMP Agent, that is to say not managed by the router-based agent, the situation would be as follows.

The assignment of the index values could for example be:

ifIndex	ifType	Description
1	dds (TBD)	Line A - Network Interface
2	rs232(33)	Line A - Data Interface
3	dds (TBD)	Line B - Network Interface
4	rs232(33)	Line B - Data Interface

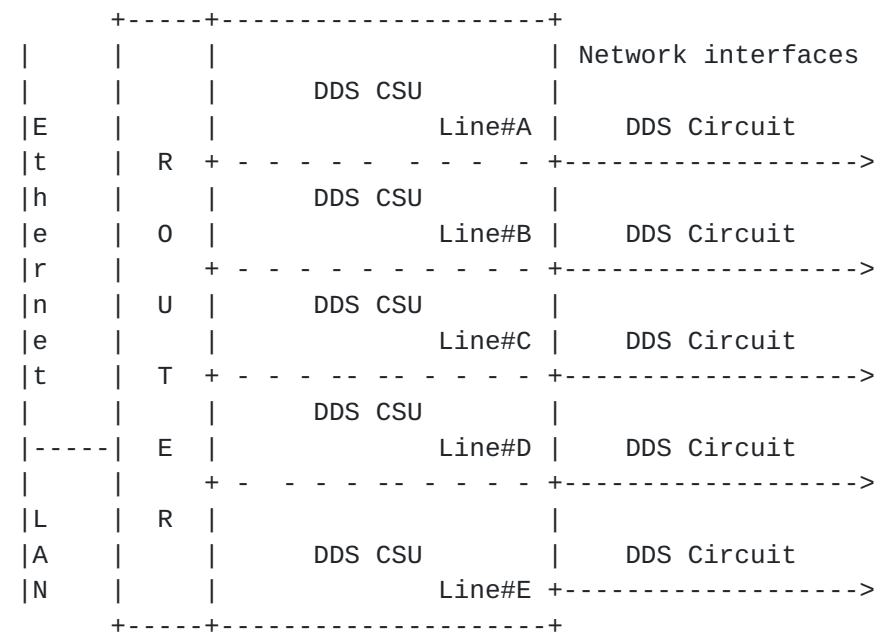
5	dds (TBD)	Line C - Network Interface
6	rs232(33)	Line C - Data Interface
7	dds (TBD)	Line D - Network Interface
8	rs232(33)	Line D - Data Interface

Again, the relationship between the Data Interface and the corresponding Network Interface is captured in the ifStackTable.

ifStackTable Entries for directly-managed DSU/CSU shelf:

Higher Layer	Lower Layer
2	1
4	3
6	5
8	7

The next diagram demonstrates how an SNMP managable CSU might be integrated within the router itself.



If the DDS CSUs are integrated within the router itself, the interface organization might be as follows.

The assignment of the index values could for example be:

ifIndex	ifType	Description
1	dds (TBD)	Line A - Network Interface
2	dds (TBD)	Line B - Network Interface

3	dds (TBD)	Line C - Network Interface
4	dds (TBD)	Line D - Network Interface
5	dds (TBD)	Line E - Network Interface
6	ethernetCsmaCd	Ethernet port

In this example, there is no Data Interface present, because the Network Interfaces terminate within the router. The ifStackTable entries would only show these Network Interfaces alone.

ifStackTable Entries for a router with integral CSUs:

Higher Layer	Lower Layer
1	1
2	2
3	3
4	4

3.1. Binding between ifIndex and DDS Interfaces

For DDS circuits with the secondary channel activated, it is at the present time unclear how these interfaces will be accessed. Certainly they are capable of being managed via RFC1659, however the binding between the actual interface and an ifIndex that is representative of that interface has not yet been determined.

3.2. DDS Terminology

The terms used in this document, that describe the line conditions of a DDS interface, come from AT&T's technical reference document TR62310 - DS0 Digital Local Channel Description and Interface Specification [8].

3.3. DDS Statistics and Diagnostics

The next sections describe the alarms and diagnostics which directly pertain to DDS DSU/CSUs, as per AT&T TR 62310 [8].

3.3.1 Performance and Availability

The performance terms used are Errored Seconds (ES), Error-Free Seconds (EFS), and Severely Errored Seconds (SES).

An Errored Second is any second during which at least one bit was in error.

An Error-Free Second is a second during which there were no bits in error.

A Severely Errored Second is a second during which the error threshold of 1x10^-3 was exceeded.

3.3.2 Network Alarms and Status Conditions

When a failure occurs in the network, the network will forward a control code (possibly abnormal station code - ASC) toward the CSU at the customer interface. The exact codes transmitted and the conditions necessary for their generation are detailed in section 5.3 (page 11) of AT&T TR 62310 [8].

3.3.3 Network Control Codes

Table 5.3 on page 13 of AT&T TR 62310 specifies the Network Control codes in detail. A further discussion of the nature of the codes is not warranted here.

3.3.4 DDS Interface Loopbacks and Their Methods

The two loopbacks defined by TR 62310 [8] are CSU loopback and DSU loopback. The CSU loopback is intended to loop the network connection back to itself as close as is possible to the network interface (NI). The DSU loopback is usually implemented as a bidirectional loop located a close as possible to the DTE side of the CSU/DSU.

These loopbacks may be activated by either a non-latching method or latching method.

3.3.4.1 Non-Latching Loopbacks

The non-latching loopback is activated when the network sends a loop-code byte alternated with a test pattern byte. The loop must start after the detection of four consecutive bytes of the loop code (CSU or DSU) and remain engaged until five consecutive bytes are received without the loop code. The loop codes must be replaced with a return-code when looped back toward the network. This is used to synchronize the test pattern detector.

3.3.4.2 Latching Loopbacks

Latching loops are named such because a pattern is sent from the network to the CSU/DSU to cause a loopback condition which will remain in effect once the pattern has ceased. On page 17 of TR 62310 [8], the sequence for activating the latching loopbacks are described in detail.

4. Definitions

```
DDS-MIB DEFINITIONS ::= BEGIN
IMPORTS
     MODULE-IDENTITY, OBJECT-TYPE, Counter32
                                     FROM SNMPv2-SMI
     MODULE-COMPLIANCE, OBJECT-GROUP  FROM SNMPv2-CONF
     TruthValue, TimeStamp            FROM SNMPv2-TC
     ifIndex                          FROM IF-MIB
     experimental                     FROM RFC1213-MIB;

--  this is the MIB module for the DDS objects

ddsMIB MODULE-IDENTITY
  LAST-UPDATED "9604080900Z"
  ORGANIZATION "IETF Trunk MIB Working Group"
  CONTACT-INFO
          "        Mark A. Cotton
           Postal: Eastern Research, Inc.
                   225 Executive Drive
                   Moorestown, NJ 08057
              Tel: 609-273-6622
              Fax: 609-273-1847
           E-mail: mcotton@erinc.com"
  DESCRIPTION
          "The MIB module for Digital Dataphone Service
          DSU/CSU-like hardware devices."
  ::= { experimental 99 }


ddsObjects     OBJECT IDENTIFIER ::= { ddsMIB 1 }
ddsGroups      OBJECT IDENTIFIER ::= { ddsMIB 2 }
ddsCompliances OBJECT IDENTIFIER ::= { ddsMIB 3 }


-- The DDS Objects consists of three tables:
--
--         DDS Configuration
--         DDS Diagnostics
--         DDS Statistics
```

```
-- ***************************
-- the DDS Configuration Table
-- ***************************

    ddsConfigTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF DdsConfigEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
           "The DDS Configuration table contains the basic
           configuration variables for the CSU/DSU."
        ::= { ddsObjects 1 }

    ddsConfigEntry OBJECT-TYPE
        SYNTAX  DdsConfigEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
           "An entry in the DDS Configuration table."
        INDEX   { ifIndex }
        ::= { ddsConfigTable 1 }

    DdsConfigEntry ::=
        SEQUENCE {
            ddsPrimaryChannelSpeed   INTEGER,
            ddsAllowSecondaryChannel TruthValue,
            ddsAllowNetworkLoops     TruthValue,
            ddsTransmitClockSource   INTEGER
        }

    ddsPrimaryChannelSpeed OBJECT-TYPE
        SYNTAX  INTEGER {
                    bps2400(1),
                    bps4800(2),
                    bps9600(3),
                    bps19200(4),
                    bps56000(5),
                    bps64000(6)
                }
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This variable configures the speed of the DDS
           circuit.  This object will actually configure the
           speed of the line interface circuitry which connects
           to the DDS line.  The line interface circuitry must
```

be programmed to the same rate as that of the DDS circuit that is provided by the carrier."

```
            ::= { ddsConfigEntry 1 }

        ddsAllowSecondaryChannel OBJECT-TYPE
            SYNTAX  TruthValue
            MAX-ACCESS  read-write
            STATUS  current
            DESCRIPTION
               "This variable allows or disallows the secondary
               DDS channel which will run at the following speeds.

               Primary channel speed   Secondary channel speed
               ---------------------   -----------------------
                       2400 bps                133 bps
                       4800 bps                266 bps
                       9600 bps                533 bps
                       19200 bps               1066 bps
                       56000 bps               2666 bps"

            ::= { ddsConfigEntry 2 }

        ddsAllowNetworkLoops OBJECT-TYPE
            SYNTAX  TruthValue
            MAX-ACCESS  read-write
            STATUS  current
            DESCRIPTION
               "This variable represents the loopback config-
               uration of the DDS interface.  If it is desired
               that the CSU/DSU should not respond to latching
               or non-latching loops from the network, then the
               variable should be set to the disabled state.  If
               it is desirable to have the CSU/DSU respond to loops
               from the network, then this variable should be set
               to enabled."
            ::= { ddsConfigEntry 3 }

        ddsTransmitClockSource OBJECT-TYPE
            SYNTAX  INTEGER {
                              loopTiming(1),
                              localTiming(2),
                              throughTiming(3)
                    }
            MAX-ACCESS  read-write
            STATUS  current
            DESCRIPTION
               "This variable indicates where the CSU/DSU should
               derive its timing from.  The timing can either
               come from the internal oscillator (local), the DTE
               interface (through), or the network interface (loop -
```

```
                the most common option)."
            ::= { ddsConfigEntry 4 }
```

```
-- ************************
-- the DDS Diagnostics Table
-- ************************

    ddsDiagTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF DdsDiagEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
          "The DDS diagnostic table contains the diagnostic element
          variables for the CSU/DSU."
        ::= { ddsMIB 2 }

    ddsDiagEntry OBJECT-TYPE
        SYNTAX  DdsDiagEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
           "An entry in the DDS diagnostic table."
        INDEX   { ifIndex }
        ::= { ddsDiagTable 1 }

    DdsDiagEntry ::=
        SEQUENCE {
            ddsLoopbackConfig   INTEGER,
            ddsSendTestCode     INTEGER,
            ddsInsertTestError  TruthValue,
            ddsTestErrorSeconds Counter32,
            ddsLastTestStart    TimeStamp
        }

    ddsLoopbackConfig OBJECT-TYPE
        SYNTAX  INTEGER (1..7)
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
           "This object contains the loopback configuration of the
           network interface of the DSU/CSU.  It contains various
           fields merged together to form a collection of bits in
           a single variable.  The bit-definitions are as follows.

           1       ddsNoLoop    - No loopback activated.
           2       ddsCSULoop   - Activate the CSU loopback.
                                  This means that the DDS line
                                  should be looped back toward the
                                  network.
```

4 ddsLocalLoop - Activate the local network loop.

```
                                    This engages the local loopback
                                    of the DSU/CSU which means that
                                    the network interface should be
                                    looped back toward itself.
                                    This is used for self-diagnostic
                                    purposes and is not a mode which
                                    can be engaged by the TELCO.

           It should be noted that ddsNoLoop should be the only
           field set if all loops are to be disabled."

       ::= { ddsDiagEntry 1 }

   ddsSendTestCode OBJECT-TYPE
       SYNTAX  INTEGER {
                   sendNoCode(1),
                   sendBERT2047(2),
                   sendAllZeroes(3),
                   sendRemoteLoopCode(4)
               }
       MAX-ACCESS  read-write
       STATUS  current
       DESCRIPTION
          "Activate the bit error-rate tester on the CSU/DSU,
          which would send the BERT pattern toward the network
          interface.  This object is also used for issuing a
          remote loopback command to the far-end DDS DSU/CSU."
       ::= { ddsDiagEntry 2 }

   ddsInsertTestError OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
          "Inserts a single bit error toward the network
          interface. This object will only ever read FALSE."
       ::= { ddsDiagEntry 3 }

   ddsTestErrorSeconds OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
           "The errored-seconds counter.  This object reflects
           the counter which observes the bit-error-rate tester
           errors being received on the network interface of the
           DSU/CSU."
       ::= { ddsDiagEntry 4 }
```

```
        ddsLastTestStart OBJECT-TYPE
```

```
            SYNTAX TimeStamp
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
                "Time stamp for when the BERT test was activated."
            ::= { ddsDiagEntry 5 }


    -- ************************
    -- the DDS Statistics Table
    -- ************************

        ddsStatisticsTable OBJECT-TYPE
            SYNTAX  SEQUENCE OF DdsStatsEntry
            MAX-ACCESS  not-accessible
            STATUS  current
            DESCRIPTION
               "The DDS alarm table contains the statistic counters for
               the CSU/DSU."
            ::= { ddsMIB 3 }

        ddsStatsEntry OBJECT-TYPE
            SYNTAX  DdsStatsEntry
            MAX-ACCESS  not-accessible
            STATUS  current
            DESCRIPTION
               "An entry in the DDS statistics table."
            INDEX   { ifIndex }
            ::= { ddsStatisticsTable 1 }

        DdsStatsEntry ::=
            SEQUENCE {
                ddsLineStatus INTEGER,
                ddsLOSCount   Counter32,
                ddsOOSCount   Counter32,
                ddsCMICount   Counter32,
                ddsOOFCount   Counter32,
                ddsFERRCount  Counter32,
                ddsBPVCount   Counter32
            }

        ddsLineStatus OBJECT-TYPE
            SYNTAX  INTEGER (1..1023)
            MAX-ACCESS  read-only
            STATUS  current
            DESCRIPTION
               "This object contains the line status of the network
               interface of the DDS DSU/CSU merged together to form
```

a collection of bits in a single variable. The bit definitions are as follows.

```
            1       ddsNoAlarm   - No alarm is present.
            2       ddsLOS       - Loss of signal.
            4       ddsOOS       - Out of service.
            8       ddsCMI       - Control mode idle.
            16      ddsOOF       - Out of frame.
            32      ddsFERR      - Frame error.
            64      ddsBPV       - Bipolar violation.
            128     ddsCSULoop   - The CSU loop is active.
            256     ddsLocalLoop - The local network loop is active.
            512     ddsOtherLoop - Some other loopback is active.

            It should be noted that ddsNoAlarm status may only be
            set when no other alarm or diagnostic condition is
            present."

        ::= { ddsStatsEntry 1 }

    ddsLOSCount OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
            "The loss-of-signal errored-second count.  This is
            an error condition that occurs when the line interface
            has detected that no receive signal is present.  This
            is usually declared after receiving 32 consecutive
            zeroes on the receive data stream of the network
            interface."
        ::= { ddsStatsEntry 2 }

    ddsOOSCount OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
            "The out-of-service errored-second count.  This is
            described in section 9.7.4 of AT&T TR 62310."
        ::= { ddsStatsEntry 3 }

    ddsCMICount OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
            "The control-mode-idle errored-seconds count.  This is
            described in section 9.7.4 of AT&T TR 62310."
        ::= { ddsStatsEntry 4 }
```

```
        ddsOOFCount OBJECT-TYPE
```

```
            SYNTAX  Counter32
            MAX-ACCESS  read-only
            STATUS  current
            DESCRIPTION
               "The out-of-frame errored-seconds count.  This is
               described in section 9.7.4 of AT&T TR 62310."
            ::= { ddsStatsEntry 5 }

        ddsFERRCount OBJECT-TYPE
            SYNTAX  Counter32
            MAX-ACCESS  read-only
            STATUS  current
            DESCRIPTION
               "The framing-error errored-second count.  This condition
               may only be detected in 64KBPS mode which uses a framing
               pattern.  This errored-second counter is incremented
               whenever a second is sampled during which a framing
               error occurred."
            ::= { ddsStatsEntry 6 }

        ddsBPVCount OBJECT-TYPE
            SYNTAX  Counter32
            MAX-ACCESS  read-only
            STATUS  current
            DESCRIPTION
               "The bipolar-violation errored-seconds count.  This
               counter is incremented whenever a second is sampled
               during which at least one bipolar violation occurred."
            ::= { ddsStatsEntry 7 }


        -- ***********************
        -- The DDS Interface Group
        -- ***********************

        ddsInterfaceGroup OBJECT-GROUP
            OBJECTS {
                 ddsPrimaryChannelSpeed,
                 ddsAllowSecondaryChannel,
                 ddsAllowNetworkLoops,
                 ddsTransmitClockSource,
                 ddsLoopbackConfig,
                 ddsSendTestCode,
                 ddsInsertTestError,
                 ddsTestErrorSeconds,
                 ddsLastTestStart,
                 ddsLineStatus,
                 ddsLOSCount,
```

```
                ddsOOSCount,
                ddsCMICount,
```

```
                ddsOOFCount,
                ddsFERRCount,
                ddsBPVCount
            }

            STATUS current
            DESCRIPTION
                "The objects required to instrument a Digital Dataphone
                System (DDS) Interface."
            ::= { ddsGroups 1 }


        -- ************************
        -- DDS Interface Compliance
        -- ************************

        ddsInterfaceCompliance MODULE-COMPLIANCE
            STATUS current
            DESCRIPTION
               "The compliance statement for Digital Dataphone System
               (DDS) interfaces."
            MODULE -- this module
            MANDATORY-GROUPS { ddsInterfaceGroup }
            ::= { ddsCompliances 1}


     END
```

5. Acknowledgements

This document was produced by the Trunk MIB Working Group:

Tracy Cox	Bellcore
Fred Baker	Advanced Computer Communications
James Watt	Newbridge
Bill Versteeg	Versteeg Codeworks
Steve Buchko	Newbridge
Greg Celmainis	Newbridge
Kaj Tesink	Bellcore
Al Bryenton	Bell Northern Research
Tom Easterday	CIC
John Labbe	Merit Corporation
Chris Sullivan	Gandalf Ltd
Grant Hall	Gandalf Ltd
Laurence V. Marks	IBM Corp.

Myron Hattig ADC Kentrox

Does my name go in here at all?

James Watt deserves my thanks for working with me in order to ensure that this memo is compliant with the accepted philosophy of the management framework.

I would like to thank Michael Nicolazzo and James Pollock, both of Eastern Research, for supplying their comments and suggestions.

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] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[6] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.

[7] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization, International Standard 8825, December 1987.

[8] AT&T Technical Reference, TR 62310, DS0 Digital Local Channel Description and Interface Specification, August 1993.

[9] AT&T Technical Reference, TR 62411, ACCUNET T1.5 Service Description And Interface Specification, December 1990.

[10] AT&T Technical Reference, TR 62421, ACCUNET Spectrum of Digital Service, December 1989.

[11] CCITT Volume VIII, Recommendation V.54, Loop Test Devices for Modems, November 1980.

7. Security Considerations

Security issues are not discussed in this memo.

8. Author's Address

Mark A. Cotton
Eastern Research, Inc.
225 Executive Drive
Moorestown, New Jersey 08057

Phone: (609)-273-6622
EMail: mcotton@erinc.com