

Network Working Group  
Request for Comments: 2495  
Obsoletes: 1406  
Category: Standards Track

D. Fowler, Editor  
Newbridge Networks  
January 1999

## Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types

### 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.

### Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

### 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 objects used for managing DS1, E1, DS2 and E2 interfaces. This document is a companion document with Definitions of Managed Objects for the DS0 (RFC 2494 [30]), DS3/E3 (RFC 2496 [28]), and the work in progress, SONET/SDH Interface Types.

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.

### Table of Contents

|  |    |
|--|----|
| 1 The SNMP Management Framework .....                  | 2  |
| 1.1 Changes from RFC1406 .....                         | 3  |
| 2 Overview .....                                       | 4  |
| 2.1 Use of ifTable for DS1 Layer .....                 | 5  |
| 2.2 Usage Guidelines .....                             | 6  |
| 2.2.1 Usage of ifStackTable for Routers and DSUs ..... | 6  |
| 2.2.2 Usage of ifStackTable for DS1/E1 on DS2/E2 ..... | 8  |
| 2.2.3 Usage of Channelization for DS3, DS1, DS0 .....  | 9  |
| 2.2.4 Usage of Channelization for DS3, DS2, DS1 .....  | 9  |
| 2.2.5 Usage of Loopbacks .....                         | 10 |
| 2.3 Objectives of this MIB Module .....                | 11 |
| 2.4 DS1 Terminology .....                              | 11 |

|  |    |
|--|----|
| 2.4.1 Error Events .....                                       | 12 |
| 2.4.2 Performance Defects .....                                | 12 |
| 2.4.3 Performance Parameters .....                             | 14 |
| 2.4.4 Failure States .....                                     | 17 |
| 2.4.5 Other Terms .....  | 21 |
| 3 Object Definitions .....                                     | 21 |
| 3.1 The DS1 Near End Group .....                               | 22 |
| 3.1.1 The DS1 Configuration Table .....                        | 22 |
| 3.1.2 The DS1 Current Table .....                              | 33 |
| 3.1.3 The DS1 Interval Table .....                             | 36 |
| 3.1.4 The DS1 Total Table .....                                | 39 |
| 3.1.5 The DS1 Channel Table .....                              | 42 |
| 3.2 The DS1 Far End Group .....                                | 43 |
| 3.2.1 The DS1 Far End Current Table .....                      | 43 |
| 3.2.2 The DS1 Far End Interval Table .....                     | 47 |
| 3.2.3 The DS1 Far End Total Table .....                        | 50 |
| 3.3 The DS1 Fractional Table .....                             | 53 |
| 3.4 The DS1 Trap Group .....                                   | 55 |
| 3.5 Conformance Groups .....                                   | 61 |
| 4 Appendix A - Use of dsx1IfIndex and dsx1LineIndex .....      | 66 |
| 5 Appendix B - The delay approach to Unavailable Seconds. .... | 69 |
| 6 Intellectual Property .....                                  | 70 |
| 7 Acknowledgments .....  | 70 |
| 8 References .....   | 71 |
| 9 Security Considerations .....                                | 73 |
| 10 Author's Address .....                                      | 74 |
| 11 Full Copyright Statement .....                              | 75 |

## 1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2271 [1].
- o 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 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in RFC 1902 [5], RFC 1903 [6] and RFC 1904 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is

called SNMPv3 and described in RFC 1906 [10], RFC 2272 [11] and RFC 2274 [12].

- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2273 [14] and the view-based access control mechanism described in RFC 2275 [15]. 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.

### 1.1. Changes from RFC1406

The changes from RFC1406 are the following:

- (1) The Fractional Table has been deprecated.
- (2) This document uses SMIV2.
- (3) Usage is given for ifTable and ifXTable.
- (4) Example usage of ifStackTable is included.
- (5) dsx1IfIndex has been deprecated.
- (6) Support for DS2 and E2 have been added.
- (7) Additional lineTypes for DS2, E2, and unframed E1 were added.
- (8) The definition of valid intervals has been clarified for the case where the agent proxied for other devices. In particular, the treatment of missing intervals has been clarified.

- (9) An inward loopback has been added.
- (10) Additional lineStatus bits have been added for Near End in Unavailable Signal State, Carrier Equipment Out of Service, DS2 Payload AIS, and DS2 Performance Threshold.
- (11) A read-write line Length object has been added.
- (12) Signal mode of other has been added.
- (13) Added a lineStatus last change, trap and enabler.
- (14) The el(19) ifType has been obsoleted so this MIB does not list it as a supported ifType.
- (15) Textual Conventions for statistics objects have been used.
- (16) A new object, dsx1LoopbackStatus has been introduced to reflect the loopbacks established on a DS1 interface and the source to the requests. dsx1LoopbackConfig continues to be the desired loopback state while dsx1LoopbackStatus reflects the actual state.
- (17) A dual loopback has been added to allow the setting of an inward loopback and a line loopback at the same time.
- (18) An object indicating which channel to use within a parent object (i.e. DS3) has been added.
- (19) An object has been added to indicate whether or not this DS1/E1 is channelized.
- (20) Line coding type of B6ZS has been added for DS2

## 2. Overview

These objects are used when the particular media being used to realize an interface is a DS1/E1/DS2/E2 interface. At present, this applies to these values of the ifType variable in the Internet-standard MIB:

ds1 (18)

The definitions contained herein are based on the AT&T T-1 Superframe (a.k.a., D4) and Extended Superframe (ESF) formats [17, 18], the latter of which conforms to ANSI specifications [19], and the CCITT Recommendations [20, 21], referred to as E1 for the rest of this memo.

The various DS1 and E1 line disciplines are similar enough that separate MIBs are unwarranted, although there are some differences. For example, Loss of Frame is defined more rigorously in the ESF specification than in the D4 specification, but it is defined in both. Therefore, interface types el(19) and g703at2mb(67) have been obsoleted.

Where it is necessary to distinguish between the flavors of E1 with and without CRC, E1-CRC denotes the "with CRC" form (G.704 Table 4b) and E1-noCRC denotes the "without CRC" form (G.704 Table 4a).

## 2.1. Use of ifTable for DS1 Layer

Only the ifGeneralGroup needs to be supported.

| ifTable Object         | Use for DS1 Layer  |
|------------------------|--|
| ifIndex                | Interface index.   |
| ifDescr                | See interfaces MIB [16]  |
| ifType                 | ds1(18)  |
| ifSpeed                | Speed of line rate<br>DS1 - 1544000<br>E1 - 2048000<br>DS2 - 6312000<br>E2 - 8448000   |
| ifPhysAddress          | The value of the Circuit Identifier.<br>If no Circuit Identifier has been assigned this object should have an octet string with zero length. |
| ifAdminStatus          | See interfaces MIB [16]  |
| ifOperStatus           | See interfaces MIB [16]  |
| ifLastChange           | See interfaces MIB [16]  |
| ifName                 | See interfaces MIB [16].   |
| ifLinkUpDownTrapEnable | Set to enabled(1).   |
| ifHighSpeed            | Speed of line in Mega-bits per second<br>(2, 6, or 8)  |
| ifConnectorPresent     | Set to true(1) normally, except for  |

cases such as DS1/E1 over AAL1/ATM where false(2) is appropriate

## 2.2. Usage Guidelines

### 2.2.1. Usage of ifStackTable for Routers and DSUs

The object dsx1IfIndex has been deprecated. This object previously allowed a very special proxy situation to exist for Routers and CSUs. This section now describes how to use ifStackTable to represent this relationship.

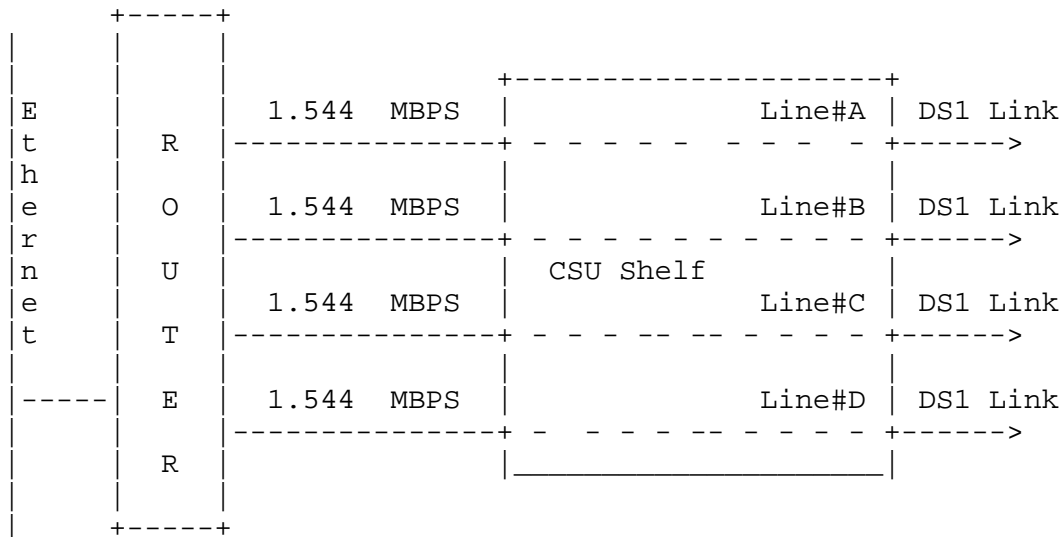
The paragraphs discussing dsx1IfIndex and dsx1LineIndex have been preserved in Appendix A for informational purposes.

The ifStackTable is used in the proxy case to represent the association between pairs of interfaces, e.g. this T1 is attached to that T1. This use is consistent with the use of the ifStackTable to show the association between various sub-layers of an interface. In both cases entire PDUs are exchanged between the interface pairs - in the case of a T1, entire T1 frames are exchanged; in the case of PPP and HDLC, entire HDLC frames are exchanged. This usage is not meant to suggest the use of the ifStackTable to represent Time Division Multiplexing (TDM) connections in general.

External&Internal interface scenario: the SNMP Agent resides on a host external from the device supporting DS1 interfaces (e.g., a router). The Agent represents both the host and the DS1 device.

#### Example:

A shelf full of CSUs connected to a Router. An SNMP Agent residing on the router proxies for itself and the CSU. The router has also an Ethernet interface:



The assignment of the index values could for example be:

| ifIndex | Description        |
|---------|--------------------|
| 1       | Ethernet           |
| 2       | Line#A Router      |
| 3       | Line#B Router      |
| 4       | Line#C Router      |
| 5       | Line#D Router      |
| 6       | Line#A CSU Router  |
| 7       | Line#B CSU Router  |
| 8       | Line#C CSU Router  |
| 9       | Line#D CSU Router  |
| 10      | Line#A CSU Network |
| 11      | Line#B CSU Network |
| 12      | Line#C CSU Network |
| 13      | Line#D CSU Network |

The ifStackTable is then used to show the relationships between the various DS1 interfaces.

| ifStackTable Entries |            |
|----------------------|------------|
| HigherLayer          | LowerLayer |
| 2                    | 6          |
| 3                    | 7          |
| 4                    | 8          |
| 5                    | 9          |
| 6                    | 10         |
| 7                    | 11         |
| 8                    | 12         |
| 9                    | 13         |

If the CSU shelf is managed by itself by a local SNMP Agent, the situation would be identical, except the Ethernet and the 4 router interfaces are deleted. Interfaces would also be numbered from 1 to 8.

| ifIndex | Description        |
|---------|--------------------|
| 1       | Line#A CSU Router  |
| 2       | Line#B CSU Router  |
| 3       | Line#C CSU Router  |
| 4       | Line#D CSU Router  |
| 5       | Line#A CSU Network |
| 6       | Line#B CSU Network |
| 7       | Line#C CSU Network |
| 8       | Line#D CSU Network |

#### ifStackTable Entries

| HigherLayer | LowerLayer |
|-------------|------------|
| 1           | 5          |
| 2           | 6          |
| 3           | 7          |
| 4           | 8          |

#### 2.2.2. Usage of ifStackTable for DS1/E1 on DS2/E2

An example is given of how DS1/E2 interfaces are stacked on DS2/E2 interfaces. It is not necessary nor is it always desirable to represent DS2 interfaces. If this is required, the following stacking should be used. All ifTypes are ds1. The DS2 is determined by examining ifSpeed or dsx1LineType.

| ifIndex | Description |
|---------|-------------|
| 1       | DS1 #1      |
| 2       | DS1 #2      |
| 3       | DS1 #3      |
| 4       | DS1 #4      |
| 5       | DS2         |

#### ifStackTable Entries

| HigherLayer | LowerLayer |
|-------------|------------|
| 1           | 5          |
| 2           | 5          |
| 3           | 5          |
| 4           | 5          |

### 2.2.3. Usage of Channelization for DS3, DS1, DS0

An example is given here to explain the channelization objects in the DS3, DS1, and DS0 MIBs to help the implementor use the objects correctly. Treatment of E3 and E1 would be similar, with the number of DS0s being different depending on the framing of the E1.

Assume that a DS3 (with ifIndex 1) is Channelized into DS1s (without DS2s). The object dsx3Channelization is set to enabledDs1. There will be 28 DS1s in the ifTable. Assume the entries in the ifTable for the DS1s are created in channel order and the ifIndex values are 2 through 29. In the DS1 MIB, there will be an entry in the dsx1ChanMappingTable for each ds1. The entries will be as follows:

#### dsx1ChanMappingTable Entries

| ifIndex | dsx1Ds1ChannelNumber | dsx1ChanMappedIfIndex |
|---------|----------------------|-----------------------|
| 1       | 1                    | 2                     |
| 1       | 2                    | 3                     |
| .....   |                      |                       |
| 1       | 28                   | 29                    |

In addition, the DS1s are channelized into DS0s. The object dsx1Channelization is set to enabledDS0 for each DS1. When this object is set to this value, 24 DS0s are created by the agent. There will be 24 DS0s in the ifTable for each DS1. If the dsx1Channelization is set to disabled, the 24 DS0s are destroyed.

Assume the entries in the ifTable are created in channel order and the ifIndex values for the DS0s in the first DS1 are 30 through 53. In the DS0 MIB, there will be an entry in the dsx0ChanMappingTable for each DS0. The entries will be as follows:

#### dsx0ChanMappingTable Entries

| ifIndex | dsx0Ds0ChannelNumber | dsx0ChanMappedIfIndex |
|---------|----------------------|-----------------------|
| 2       | 1                    | 30                    |
| 2       | 2                    | 31                    |
| .....   |                      |                       |
| 2       | 24                   | 53                    |

### 2.2.4. Usage of Channelization for DS3, DS2, DS1

An example is given here to explain the channelization objects in the DS3 and DS1 MIBs to help the implementor use the objects correctly.

Assume that a DS3 (with ifIndex 1) is Channelized into DS2s. The object dsx3Channelization is set to enabledDs2. There will be 7 DS2s (ifType of DS1) in the ifTable. Assume the entries in the ifTable for the DS2s are created in channel order and the ifIndex values are 2 through 8. In the DS1 MIB, there will be an entry in the dsx1ChanMappingTable for each DS2. The entries will be as follows:

dsx1ChanMappingTable Entries

| ifIndex | dsx1Ds1ChannelNumber | dsx1ChanMappedIfIndex |
|---------|----------------------|-----------------------|
| 1       | 1                    | 2                     |
| 1       | 2                    | 3                     |
| .....   |                      |                       |
| 1       | 7                    | 8                     |

In addition, the DS2s are channelized into DS1s. The object dsx1Channelization is set to enabledDS1 for each DS2. There will be 4 DS1s in the ifTable for each DS2. Assume the entries in the ifTable are created in channel order and the ifIndex values for the DS1s in the first DS2 are 9 through 12, then 13 through 16 for the second DS2, and so on. In the DS1 MIB, there will be an entry in the dsx1ChanMappingTable for each DS1. The entries will be as follows:

dsx1ChanMappingTable Entries

| ifIndex | dsx1Ds1ChannelNumber | dsx1ChanMappedIfIndex |
|---------|----------------------|-----------------------|
| 2       | 1                    | 9                     |
| 2       | 2                    | 10                    |
| 2       | 3                    | 11                    |
| 2       | 4                    | 12                    |
| 3       | 1                    | 13                    |
| 3       | 2                    | 14                    |
| ...     |                      |                       |
| 8       | 4                    | 36                    |

#### 2.2.5. Usage of Loopbacks

This section discusses the behaviour of objects related to loopbacks.

The object dsx1LoopbackConfig represents the desired state of loopbacks on this interface. Using this object a Manager can request:

```

LineLoopback
PayloadLoopback (if ESF framing)
InwardLoopback
DualLoopback (Line + Inward)
NoLoopback

```

The remote end can also request loopbacks either through the FDL channel if ESF or inband if D4. The loopbacks that can be request this way are:

- LineLoopback
- PayloadLoopback (if ESF framing)
- NoLoopback

To model the current state of loopbacks on a DS1 interface, the object `dsx1LoopbackStatus` defines which loopback is currently applies to an interface. This objects, which is a bitmap, will have bits turned on which reflect the currently active loopbacks on the interface as well as the source of those loopbacks.

The following restrictions/rules apply to loopbacks:

The far end cannot undo loopbacks set by a manager.

A manager can undo loopbacks set by the far end.

Both a line loopback and an inward loopback can be set at the same time. Only these two loopbacks can co-exist and either one may be set by the manager or the far end. A LineLoopback request from the far end is incremental to an existing Inward loopback established by a manager. When a NoLoopback is received from the far end in this case, the InwardLoopback remains in place.

### 2.3. Objectives of this MIB Module

There are numerous things that could be included in a MIB for DS1 signals: the management of multiplexors, CSUs, DSUs, and the like. The intent of this document is to facilitate the common management of all devices with DS1, E1, DS2, or E3 interfaces. As such, a design decision was made up front to very closely align the MIB with the set of objects that can generally be read from these types devices that are currently deployed.

J2 interfaces are not supported by this MIB.

### 2.4. DS1 Terminology

The terminology used in this document to describe error conditions on a DS1 interface as monitored by a DS1 device are based on the late but not final draft of what became the ANSI T1.231 standard [11]. If the definition in this document does not match the definition in the ANSI T1.231 document, the implementer should follow the definition described in this document.

### 2.4.1. Error Events

#### Bipolar Violation (BPV) Error Event

A BPV error event for an AMI-coded signal is the occurrence of a pulse of the same polarity as the previous pulse. (See T1.231 Section 6.1.1.1.1) A BPV error event for a B8ZS- or HDB3- coded signal is the occurrence of a pulse of the same polarity as the previous pulse without being a part of the zero substitution code.

#### Excessive Zeroes (EXZ) Error Event

An Excessive Zeroes error event for an AMI-coded signal is the occurrence of more than fifteen contiguous zeroes. (See T1.231 Section 6.1.1.1.2) For a B8ZS coded signal, the defect occurs when more than seven contiguous zeroes are detected.

#### Line Coding Violation (LCV) Error Event

A Line Coding Violation (LCV) is the occurrence of either a Bipolar Violation (BPV) or Excessive Zeroes (EXZ) Error Event. (Also known as CV-L; See T1.231 Section 6.5.1.1)

#### Path Coding Violation (PCV) Error Event

A Path Coding Violation error event is a frame synchronization bit error in the D4 and E1-noCRC formats, or a CRC or frame synch. bit error in the ESF and E1-CRC formats. (Also known as CV-P; See T1.231 Section 6.5.2.1)

#### Controlled Slip (CS) Error Event

A Controlled Slip is the replication or deletion of the payload bits of a DS1 frame. (See T1.231 Section 6.1.1.2.3) A Controlled Slip may be performed when there is a difference between the timing of a synchronous receiving terminal and the received signal. A Controlled Slip does not cause an Out of Frame defect.

### 2.4.2. Performance Defects

#### Out Of Frame (OOF) Defect

An OOF defect is the occurrence of a particular density of Framing Error events. (See T1.231 Section 6.1.2.2.1)

For DS1 links, an Out of Frame defect is declared when the receiver detects two or more framing errors within a 3 msec period for ESF signals and 0.75 msec for D4 signals, or two or more errors out of five or fewer consecutive framing-bits.

For E1 links, an Out Of Frame defect is declared when three consecutive frame alignment signals have been received with an error (see G.706 Section 4.1 [26]).

For DS2 links, an Out of Frame defect is declared when 7 or more consecutive errored framing patterns (4 multiframe) are received. The LOF is cleared when 3 or more consecutive correct framing patterns are received.

Once an Out Of Frame Defect is declared, the framer starts searching for a correct framing pattern. The Out of Frame defect ends when the signal is in frame.

In-frame occurs when there are fewer than two frame bit errors within 3 msec period for ESF signals and 0.75 msec for D4 signals.

For E1 links, in-frame occurs when a) in frame N the frame alignment signal is correct and b) in frame N+1 the frame alignment signal is absent (i.e., bit 2 in TS0 is a one) and c) in frame N+2 the frame alignment signal is present and correct. (See G.704 Section 4.1)

#### Alarm Indication Signal (AIS) Defect

For D4 and ESF links, the 'all ones' condition is detected at a DS1 line interface upon observing an unframed signal with a one's density of at least 99.9% present for a time equal to or greater than T, where  $3\text{ ms} \leq T \leq 75\text{ ms}$ . The AIS is terminated upon observing a signal not meeting the one's density or the unframed signal criteria for a period equal to or greater than T. (See G.775, Section 5.4)

For E1 links, the 'all-ones' condition is detected at the line interface as a string of 512 bits containing fewer than three zero bits (see O.162 [23] Section 3.3.2).

For DS2 links, the DS2 AIS shall be sent from the NT1 to the user to indicate a loss of the 6,312 kbps frame capability on the network side. The DS2 AIS is defined as a bit array of 6,312 kbps in which all binary bits are set to '1'.

The DS2 AIS detection and removal shall be implemented according to ITU-T Draft Recommendation G.775 [31] Section 5.5:

- a DS2 AIS defect is detected when the incoming signal has two (2) or less ZEROs in a sequence of 3156 bits (0.5 ms).
- a DS2 AIS defect is cleared when the incoming signal has three (3) or more ZEROs in a sequence of 3156 bits (0.5 ms).

### 2.4.3. Performance Parameters

All performance parameters are accumulated in fifteen minute intervals and up to 96 intervals (24 hours worth) are kept by an agent. Fewer than 96 intervals of data will be available if the agent has been restarted within the last 24 hours. In addition, there is a rolling 24-hour total of each performance parameter. Performance parameters continue to be collected when the interface is down.

There is no requirement for an agent to ensure fixed relationship between the start of a fifteen minute interval and any wall clock; however some agents may align the fifteen minute intervals with quarter hours.

Performance parameters are of types PerfCurrentCount, PerfIntervalCount and PerfTotalCount. These textual conventions are all Gauge32, and they are used because it is possible for these objects to decrease. Objects may decrease when Unavailable Seconds occurs across a fifteen minutes interval boundary. See Unavailable Seconds discussion later in this section.

#### Line Errored Seconds (LES)

A Line Errored Second is a second in which one or more Line Code Violation error events were detected. (Also known as ES-L; See T1.231 Section 6.5.1.2)

#### Controlled Slip Seconds (CSS)

A Controlled Slip Second is a one-second interval containing one or more controlled slips. (See T1.231 Section 6.5.2.8) This is not incremented during an Unavailable Second.

#### Errored Seconds (ES)

For ESF and E1-CRC links an Errored Second is a second with one or more Path Code Violation OR one or more Out of Frame defects OR one or more Controlled Slip events OR a detected AIS defect. (See T1.231 Section 6.5.2.2 and G.826 [32] Section B.1)

For D4 and E1-noCRC links, the presence of Bipolar Violations also triggers an Errored Second.

This is not incremented during an Unavailable Second.

#### Bursty Errored Seconds (BES)

A Bursty Errored Second (also known as Errored Second type B in T1.231 Section 6.5.2.4) is a second with fewer than 320 and more than 1 Path Coding Violation error events, no Severely Errored Frame defects and no detected incoming AIS defects. Controlled slips are not included in this parameter.

This is not incremented during an Unavailable Second. It applies to ESF signals only.

#### Severely Errored Seconds (SES)

A Severely Errored Second for ESF signals is a second with 320 or more Path Code Violation Error Events OR one or more Out of Frame defects OR a detected AIS defect. (See T1.231 Section 6.5.2.5)

For E1-CRC signals, a Severely Errored Second is a second with 832 or more Path Code Violation error events OR one or more Out of Frame defects.

For E1-noCRC signals, a Severely Errored Second is a 2048 LCVs or more.

For D4 signals, a Severely Errored Second is a count of one-second intervals with Framing Error events, or an OOF defect, or 1544 LCVs or more.

Controlled slips are not included in this parameter.

This is not incremented during an Unavailable Second.

#### Severely Errored Framing Second (SEFS)

An Severely Errored Framing Second is a second with one or more Out of Frame defects OR a detected AIS defect. (Also known as SAS-P (SEF/AIS second); See T1.231 Section 6.5.2.6)

#### Degraded Minutes

A Degraded Minute is one in which the estimated error rate exceeds 1E-6 but does not exceed 1E-3 (see G.821 [24]).

Degraded Minutes are determined by collecting all of the Available Seconds, removing any Severely Errored Seconds grouping the result in 60-second long groups and counting a 60-second long group (a.k.a., minute) as degraded if the cumulative errors during the seconds present in the group exceed 1E-6. Available seconds are merely those seconds which are not Unavailable as described below.

#### Unavailable Seconds (UAS)

Unavailable Seconds (UAS) are calculated by counting the number of seconds that the interface is unavailable. The DS1 interface is said to be unavailable from the onset of 10 contiguous SESSs, or the onset of the condition leading to a failure (see Failure States). If the condition leading to the failure was immediately preceded by one or more contiguous SESSs, then the DS1 interface unavailability starts from the onset of these SESSs. Once unavailable, and if no failure is present, the DS1 interface becomes available at the onset of 10 contiguous seconds with no SESSs. Once unavailable, and if a failure is present, the DS1 interface becomes available at the onset of 10 contiguous seconds with no SESSs, if the failure clearing time is less than or equal to 10 seconds. If the failure clearing time is more than 10 seconds, the DS1 interface becomes available at the onset of 10 contiguous seconds with no SESSs, or the onset period leading to the successful clearing condition, whichever occurs later. With respect to the DS1 error counts, all counters are incremented while the DS1 interface is deemed available. While the interface is deemed unavailable, the only count that is incremented is UASs.

Note that this definition implies that the agent cannot determine until after a ten second interval has passed whether a given one-second interval belongs to available or unavailable time. If the agent chooses to update the various performance statistics in real time then it must be prepared to retroactively reduce the ES, BES, SES, and SEFS counts by 10 and increase the UAS count by 10 when it determines that available time has been entered. It must also be prepared to adjust the PCV count and the DM count as necessary since these parameters are not accumulated during unavailable time. It must be similarly prepared to retroactively decrease the UAS count by 10 and increase the ES, BES, and DM counts as necessary upon entering available time. A special case exists when the 10 second period leading to available or unavailable time crosses a 900 second statistics window boundary, as the foregoing description implies that the ES, BES, SES, SEFS, DM, and UAS counts the PREVIOUS interval must be adjusted. In this case successive GETs of the affected dsx1IntervalSESS and dsx1IntervalUASs objects will return differing values if the first GET occurs during the first few seconds of the window.

The agent may instead choose to delay updates to the various statistics by 10 seconds in order to avoid retroactive adjustments to the counters. A way to do this is sketched in Appendix B.

In any case, a linkDown trap shall be sent only after the agent has determined for certain that the unavailable state has been entered, but the time on the trap will be that of the first UAS (i.e., 10 seconds earlier). A linkUp trap shall be handled similarly.

According to ANSI T1.231 unavailable time begins at the `_onset_` of 10 contiguous severely errored seconds -- that is, unavailable time starts with the `_first_` of the 10 contiguous SESSs. Also, while an interface is deemed unavailable all counters for that interface are frozen except for the UAS count. It follows that an implementation which strictly complies with this standard must `_not_` increment any counters other than the UAS count -- even temporarily -- as a result of anything that happens during those 10 seconds. Since changes in the signal state lag the data to which they apply by 10 seconds, an ANSI-compliant implementation must pass the the one-second statistics through a 10-second delay line prior to updating any counters. That can be done by performing the following steps at the end of each one second interval.

- i) Read near/far end CV counter and alarm status flags from the hardware.
- ii) Accumulate the CV counts for the preceding second and compare them to the ES and SES threshold for the layer in question. Update the signal state and shift the one-second CV counts and ES/SES flags into the 10-element delay line. Note that far-end one-second statistics are to be flagged as "absent" during any second in which there is an incoming defect at the layer in question or at any lower layer.
- iii) Update the current interval statistics using the signal state from the `_previous_` update cycle and the one-second CV counts and ES/SES flags shifted out of the 10-element delay line.

This approach is further described in Appendix B.

#### 2.4.4. Failure States

The following failure states are received, or detected failures, that are reported in the `dsxlLineStatus` object. When a DS1 interface would, if ever, produce the conditions leading to the failure state is described in the appropriate specification.

#### Far End Alarm Failure

The Far End Alarm failure is also known as "Yellow Alarm" in the DS1 case, "Distant Alarm" in the E1 case, and "Remote Alarm" in the DS2 case.

For D4 links, the Far End Alarm failure is declared when bit 6 of all channels has been zero for at least 335 ms and is cleared when bit 6 of at least one channel is non-zero for a period T, where T is usually less than one second and always less than 5 seconds. The Far End Alarm failure is not declared for D4 links when a Loss of Signal is detected.

For ESF links, the Far End Alarm failure is declared if the Yellow Alarm signal pattern occurs in at least seven out of ten contiguous 16-bit pattern intervals and is cleared if the Yellow Alarm signal pattern does not occur in ten contiguous 16-bit signal pattern intervals.

For E1 links, the Far End Alarm failure is declared when bit 3 of time-slot zero is received set to one on two consecutive occasions. The Far End Alarm failure is cleared when bit 3 of time-slot zero is received set to zero.

For DS2 links, if a loss of frame alignment (LOF or LOS) and/or DS2 AIS condition, is detected, the RAI signal shall be generated and transmitted to the remote side.

The Remote Alarm Indication(RAI) signal is defined on m-bits as a repetition of the 16bit sequence consisting of eight binary '1s' and eight binary '0s' in m-bits(1111111100000000). When the RAI signal is not sent (in normal operation),the HDLC flag pattern (01111110) in the m-bit is sent.

The RAI failure is detected when 16 or more consecutive RAI-patterns (1111111100000000) are received. The RAI failure is cleared when 4 or more consecutive incorrect-RAI-patterns are received.

#### Alarm Indication Signal (AIS) Failure

The Alarm Indication Signal failure is declared when an AIS defect is detected at the input and the AIS defect still exists after the Loss Of Frame failure (which is caused by the unframed nature of the 'all-ones' signal) is declared. The AIS failure is cleared when the Loss Of Frame failure is cleared. (See T1.231 Section 6.2.1.2.1)

An AIS defect at a 6312 kbit/s (G.704) interface is detected when the incoming signal has two {2} or less ZEROs in a sequence of 3156 bits (0.5ms).

The AIS signal defect is cleared when the incoming signal has three {3} or more ZEROs in a sequence of 3156 bits (0.5ms).

#### Loss Of Frame Failure

For DS1 links, the Loss Of Frame failure is declared when an OOF or LOS defect has persisted for T seconds, where  $2 \leq T \leq 10$ . The Loss Of Frame failure is cleared when there have been no OOF or LOS defects during a period T where  $0 \leq T \leq 20$ . Many systems will perform "hit integration" within the period T before declaring or clearing the failure e.g., see TR 62411 [25].

For E1 links, the Loss Of Frame Failure is declared when an OOF defect is detected.

#### Loss Of Signal Failure

For DS1, the Loss Of Signal failure is declared upon observing 175 +/- 75 contiguous pulse positions with no pulses of either positive or negative polarity. The LOS failure is cleared upon observing an average pulse density of at least 12.5% over a period of 175 +/- 75 contiguous pulse positions starting with the receipt of a pulse.

For E1 links, the Loss Of Signal failure is declared when greater than 10 consecutive zeroes are detected (see 0.162 Section 3.4'<.4).

A LOS defect at 6312kbit/s interfaces is detected when the incoming signal has "no transitions", i.e. when the signal level is less than or equal to a signal level of 35dB below nominal, for N consecutive pulse intervals, where  $10 \leq N \leq 255$ .

The LOS defect is cleared when the incoming signal has "transitions", i.e. when the signal level is greater than or equal to a signal level of 9dB below nominal, for N consecutive pulse intervals, where  $10 \leq N \leq 255$ .

A signal with "transitions" corresponds to a G.703 compliant signal.

#### Loopback Pseudo-Failure

The Loopback Pseudo-Failure is declared when the near end equipment has placed a loopback (of any kind) on the DS1. This allows a management entity to determine from one object whether the DS1 can be considered to be in service or not (from the point of view of the near end equipment).

#### TS16 Alarm Indication Signal Failure

For E1 links, the TS16 Alarm Indication Signal failure is declared when time-slot 16 is received as all ones for all frames of two consecutive multiframes (see G.732 Section 4.2.6). This condition is never declared for DS1.

#### Loss Of MultiFrame Failure

The Loss Of MultiFrame failure is declared when two consecutive multiframe alignment signals (bits 4 through 7 of TS16 of frame 0) have been received with an error. The Loss Of Multiframe failure is cleared when the first correct multiframe alignment signal is received. The Loss Of Multiframe failure can only be declared for E1 links operating with G.732 [27] framing (sometimes called "Channel Associated Signalling" mode).

#### Far End Loss Of Multiframe Failure

The Far End Loss Of Multiframe failure is declared when bit 2 of TS16 of frame 0 is received set to one on two consecutive occasions. The Far End Loss Of Multiframe failure is cleared when bit 2 of TS16 of frame 0 is received set to zero. The Far End Loss Of Multiframe failure can only be declared for E1 links operating in "Channel Associated Signalling" mode. (See G.732)

#### DS2 Payload AIS Failure

The DS2 Payload AIS is detected when the incoming signal of the 6,312 kbps frame payload [TS1-TS96] has 2 or less 0's in a sequence of 3072 bits (0.5ms). The DS2 Payload AIS is cleared when the incoming signal of the 6,312 kbps frame payload [TS1-TS96] has 3 or more 0's in a sequence of 3072 bits (0.5 ms).

#### DS2 Performance Threshold

DS2 Performance Threshold Failure monitors equipment performance and is based on the CRC (Cyclic Redundancy Check) Procedure defined in G.704.

The DS2 Performance Threshold Failure is detected when the bit error ratio exceeds  $10^{-4}$  (Performance Threshold), and the DS2 Performance Threshold Failure shall be cleared when the bit error ratio decreased to less than  $10^{-6}$ .

### 2.4.5. Other Terms

#### Circuit Identifier

This is a character string specified by the circuit vendor, and is useful when communicating with the vendor during the troubleshooting process.

#### Proxy

In this document, the word proxy is meant to indicate an application which receives SNMP messages and replies to them on behalf of the devices which implement the actual DS3/E3 interfaces. The proxy may have already collected the information about the DS3/E3 interfaces into its local database and may not necessarily forward the requests to the actual DS3/E3 interface. It is expected in such an application that there are periods of time where the proxy is not communicating with the DS3/E3 interfaces. In these instances the proxy will not necessarily have up-to-date configuration information and will most likely have missed the collection of some statistics data. Missed statistics data collection will result in invalid data in the interval table.

## 3. Object Definitions

DS1-MIB DEFINITIONS ::= BEGIN

#### IMPORTS

|                                      |                       |
|--------------------------------------|-----------------------|
| MODULE-IDENTITY, OBJECT-TYPE,        |                       |
| NOTIFICATION-TYPE, transmission      | FROM SNMPv2-SMI       |
| DisplayString, TimeStamp, TruthValue | FROM SNMPv2-TC        |
| MODULE-COMPLIANCE, OBJECT-GROUP,     |                       |
| NOTIFICATION-GROUP                   | FROM SNMPv2-CONF      |
| InterfaceIndex, ifIndex              | FROM IF-MIB           |
| PerfCurrentCount, PerfIntervalCount, |                       |
| PerfTotalCount                       | FROM PerfHist-TC-MIB; |

#### ds1 MODULE-IDENTITY

LAST-UPDATED "9808011830Z"

ORGANIZATION "IETF Trunk MIB Working Group"

#### CONTACT-INFO

" David Fowler

Postal: Newbridge Networks Corporation  
600 March Road  
Kanata, Ontario, Canada K2K 2E6

Tel: +1 613 591 3600

Fax: +1 613 599 3667

E-mail: davef@newbridge.com"

#### DESCRIPTION

"The MIB module to describe DS1, E1, DS2, and E2 interfaces objects."

::= { transmission 18 }

-- note that this subsumes cept (19) and g703at2mb (67)  
-- there is no separate CEPT or G703AT2MB MIB

-- The DS1 Near End Group

-- The DS1 Near End Group consists of five tables:

-- DS1 Configuration  
-- DS1 Current  
-- DS1 Interval  
-- DS1 Total  
-- DS1 Channel Table

-- The DS1 Configuration Table

dsx1ConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx1ConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The DS1 Configuration table."

::= { ds1 6 }

dsx1ConfigEntry OBJECT-TYPE

SYNTAX Dsx1ConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the DS1 Configuration table."

INDEX { dsx1LineIndex }

::= { dsx1ConfigTable 1 }

Dsx1ConfigEntry ::=

SEQUENCE {

dsx1LineIndex

dsx1IfIndex

dsx1TimeElapsed

dsx1ValidIntervals

dsx1LineType

dsx1LineCoding

InterfaceIndex,

InterfaceIndex,

INTEGER,

INTEGER,

INTEGER,

INTEGER,

```

        dsx1SendCode                INTEGER,
        dsx1CircuitIdentifier        DisplayString,
        dsx1LoopbackConfig           INTEGER,
        dsx1LineStatus               INTEGER,
        dsx1SignalMode               INTEGER,
        dsx1TransmitClockSource       INTEGER,
        dsx1Fd1                      INTEGER,
        dsx1InvalidIntervals          INTEGER,
        dsx1LineLength               INTEGER,
        dsx1LineStatusLastChange      TimeStamp,
        dsx1LineStatusChangeTrapEnable INTEGER,
        dsx1LoopbackStatus            INTEGER,
        dsx1Ds1ChannelNumber          INTEGER,
        dsx1Channelization            INTEGER
    }

dsx1LineIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object should be made equal to ifIndex.  The
        next paragraph describes its previous usage.
        Making the object equal to ifIndex allows proper
        use of ifStackTable and ds0/ds0bundle mibs.

        Previously, this object is the identifier of a DS1
        Interface on a managed device.  If there is an
        ifEntry that is directly associated with this and
        only this DS1 interface, it should have the same
        value as ifIndex.  Otherwise, number the
        dsx1LineIndices with an unique identifier
        following the rules of choosing a number that is
        greater than ifNumber and numbering the inside
        interfaces (e.g., equipment side) with even
        numbers and outside interfaces (e.g, network side)
        with odd numbers."
    ::= { dsx1ConfigEntry 1 }

dsx1IfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This value for this object is equal to the value
        of ifIndex from the Interfaces table of MIB II
        (RFC 1213)."
    ::= { dsx1ConfigEntry 2 }

```

## dsx1TimeElapsed 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 near end current error-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."

::= { dsx1ConfigEntry 3 }

## dsx1ValidIntervals OBJECT-TYPE

SYNTAX INTEGER (0..96)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of previous near end intervals for which data was collected. The value will be 96 unless the interface was brought online within the last 24 hours, in which case the value will be the number of complete 15 minute near end intervals since the interface has been online. 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."

::= { dsx1ConfigEntry 4 }

## dsx1LineType OBJECT-TYPE

SYNTAX INTEGER {  
    other(1),  
    dsx1ESF(2),  
    dsx1D4(3),  
    dsx1E1(4),  
    dsx1E1CRC(5),  
    dsx1E1MF(6),  
    dsx1E1CRCMF(7),  
    dsx1Unframed(8),  
    dsx1E1Unframed(9),  
    dsx1DS2M12(10),  
    dsx2E2(11)  
}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This variable indicates the variety of DS1 Line implementing this circuit. The type of circuit affects the number of bits per second that the circuit can reasonably carry, as well as the interpretation of the usage and error statistics. The values, in sequence, describe:

| TITLE:         | SPECIFICATION:                                   |
|----------------|--|
| dsx1ESF        | Extended SuperFrame DS1 (T1.107)                 |
| dsx1D4         | AT&T D4 format DS1 (T1.107)                      |
| dsx1E1         | ITU-T Recommendation G.704<br>(Table 4a)         |
| dsx1E1-CRC     | ITU-T Recommendation G.704<br>(Table 4b)         |
| dsxE1-MF       | G.704 (Table 4a) with TS16<br>multiframe enabled |
| dsx1E1-CRC-MF  | G.704 (Table 4b) with TS16<br>multiframe enabled |
| dsx1Unframed   | DS1 with No Framing                              |
| dsx1E1Unframed | E1 with No Framing (G.703)                       |
| dsx1DS2M12     | DS2 frame format (T1.107)                        |
| dsx1E2         | E2 frame format (G.704)                          |

For clarification, the capacity for each E1 type is as listed below:

dsx1E1Unframed - E1, no framing = 32 x 64k = 2048k  
 dsx1E1 or dsx1E1CRC - E1, with framing,  
     no signalling = 31 x 64k = 1984k  
 dsx1E1MF or dsx1E1CRCMF - E1, with framing,  
     signalling = 30 x 64k = 1920k

For further information See ITU-T Recomm G.704"  
 ::= { dsx1ConfigEntry 5 }

dsx1LineCoding OBJECT-TYPE

```
SYNTAX  INTEGER {
    dsx1JBZS (1),
    dsx1B8ZS (2),
    dsx1HDB3 (3),
    dsx1ZBTSI (4),
    dsx1AMI  (5),
    other(6),
    dsx1B6ZS(7)
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This variable describes the variety of Zero Code

Suppression used on this interface, which in turn affects a number of its characteristics.

dsx1JBZS refers the Jammed Bit Zero Suppression, in which the AT&T specification of at least one pulse every 8 bit periods is literally implemented by forcing a pulse in bit 8 of each channel. Thus, only seven bits per channel, or 1.344 Mbps, is available for data.

dsx1B8ZS refers to the use of a specified pattern of normal bits and bipolar violations which are used to replace a sequence of eight zero bits.

ANSI Clear Channels may use dsx1ZBTISI, or Zero Byte Time Slot Interchange.

E1 links, with or without CRC, use dsx1HDB3 or dsx1AMI.

dsx1AMI refers to a mode wherein no zero code suppression is present and the line encoding does not solve the problem directly. In this application, the higher layer must provide data which meets or exceeds the pulse density requirements, such as inverting HDLC data.

dsx1B6ZS refers to the user of a specified pattern of normal bits and bipolar violations which are used to replace a sequence of six zero bits. Used for DS2."

```
::= { dsx1ConfigEntry 6 }
```

```
dsx1SendCode OBJECT-TYPE
```

```
SYNTAX INTEGER {
    dsx1SendNoCode(1),
    dsx1SendLineCode(2),
    dsx1SendPayloadCode(3),
    dsx1SendResetCode(4),
    dsx1SendQRS(5),
    dsx1Send511Pattern(6),
    dsx1Send3in24Pattern(7),
    dsx1SendOtherTestPattern(8)
}
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"This variable indicates what type of code is being sent across the DS1 interface by the device. Setting this variable causes the interface to send the code requested. The values mean:

dsx1SendNoCode

sending looped or normal data

dsx1SendLineCode

sending a request for a line loopback

dsx1SendPayloadCode

sending a request for a payload loopback

dsx1SendResetCode

sending a loopback termination request

dsx1SendQRS

sending a Quasi-Random Signal (QRS) test pattern

dsx1Send511Pattern

sending a 511 bit fixed test pattern

dsx1Send3in24Pattern

sending a fixed test pattern of 3 bits set in 24

dsx1SendOtherTestPattern

sending a test pattern other than those described by this object"

::= { dsx1ConfigEntry 7 }

dsx1CircuitIdentifier OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This variable contains the transmission vendor's circuit identifier, for the purpose of facilitating troubleshooting."

::= { dsx1ConfigEntry 8 }

dsx1LoopbackConfig OBJECT-TYPE

SYNTAX INTEGER {  
     dsx1NoLoop(1),  
     dsx1PayloadLoop(2),  
     dsx1LineLoop(3),  
     dsx1OtherLoop(4),

```

        dsx1InwardLoop(5),
        dsx1DualLoop(6)
    }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "This variable represents the desired loopback
    configuration of the DS1 interface.  Agents
    supporting read/write access should return
    inconsistentValue in response to a requested
    loopback state that the interface does not
    support.  The values mean:

    dsx1NoLoop
        Not in the loopback state.  A device that is not
        capable of performing a loopback on the interface
        shall always return this as its value.

    dsx1PayloadLoop
        The received signal at this interface is looped
        through the device.  Typically the received signal
        is looped back for retransmission after it has
        passed through the device's framing function.

    dsx1LineLoop
        The received signal at this interface does not go
        through the device (minimum penetration) but is
        looped back out.

    dsx1OtherLoop
        Loopbacks that are not defined here.

    dsx1InwardLoop
        The transmitted signal at this interface is
        looped back and received by the same interface.
        What is transmitted onto the line is product
        dependent.

    dsx1DualLoop
        Both dsx1LineLoop and dsx1InwardLoop will be
        active simultaneously."
 ::= { dsx1ConfigEntry 9 }

dsx1LineStatus OBJECT-TYPE
    SYNTAX  INTEGER (1..131071)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

"This variable indicates the Line Status of the interface. It contains loopback, failure, received 'alarm' and transmitted 'alarms' information.

The dsxlLineStatus is a bit map represented as a sum, therefore, it can represent multiple failures (alarms) and a LoopbackState simultaneously.

dsxlNoAlarm must be set if and only if no other flag is set.

If the dsxlloopbackState bit is set, the loopback in effect can be determined from the dsxlloopbackConfig object.

The various bit positions are:

|       |                      |                                      |
|-------|----------------------|--------------------------------------|
| 1     | dsxlNoAlarm          | No alarm present                     |
| 2     | dsxlRcvFarEndLOF     | Far end LOF (a.k.a., Yellow Alarm)   |
| 4     | dsxlXmtFarEndLOF     | Near end sending LOF Indication      |
| 8     | dsxlRcvAIS           | Far end sending AIS                  |
| 16    | dsxlXmtAIS           | Near end sending AIS                 |
| 32    | dsxlLossOfFrame      | Near end LOF (a.k.a., Red Alarm)     |
| 64    | dsxlLossOfSignal     | Near end Loss Of Signal              |
| 128   | dsxlLoopbackState    | Near end is looped                   |
| 256   | dsxlT16AIS           | E1 TS16 AIS                          |
| 512   | dsxlRcvFarEndLOMF    | Far End Sending TS16 LOMF            |
| 1024  | dsxlXmtFarEndLOMF    | Near End Sending TS16 LOMF           |
| 2048  | dsxlRcvTestCode      | Near End detects a test code         |
| 4096  | dsxlOtherFailure     | any line status not defined here     |
| 8192  | dsxlUnavailSigState  | Near End in Unavailable Signal State |
| 16384 | dsxlNetEquipOOS      | Carrier Equipment Out of Service     |
| 32768 | dsxlRcvPayloadAIS    | DS2 Payload AIS                      |
| 65536 | dsxlDs2PerfThreshold | DS2 Performance Threshold Exceeded"  |

```
::= { dsxlConfigEntry 10 }
```

dsxlSignalMode OBJECT-TYPE

```
SYNTAX  INTEGER {
    none (1),
    robbedBit (2),
    bitOriented (3),
    messageOriented (4),
    other (5)
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"'none' indicates that no bits are reserved for signaling on this channel.

'robbedBit' indicates that DS1 Robbed Bit Signaling is in use.

'bitOriented' indicates that E1 Channel Associated Signaling is in use.

'messageOriented' indicates that Common Channel Signaling is in use either on channel 16 of an E1 link or channel 24 of a DS1."

::= { dsx1ConfigEntry 11 }

dsx1TransmitClockSource OBJECT-TYPE

```
SYNTAX  INTEGER {
    loopTiming(1),
    localTiming(2),
    throughTiming(3)
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The source of Transmit Clock.

'loopTiming' indicates that the recovered receive clock is used as the transmit clock.

'localTiming' indicates that a local clock source is used or when an external clock is attached to the box containing the interface.

'throughTiming' indicates that recovered receive clock from another interface is used as the transmit clock."

::= { dsx1ConfigEntry 12 }

dsx1Fd1 OBJECT-TYPE

```
SYNTAX  INTEGER (1..15)
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This bitmap describes the use of the facilities data link, and is the sum of the capabilities. Set any bits that are appropriate:

```
other(1),
dsx1AnsiT1403(2),
dsx1Att54016(4),
```

dsx1FdlNone(8)

'other' indicates that a protocol other than one following is used.

'dsx1AnsiT1403' refers to the FDL exchange recommended by ANSI.

'dsx1Att54016' refers to ESF FDL exchanges.

'dsx1FdlNone' indicates that the device does not use the FDL."

::= { dsx1ConfigEntry 13 }

dsx1InvalidIntervals OBJECT-TYPE

SYNTAX INTEGER (0..96)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of intervals in the range from 0 to dsx1ValidIntervals 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)."

::= { dsx1ConfigEntry 14 }

dsx1LineLength OBJECT-TYPE

SYNTAX INTEGER (0..64000)

UNITS "meters"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The length of the ds1 line in meters. This object provides information for line build out circuitry. This object is only useful if the interface has configurable line build out circuitry."

::= { dsx1ConfigEntry 15 }

dsx1LineStatusLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of MIB II's sysUpTime object at the time this DS1 entered its current line status state. If the current state was entered prior to

the last re-initialization of the proxy-agent,  
then this object contains a zero value."  
 ::= { dsx1ConfigEntry 16 }

dsx1LineStatusChangeTrapEnable OBJECT-TYPE  
SYNTAX INTEGER {  
enabled(1),  
disabled(2)  
}  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"Indicates whether dsx1LineStatusChange traps  
should be generated for this interface."  
DEFVAL { disabled }  
 ::= { dsx1ConfigEntry 17 }

dsx1LoopbackStatus OBJECT-TYPE  
SYNTAX INTEGER (1..127)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"This variable represents the current state of the  
loopback on the DS1 interface. It contains  
information about loopbacks established by a  
manager and remotely from the far end.

The dsx1LoopbackStatus is a bit map represented as  
a sum, therefore it can represent multiple  
loopbacks simultaneously.

The various bit positions are:

- 1 dsx1NoLoopback
- 2 dsx1NearEndPayloadLoopback
- 4 dsx1NearEndLineLoopback
- 8 dsx1NearEndOtherLoopback
- 16 dsx1NearEndInwardLoopback
- 32 dsx1FarEndPayloadLoopback
- 64 dsx1FarEndLineLoopback"

::= { dsx1ConfigEntry 18 }

dsx1Ds1ChannelNumber OBJECT-TYPE  
SYNTAX INTEGER (0..28)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"This variable represents the channel number of

the DS1/E1 on its parent Ds2/E2 or DS3/E3. A value of 0 indicated this DS1/E1 does not have a parent DS3/E3."

```
::= { dsx1ConfigEntry 19 }
```

```
dsx1Channelization OBJECT-TYPE
```

```
SYNTAX      INTEGER {
                disabled(1),
                enabledDs0(2),
                enabledDs1(3)
            }
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

```
DESCRIPTION
```

"Indicates whether this ds1/e1 is channelized or unchannelized. The value of enabledDs0 indicates that this is a DS1 channelized into DS0s. The value of enabledDs1 indicated that this is a DS2 channelized into DS1s. Setting this value will cause the creation or deletion of entries in the ifTable for the DS0s that are within the DS1."

```
::= { dsx1ConfigEntry 20 }
```

```
-- The DS1 Current Table
```

```
dsx1CurrentTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF Dsx1CurrentEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

"The DS1 current table contains various statistics being collected for the current 15 minute interval."

```
::= { ds1 7 }
```

```
dsx1CurrentEntry OBJECT-TYPE
```

```
SYNTAX      Dsx1CurrentEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

"An entry in the DS1 Current table."

```
INDEX      { dsx1CurrentIndex }
```

```
::= { dsx1CurrentTable 1 }
```

```
Dsx1CurrentEntry ::=
```

```
SEQUENCE {
```

```
    dsx1CurrentIndex
```

```
    dsx1CurrentESS
```

```
    InterfaceIndex,
```

```
    PerfCurrentCount,
```

```

        dsx1CurrentSESS          PerfCurrentCount,
        dsx1CurrentSEFSS        PerfCurrentCount,
        dsx1CurrentUASS         PerfCurrentCount,
        dsx1CurrentCSSS         PerfCurrentCount,
        dsx1CurrentPCVs         PerfCurrentCount,
        dsx1CurrentLESS         PerfCurrentCount,
        dsx1CurrentBESS         PerfCurrentCount,
        dsx1CurrentDMS          PerfCurrentCount,
        dsx1CurrentLCVs         PerfCurrentCount
    }

dsx1CurrentIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the
        DS1 interface to which this entry is applicable.
        The interface identified by a particular value of
        this index is the same interface as identified by
        the same value as a dsx1LineIndex object
        instance."
    ::= { dsx1CurrentEntry 1 }

dsx1CurrentESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Errored Seconds."
    ::= { dsx1CurrentEntry 2 }

dsx1CurrentSESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Seconds."
    ::= { dsx1CurrentEntry 3 }

dsx1CurrentSEFSSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Framing Seconds."
    ::= { dsx1CurrentEntry 4 }

```

```
dsx1CurrentUASSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds."
    ::= { dsx1CurrentEntry 5 }

dsx1CurrentCSSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Controlled Slip Seconds."
    ::= { dsx1CurrentEntry 6 }

dsx1CurrentPCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Path Coding Violations."
    ::= { dsx1CurrentEntry 7 }

dsx1CurrentLESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Errored Seconds."
    ::= { dsx1CurrentEntry 8 }

dsx1CurrentBESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Bursty Errored Seconds."
    ::= { dsx1CurrentEntry 9 }

dsx1CurrentDMs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Degraded Minutes."
    ::= { dsx1CurrentEntry 10 }
```

```

dsx1CurrentLCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Code Violations (LCVs)."
```

::= { dsx1CurrentEntry 11 }

-- The DS1 Interval Table

```

dsx1IntervalTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx1IntervalEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "The DS1 Interval Table contains various
        statistics collected by each DS1 Interface over
        the previous 24 hours of operation. The past 24
        hours are broken into 96 completed 15 minute
        intervals. Each row in this table represents one
        such interval (identified by dsx1IntervalNumber)
        for one specific instance (identified by
        dsx1IntervalIndex)."
```

::= { ds1 8 }

```

dsx1IntervalEntry OBJECT-TYPE
    SYNTAX  Dsx1IntervalEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "An entry in the DS1 Interval table."
```

INDEX { dsx1IntervalIndex, dsx1IntervalNumber }

::= { dsx1IntervalTable 1 }

```

Dsx1IntervalEntry ::=
    SEQUENCE {
        dsx1IntervalIndex          InterfaceIndex,
        dsx1IntervalNumber          INTEGER,
        dsx1IntervalESS             PerfIntervalCount,
        dsx1IntervalSESS            PerfIntervalCount,
        dsx1IntervalSEFSS           PerfIntervalCount,
        dsx1IntervalUASS            PerfIntervalCount,
        dsx1IntervalCSSS            PerfIntervalCount,
        dsx1IntervalPCVs            PerfIntervalCount,
        dsx1IntervalLESS            PerfIntervalCount,
        dsx1IntervalBESS            PerfIntervalCount,
        dsx1IntervalDMS             PerfIntervalCount,
        dsx1IntervalLCVs            PerfIntervalCount,
```

```

        dsx1IntervalValidData      TruthValue
    }

dsx1IntervalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS1
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is the same interface as identified by the
        same value as a dsx1LineIndex object instance."
    ::= { dsx1IntervalEntry 1 }

dsx1IntervalNumber OBJECT-TYPE
    SYNTAX  INTEGER (1..96)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "A number between 1 and 96, where 1 is the most
        recently completed 15 minute interval and 96 is
        the 15 minutes interval completed 23 hours and 45
        minutes prior to interval 1."
    ::= { dsx1IntervalEntry 2 }

dsx1IntervaleSSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Errored Seconds."
    ::= { dsx1IntervalEntry 3 }

dsx1IntervalSESSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Seconds."
    ::= { dsx1IntervalEntry 4 }

dsx1IntervalSEFSSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Framing Seconds."

```

```
 ::= { dsx1IntervalEntry 5 }

dsx1IntervalUASS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds.  This object
         may decrease if the occurrence of unavailable
         seconds occurs across an interval boundary."
    ::= { dsx1IntervalEntry 6 }

dsx1IntervalCSSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Controlled Slip Seconds."
    ::= { dsx1IntervalEntry 7 }

dsx1IntervalPCVs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Path Coding Violations."
    ::= { dsx1IntervalEntry 8 }

dsx1IntervalLESSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Errored Seconds."
    ::= { dsx1IntervalEntry 9 }

dsx1IntervalBESSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Bursty Errored Seconds."
    ::= { dsx1IntervalEntry 10 }

dsx1IntervalDMs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
```

## DESCRIPTION

"The number of Degraded Minutes."

::= { dsx1IntervalEntry 11 }

dsx1IntervalLCVs OBJECT-TYPE  
 SYNTAX PerfIntervalCount  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"The number of Line Code Violations."

::= { dsx1IntervalEntry 12 }

dsx1IntervalValidData OBJECT-TYPE  
 SYNTAX TruthValue  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This variable indicates if the data for this interval is valid."

::= { dsx1IntervalEntry 13 }

-- The DS1 Total Table

dsx1TotalTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF Dsx1TotalEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION

"The DS1 Total Table contains the cumulative sum of the various statistics for the 24 hour period preceding the current interval."

::= { ds1 9 }

dsx1TotalEntry OBJECT-TYPE  
 SYNTAX Dsx1TotalEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION

"An entry in the DS1 Total table."

INDEX { dsx1TotalIndex }

::= { dsx1TotalTable 1 }

Dsx1TotalEntry ::=

|                |                 |
|----------------|-----------------|
| SEQUENCE {     |                 |
| dsx1TotalIndex | InterfaceIndex, |
| dsx1TotalESS   | PerfTotalCount, |
| dsx1TotalSESS  | PerfTotalCount, |
| dsx1TotalSEFSS | PerfTotalCount, |
| dsx1TotalUASS  | PerfTotalCount, |

```

        dsx1TotalCSSs          PerfTotalCount,
        dsx1TotalPCVs          PerfTotalCount,
        dsx1TotalLESS          PerfTotalCount,
        dsx1TotalBESS          PerfTotalCount,
        dsx1TotalDMS           PerfTotalCount,
        dsx1TotalLCVs          PerfTotalCount
    }

dsx1TotalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS1
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is the same interface as identified by the
        same value as a dsx1LineIndex object instance."

    ::= { dsx1TotalEntry 1 }

dsx1TotalESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The sum of Errored Seconds encountered by a DS1
        interface in the previous 24 hour interval.
        Invalid 15 minute intervals count as 0."
    ::= { dsx1TotalEntry 2 }

dsx1TotalSESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Seconds
        encountered by a DS1 interface in the previous 24
        hour interval.  Invalid 15 minute intervals count
        as 0."
    ::= { dsx1TotalEntry 3 }

dsx1TotalSEFSSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Severely Errored Framing Seconds

```

```
        encountered by a DS1 interface in the previous 24
        hour interval.  Invalid 15 minute intervals count
        as 0."
 ::= { dsx1TotalEntry 4 }

dsx1TotalUASSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds encountered by
        a DS1 interface in the previous 24 hour interval.
        Invalid 15 minute intervals count as 0."
 ::= { dsx1TotalEntry 5 }

dsx1TotalCSSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Controlled Slip Seconds encountered
        by a DS1 interface in the previous 24 hour
        interval.  Invalid 15 minute intervals count as
        0."
 ::= { dsx1TotalEntry 6 }

dsx1TotalPCVs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Path Coding Violations encountered
        by a DS1 interface in the previous 24 hour
        interval.  Invalid 15 minute intervals count as
        0."
 ::= { dsx1TotalEntry 7 }

dsx1TotalLESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Errored Seconds encountered by
        a DS1 interface in the previous 24 hour interval.
        Invalid 15 minute intervals count as 0."
 ::= { dsx1TotalEntry 8 }

dsx1TotalBESSs OBJECT-TYPE
```

```

SYNTAX PerfTotalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of Bursty Errored Seconds (BESs)
    encountered by a DS1 interface in the previous 24
    hour interval. Invalid 15 minute intervals count
    as 0."
 ::= { dsx1TotalEntry 9 }

dsx1TotalDMS OBJECT-TYPE
SYNTAX PerfTotalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of Degraded Minutes (DMS) encountered
    by a DS1 interface in the previous 24 hour
    interval. Invalid 15 minute intervals count as
    0."
 ::= { dsx1TotalEntry 10 }

dsx1TotalLCVs OBJECT-TYPE
SYNTAX PerfTotalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of Line Code Violations (LCVs)
    encountered by a DS1 interface in the current 15
    minute interval. Invalid 15 minute intervals
    count as 0."
 ::= { dsx1TotalEntry 11 }

-- The DS1 Channel Table
dsx1ChanMappingTable OBJECT-TYPE
SYNTAX SEQUENCE OF Dsx1ChanMappingEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "The DS1 Channel Mapping table. This table maps a
    DS1 channel number on a particular DS3 into an
    ifIndex. In the presence of DS2s, this table can
    be used to map a DS2 channel number on a DS3 into
    an ifIndex, or used to map a DS1 channel number on
    a DS2 onto an ifIndex."
 ::= { ds1 16 }

dsx1ChanMappingEntry OBJECT-TYPE
SYNTAX Dsx1ChanMappingEntry

```

MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION

"An entry in the DS1 Channel Mapping table. There is an entry in this table corresponding to each ds1 ifEntry within any interface that is channelized to the individual ds1 ifEntry level.

This table is intended to facilitate mapping from channelized interface / channel number to DS1 ifEntry. (e.g. mapping (DS3 ifIndex, DS1 Channel Number) -> ifIndex)

While this table provides information that can also be found in the ifStackTable and dsx1ConfigTable, it provides this same information with a single table lookup, rather than by walking the ifStackTable to find the various constituent ds1 ifTable entries, and testing various dsx1ConfigTable entries to check for the entry with the applicable DS1 channel number."

INDEX { ifIndex, dsx1Ds1ChannelNumber }  
 ::= { dsx1ChanMappingTable 1 }

Dsx1ChanMappingEntry ::=

```

SEQUENCE {
    dsx1ChanMappedIfIndex InterfaceIndex
}

```

dsx1ChanMappedIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

"This object indicates the ifIndex value assigned by the agent for the individual ds1 ifEntry that corresponds to the given DS1 channel number (specified by the INDEX element dsx1Ds1ChannelNumber) of the given channelized interface (specified by INDEX element ifIndex)."

::= { dsx1ChanMappingEntry 1 }

-- The DS1 Far End Current Table

dsx1FarEndCurrentTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx1FarEndCurrentEntry  
 MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The DS1 Far End Current table contains various statistics being collected for the current 15 minute interval. The statistics are collected from the far end messages on the Facilities Data Link. The definitions are the same as described for the near-end information."

::= { ds1 10 }

dsx1FarEndCurrentEntry OBJECT-TYPE

SYNTAX Dsx1FarEndCurrentEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the DS1 Far End Current table."

INDEX { dsx1FarEndCurrentIndex }

::= { dsx1FarEndCurrentTable 1 }

Dsx1FarEndCurrentEntry ::=

SEQUENCE {

|                            |                   |
|----------------------------|-------------------|
| dsx1FarEndCurrentIndex     | InterfaceIndex,   |
| dsx1FarEndTimeElapsed      | INTEGER,          |
| dsx1FarEndValidIntervals   | INTEGER,          |
| dsx1FarEndCurrentESS       | PerfCurrentCount, |
| dsx1FarEndCurrentSESS      | PerfCurrentCount, |
| dsx1FarEndCurrentSEFSS     | PerfCurrentCount, |
| dsx1FarEndCurrentUASS      | PerfCurrentCount, |
| dsx1FarEndCurrentCSSS      | PerfCurrentCount, |
| dsx1FarEndCurrentLESS      | PerfCurrentCount, |
| dsx1FarEndCurrentPCVs      | PerfCurrentCount, |
| dsx1FarEndCurrentBESS      | PerfCurrentCount, |
| dsx1FarEndCurrentDMS       | PerfCurrentCount, |
| dsx1FarEndInvalidIntervals | INTEGER           |

}

dsx1FarEndCurrentIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index value which uniquely identifies the DS1 interface to which this entry is applicable. The interface identified by a particular value of this index is identical to the interface identified by the same value of dsx1LineIndex."

::= { dsx1FarEndCurrentEntry 1 }

**dsx1FarEndTimeElapsed 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 far end current error-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."

::= { dsx1FarEndCurrentEntry 2 }

**dsx1FarEndValidIntervals OBJECT-TYPE**

SYNTAX INTEGER (0..96)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of previous far end intervals for which data was collected. The value will be 96 unless the interface was brought online within the last 24 hours, in which case the value will be the number of complete 15 minute far end intervals since the interface has been online."

::= { dsx1FarEndCurrentEntry 3 }

**dsx1FarEndCurrentESSs OBJECT-TYPE**

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of Far End Errored Seconds."

::= { dsx1FarEndCurrentEntry 4 }

**dsx1FarEndCurrentSESSs OBJECT-TYPE**

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of Far End Severely Errored Seconds."

::= { dsx1FarEndCurrentEntry 5 }

**dsx1FarEndCurrentSEFSSs OBJECT-TYPE**

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
        "The number of Far End Severely Errored Framing
        Seconds."
 ::= { dsx1FarEndCurrentEntry 6 }

dsx1FarEndCurrentUASSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds."
    ::= { dsx1FarEndCurrentEntry 7 }

dsx1FarEndCurrentCSSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Controlled Slip Seconds."
    ::= { dsx1FarEndCurrentEntry 8 }

dsx1FarEndCurrentLESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Line Errored Seconds."
    ::= { dsx1FarEndCurrentEntry 9 }

dsx1FarEndCurrentPCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Path Coding Violations."
    ::= { dsx1FarEndCurrentEntry 10 }

dsx1FarEndCurrentBESSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Bursty Errored Seconds."
    ::= { dsx1FarEndCurrentEntry 11 }

dsx1FarEndCurrentDMs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
```

## DESCRIPTION

"The number of Far End Degraded Minutes."  
 ::= { dsx1FarEndCurrentEntry 12 }

## dsx1FarEndInvalidIntervals OBJECT-TYPE

SYNTAX INTEGER (0..96)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of intervals in the range from 0 to dsx1FarEndValidIntervals 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)."  
 ::= { dsx1FarEndCurrentEntry 13 }

-- The DS1 Far End Interval Table

## dsx1FarEndIntervalTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx1FarEndIntervalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The DS1 Far End Interval Table contains various statistics collected by each DS1 interface over the previous 24 hours of operation. The past 24 hours are broken into 96 completed 15 minute intervals. Each row in this table represents one such interval (identified by dsx1FarEndIntervalNumber) for one specific instance (identified by dsx1FarEndIntervalIndex)."  
 ::= { ds1 11 }

## dsx1FarEndIntervalEntry OBJECT-TYPE

SYNTAX Dsx1FarEndIntervalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry in the DS1 Far End Interval table."

INDEX { dsx1FarEndIntervalIndex,  
 dsx1FarEndIntervalNumber }

::= { dsx1FarEndIntervalTable 1 }

Dsx1FarEndIntervalEntry ::=

SEQUENCE {

|                          |                    |
|--------------------------|--------------------|
| dsx1FarEndIntervalIndex  | InterfaceIndex,    |
| dsx1FarEndIntervalNumber | INTEGER,           |
| dsx1FarEndIntervalESS    | PerfIntervalCount, |

```

        dsx1FarEndIntervalSESS      PerfIntervalCount,
        dsx1FarEndIntervalSEFSS     PerfIntervalCount,
        dsx1FarEndIntervalUASS      PerfIntervalCount,
        dsx1FarEndIntervalCSSs      PerfIntervalCount,
        dsx1FarEndIntervalLESS      PerfIntervalCount,
        dsx1FarEndIntervalPCVs      PerfIntervalCount,
        dsx1FarEndIntervalBESS      PerfIntervalCount,
        dsx1FarEndIntervalDMS       PerfIntervalCount,
        dsx1FarEndIntervalValidData TruthValue
    }

dsx1FarEndIntervalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS1
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is identical to the interface identified by
        the same value of dsx1LineIndex."
    ::= { dsx1FarEndIntervalEntry 1 }

dsx1FarEndIntervalNumber OBJECT-TYPE
    SYNTAX  INTEGER (1..96)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "A number between 1 and 96, where 1 is the most
        recently completed 15 minute interval and 96 is
        the 15 minutes interval completed 23 hours and 45
        minutes prior to interval 1."
    ::= { dsx1FarEndIntervalEntry 2 }

dsx1FarEndIntervaleSSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Errored Seconds."
    ::= { dsx1FarEndIntervalEntry 3 }

dsx1FarEndIntervaleSSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Severely Errored Seconds."

```

```
 ::= { dsx1FarEndIntervalEntry 4 }

dsx1FarEndIntervalSEFSS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Severely Errored Framing
        Seconds."
    ::= { dsx1FarEndIntervalEntry 5 }

dsx1FarEndIntervalUASS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds."
    ::= { dsx1FarEndIntervalEntry 6 }

dsx1FarEndIntervalCSSS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Controlled Slip Seconds."
    ::= { dsx1FarEndIntervalEntry 7 }

dsx1FarEndIntervalLESS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Line Errored Seconds."
    ::= { dsx1FarEndIntervalEntry 8 }

dsx1FarEndIntervalPCVs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Path Coding Violations."
    ::= { dsx1FarEndIntervalEntry 9 }

dsx1FarEndIntervalBESS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
```

## DESCRIPTION

"The number of Far End Bursty Errored Seconds."  
 ::= { dsx1FarEndIntervalEntry 10 }

## dsx1FarEndIntervalDms OBJECT-TYPE

SYNTAX PerfIntervalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of Far End Degraded Minutes."  
 ::= { dsx1FarEndIntervalEntry 11 }

## dsx1FarEndIntervalValidData OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This variable indicates if the data for this  
 interval is valid."  
 ::= { dsx1FarEndIntervalEntry 12 }

## -- The DS1 Far End Total Table

## dsx1FarEndTotalTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx1FarEndTotalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The DS1 Far End Total Table contains the  
 cumulative sum of the various statistics for the  
 24 hour period preceding the current interval."  
 ::= { ds1 12 }

## dsx1FarEndTotalEntry OBJECT-TYPE

SYNTAX Dsx1FarEndTotalEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry in the DS1 Far End Total table."  
 INDEX { dsx1FarEndTotalIndex }  
 ::= { dsx1FarEndTotalTable 1 }

## Dsx1FarEndTotalEntry ::=

SEQUENCE {

|                       |                 |
|-----------------------|-----------------|
| dsx1FarEndTotalIndex  | InterfaceIndex, |
| dsx1FarEndTotalESSs   | PerfTotalCount, |
| dsx1FarEndTotalSESSs  | PerfTotalCount, |
| dsx1FarEndTotalSEFSSs | PerfTotalCount, |

```

        dsx1FarEndTotalUASS          PerfTotalCount,
        dsx1FarEndTotalCSSs          PerfTotalCount,
        dsx1FarEndTotalLESSs         PerfTotalCount,
        dsx1FarEndTotalPCVs          PerfTotalCount,
        dsx1FarEndTotalBESSs         PerfTotalCount,
        dsx1FarEndTotalDMS           PerfTotalCount
    }

dsx1FarEndTotalIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The index value which uniquely identifies the DS1
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is identical to the interface identified by
        the same value of dsx1LineIndex."

    ::= { dsx1FarEndTotalEntry 1 }

dsx1FarEndTotalESSs OBJECT-TYPE
    SYNTAX      PerfTotalCount
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of Far End Errored Seconds encountered
        by a DS1 interface in the previous 24 hour
        interval.  Invalid 15 minute intervals count as
        0."

    ::= { dsx1FarEndTotalEntry 2 }

dsx1FarEndTotalSESSs OBJECT-TYPE
    SYNTAX      PerfTotalCount
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of Far End Severely Errored Seconds
        encountered by a DS1 interface in the previous 24
        hour interval.  Invalid 15 minute intervals count
        as 0."

    ::= { dsx1FarEndTotalEntry 3 }

dsx1FarEndTotalSEFSSs OBJECT-TYPE
    SYNTAX      PerfTotalCount
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

```
        "The number of Far End Severely Errored Framing
        Seconds encountered by a DS1 interface in the
        previous 24 hour interval. Invalid 15 minute
        intervals count as 0."
 ::= { dsx1FarEndTotalEntry 4 }

dsx1FarEndTotalUASS OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Unavailable Seconds encountered by
        a DS1 interface in the previous 24 hour interval.
        Invalid 15 minute intervals count as 0."
 ::= { dsx1FarEndTotalEntry 5 }

dsx1FarEndTotalCSSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Controlled Slip Seconds
        encountered by a DS1 interface in the previous 24
        hour interval. Invalid 15 minute intervals count
        as 0."
 ::= { dsx1FarEndTotalEntry 6 }

dsx1FarEndTotalLESS OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Line Errored Seconds
        encountered by a DS1 interface in the previous 24
        hour interval. Invalid 15 minute intervals count
        as 0."
 ::= { dsx1FarEndTotalEntry 7 }

dsx1FarEndTotalPCVs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Far End Path Coding Violations
        reported via the far end block error count
        encountered by a DS1 interface in the previous 24
        hour interval. Invalid 15 minute intervals count
        as 0."
```

```
::= { dsx1FarEndTotalEntry 8 }
```

```
dsx1FarEndTotalBESS OBJECT-TYPE
```

```
SYNTAX PerfTotalCount
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The number of Bursty Errored Seconds (BESS)
    encountered by a DS1 interface in the previous 24
    hour interval. Invalid 15 minute intervals count
    as 0."
```

```
::= { dsx1FarEndTotalEntry 9 }
```

```
dsx1FarEndTotalDMs OBJECT-TYPE
```

```
SYNTAX PerfTotalCount
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The number of Degraded Minutes (DMs) encountered
    by a DS1 interface in the previous 24 hour
    interval. Invalid 15 minute intervals count as
    0."
```

```
::= { dsx1FarEndTotalEntry 10 }
```

```
-- The DS1 Fractional Table
```

```
dsx1FracTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF Dsx1FracEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS deprecated
```

```
DESCRIPTION
```

```
    "This table is deprecated in favour of using
    ifStackTable.
```

```
    The table was mandatory for systems dividing a DS1
    into channels containing different data streams
    that are of local interest. Systems which are
    indifferent to data content, such as CSUs, need
    not implement it.
```

```
    The DS1 fractional table identifies which DS1
    channels associated with a CSU are being used to
    support a logical interface, i.e., an entry in the
    interfaces table from the Internet-standard MIB.
```

```
    For example, consider an application managing a
    North American ISDN Primary Rate link whose
    division is a 384 kbit/s H1 _B_ Channel for Video,
```

a second H1 for data to a primary routing peer, and 12 64 kbit/s H0 \_B\_ Channels. Consider that some subset of the H0 channels are used for voice and the remainder are available for dynamic data calls.

We count a total of 14 interfaces multiplexed onto the DS1 interface. Six DS1 channels (for the sake of the example, channels 1..6) are used for Video, six more (7..11 and 13) are used for data, and the remaining 12 are in channels 12 and 14..24.

Let us further imagine that ifIndex 2 is of type DS1 and refers to the DS1 interface, and that the interfaces layered onto it are numbered 3..16.

We might describe the allocation of channels, in the dsx1FracTable, as follows:

|                          |                           |
|--------------------------|---------------------------|
| dsx1FracIfIndex.2. 1 = 3 | dsx1FracIfIndex.2.13 = 4  |
| dsx1FracIfIndex.2. 2 = 3 | dsx1FracIfIndex.2.14 = 6  |
| dsx1FracIfIndex.2. 3 = 3 | dsx1FracIfIndex.2.15 = 7  |
| dsx1FracIfIndex.2. 4 = 3 | dsx1FracIfIndex.2.16 = 8  |
| dsx1FracIfIndex.2. 5 = 3 | dsx1FracIfIndex.2.17 = 9  |
| dsx1FracIfIndex.2. 6 = 3 | dsx1FracIfIndex.2.18 = 10 |
| dsx1FracIfIndex.2. 7 = 4 | dsx1FracIfIndex.2.19 = 11 |
| dsx1FracIfIndex.2. 8 = 4 | dsx1FracIfIndex.2.20 = 12 |
| dsx1FracIfIndex.2. 9 = 4 | dsx1FracIfIndex.2.21 = 13 |
| dsx1FracIfIndex.2.10 = 4 | dsx1FracIfIndex.2.22 = 14 |
| dsx1FracIfIndex.2.11 = 4 | dsx1FracIfIndex.2.23 = 15 |
| dsx1FracIfIndex.2.12 = 5 | dsx1FracIfIndex.2.24 = 16 |

For North American (DS1) interfaces, there are 24 legal channels, numbered 1 through 24.

For G.704 interfaces, there are 31 legal channels, numbered 1 through 31. The channels (1..31) correspond directly to the equivalently numbered time-slots."

::= { ds1 13 }

dsx1FracEntry OBJECT-TYPE

SYNTAX Dsx1FracEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"An entry in the DS1 Fractional table."

INDEX { dsx1FracIndex, dsx1FracNumber }

::= { dsx1FracTable 1 }

```
Dsx1FracEntry ::=
    SEQUENCE {
        dsx1FracIndex      INTEGER,
        dsx1FracNumber      INTEGER,
        dsx1FracIfIndex     INTEGER
    }

dsx1FracIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..'7fffffff'h)
    MAX-ACCESS  read-only
    STATUS  deprecated
    DESCRIPTION
        "The index value which uniquely identifies the
        DS1 interface to which this entry is applicable
        The interface identified by a particular
        value of this index is the same interface as
        identified by the same value an dsx1LineIndex
        object instance."
    ::= { dsx1FracEntry 1 }

dsx1FracNumber OBJECT-TYPE
    SYNTAX  INTEGER (1..31)
    MAX-ACCESS  read-only
    STATUS  deprecated
    DESCRIPTION
        "The channel number for this entry."
    ::= { dsx1FracEntry 2 }

dsx1FracIfIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..'7fffffff'h)
    MAX-ACCESS  read-write
    STATUS  deprecated
    DESCRIPTION
        "An index value that uniquely identifies an
        interface. The interface identified by a particular
        value of this index is the same interface
        as identified by the same value an ifIndex
        object instance. If no interface is currently using
        a channel, the value should be zero. If a
        single interface occupies more than one time
        slot, that ifIndex value will be found in multiple
        time slots."
    ::= { dsx1FracEntry 3 }

-- Ds1 TRAPS

ds1Traps OBJECT IDENTIFIER ::= { ds1 15 }
```

```

dsx1LineStatusChange NOTIFICATION-TYPE
  OBJECTS { dsx1LineStatus,
             dsx1LineStatusLastChange }
  STATUS current
  DESCRIPTION
    "A dsx1LineStatusChange trap is sent when the
    value of an instance dsx1LineStatus changes. It
    can be utilized by an NMS to trigger polls. When
    the line status change results from a higher level
    line status change (i.e. ds3), then no traps for
    the ds1 are sent."
  ::= { ds1Traps 0 1 }

-- conformance information
ds1Conformance OBJECT IDENTIFIER ::= { ds1 14 }

ds1Groups      OBJECT IDENTIFIER ::= { ds1Conformance 1 }
ds1Compliances OBJECT IDENTIFIER ::= { ds1Conformance 2 }

-- compliance statements

ds1Compliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for T1 and E1
    interfaces."
  MODULE -- this module
    MANDATORY-GROUPS { ds1NearEndConfigGroup,
                       ds1NearEndStatisticsGroup }

    GROUP ds1FarEndGroup
    DESCRIPTION
      "Implementation of this group is optional for all
      systems that attach to a DS1 Interface."

    GROUP ds1NearEndOptionalConfigGroup
    DESCRIPTION
      "Implementation of this group is optional for all
      systems that attach to a DS1 Interface."

    GROUP ds1DS2Group
    DESCRIPTION
      "Implementation of this group is mandatory for all
      systems that attach to a DS2 Interface."

    GROUP ds1TransStatsGroup

```

## DESCRIPTION

"This group is the set of statistics appropriate for all systems which attach to a DS1 Interface running transparent or unFramed lineType."

GROUP dslChanMappingGroup

## DESCRIPTION

"This group is the set of objects for mapping a DS3 Channel (dslChannelNumber) to ifIndex."

Implementation of this group is mandatory for systems which support the channelization of DS3s into DS1s."

OBJECT dsx1LineType

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set the line type is not required."

OBJECT dsx1LineCoding

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set the line coding is not required."

OBJECT dsx1SendCode

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set the send code is not required."

OBJECT dsx1LoopbackConfig

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set loopbacks is not required."

OBJECT dsx1SignalMode

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set the signal mode is not required."

OBJECT dsx1TransmitClockSource

MIN-ACCESS read-only

## DESCRIPTION

"The ability to set the transmit clock source is

not required."

OBJECT dsx1Fdl  
MIN-ACCESS read-only  
DESCRIPTION

"The ability to set the FDL is not required."

OBJECT dsx1LineLength  
MIN-ACCESS read-only  
DESCRIPTION

"The ability to set the line length is not required."

OBJECT dsx1Channelization  
MIN-ACCESS read-only  
DESCRIPTION

"The ability to set the channelization is not required."

::= { dslCompliances 1 }

dslMibT1PriCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Compliance statement for using this MIB for ISDN  
Primary Rate interfaces on T1 lines."

MODULE

MANDATORY-GROUPS { dslNearEndConfigGroup,  
dslNearEndStatisticsGroup }

OBJECT dsx1LineType

SYNTAX INTEGER {  
dsx1ESF(2) -- Intl Spec would be G704(2)  
-- or I.431(4)  
}

MIN-ACCESS read-only

DESCRIPTION

"Line type for T1 ISDN Primary Rate  
interfaces."

OBJECT dsx1LineCoding

SYNTAX INTEGER {  
dsx1B8ZS(2)  
}

MIN-ACCESS read-only

DESCRIPTION

"Type of Zero Code Suppression for  
T1 ISDN Primary Rate interfaces."

OBJECT dsx1SignalMode

```

    SYNTAX INTEGER {
        none(1), -- if there is no signaling channel
        messageOriented(4)
    }
    MIN-ACCESS read-only
    DESCRIPTION
        "Possible signaling modes for
        T1 ISDN Primary Rate interfaces."

OBJECT dsx1TransmitClockSource
    SYNTAX INTEGER {
        loopTiming(1)
    }
    MIN-ACCESS read-only
    DESCRIPTION
        "The transmit clock is derived from
        received clock on ISDN Primary Rate
        interfaces."

OBJECT dsx1Fdl
    MIN-ACCESS read-only
    DESCRIPTION
        "Facilities Data Link usage on T1 ISDN
        Primary Rate interfaces.
        Note: Eventually dsx1Att-54016(4) is to be
        used here since the line type is ESF."

OBJECT dsx1Channelization
    MIN-ACCESS read-only
    DESCRIPTION
        "The ability to set the channelization
        is not required."
::= { ds1Compliances 2 }

ds1MibE1PriCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Compliance statement for using this MIB for ISDN
        Primary Rate interfaces on E1 lines."
    MODULE
        MANDATORY-GROUPS { ds1NearEndConfigGroup,
                            ds1NearEndStatisticsGroup }
    OBJECT dsx1LineType
        SYNTAX INTEGER {
            dsx1E1CRC(5)
        }
        MIN-ACCESS read-only

```

## DESCRIPTION

"Line type for E1 ISDN Primary Rate interfaces."

## OBJECT dsx1LineCoding

SYNTAX INTEGER {  
    dsx1HDB3(3)  
}

MIN-ACCESS read-only

## DESCRIPTION

"Type of Zero Code Suppression for  
E1 ISDN Primary Rate interfaces."

## OBJECT dsx1SignalMode

SYNTAX INTEGER {  
    messageOriented(4)  
}

MIN-ACCESS read-only

## DESCRIPTION

"Signaling on E1 ISDN Primary Rate interfaces  
is always message oriented."

## OBJECT dsx1TransmitClockSource

SYNTAX INTEGER {  
    loopTiming(1)  
}

MIN-ACCESS read-only

## DESCRIPTION

"The transmit clock is derived from received  
clock on ISDN Primary Rate interfaces."

## OBJECT dsx1Fdl

MIN-ACCESS read-only

## DESCRIPTION

"Facilities Data Link usage on E1 ISDN  
Primary Rate interfaces.

Note: There is a 'M-Channel' in E1,  
using National Bit Sa4 (G704,  
Table 4a). It is used to implement  
management features between ET  
and NT. This is different to  
FDL in T1, which is used to carry  
control signals and performance  
data. In E1, control and status  
signals are carried using National  
Bits Sa5, Sa6 and A (RAI Ind.).  
This indicates that only the other(1) or  
eventually the dsx1Fdl-none(8) bits should

be set in this object for E1 PRI."

```

OBJECT dsx1Channelization
  MIN-ACCESS read-only
  DESCRIPTION
    "The ability to set the channelization is not
    required."
 ::= { ds1Compliances 3 }

```

```

ds1Ds2Compliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for using this MIB for DS2
    interfaces."
  MODULE
    MANDATORY-GROUPS { ds1DS2Group }

    OBJECT dsx1Channelization
      MIN-ACCESS read-only
      DESCRIPTION
        "The ability to set the channelization is not
        required."
    ::= { ds1Compliances 4 }

```

-- units of conformance

```

ds1NearEndConfigGroup OBJECT-GROUP
  OBJECTS { dsx1LineIndex,
            dsx1TimeElapsed,
            dsx1ValidIntervals,
            dsx1LineType,
            dsx1LineCoding,
            dsx1SendCode,
            dsx1CircuitIdentifier,
            dsx1LoopbackConfig,
            dsx1LineStatus,
            dsx1SignalMode,
            dsx1TransmitClockSource,
            dsx1Fdl,
            dsx1InvalidIntervals,
            dsx1LineLength,
            dsx1LoopbackStatus,
            dsx1Ds1ChannelNumber,
            dsx1Channelization }
  STATUS current
  DESCRIPTION
    "A collection of objects providing configuration

```

```

        information applicable to all DS1 interfaces."
 ::= { ds1Groups 1 }

```

ds1NearEndStatisticsGroup OBJECT-GROUP

```

OBJECTS { dsx1CurrentIndex,
          dsx1CurrentESSs,
          dsx1CurrentSESSs,
          dsx1CurrentSEFSSs,
          dsx1CurrentUASSs,
          dsx1CurrentCSSs,
          dsx1CurrentPCVs,
          dsx1CurrentLESSs,
          dsx1CurrentBESSs,
          dsx1CurrentDMs,
          dsx1CurrentLCVs,
          dsx1IntervalIndex,
          dsx1IntervalNumber,
          dsx1IntervalESSs,
          dsx1IntervalSESSs,
          dsx1IntervalSEFSSs,
          dsx1IntervalUASSs,
          dsx1IntervalCSSs,
          dsx1IntervalPCVs,
          dsx1IntervalLESSs,
          dsx1IntervalBESSs,
          dsx1IntervalDMs,
          dsx1IntervalLCVs,
          dsx1IntervalValidData,
          dsx1TotalIndex,
          dsx1TotalESSs,
          dsx1TotalSESSs,
          dsx1TotalSEFSSs,
          dsx1TotalUASSs,
          dsx1TotalCSSs,
          dsx1TotalPCVs,
          dsx1TotalLESSs,
          dsx1TotalBESSs,
          dsx1TotalDMs,
          dsx1TotalLCVs }

```

STATUS current

DESCRIPTION

```

        "A collection of objects providing statistics
        information applicable to all DS1 interfaces."
 ::= { ds1Groups 2 }

```

ds1FarEndGroup OBJECT-GROUP

```

OBJECTS { dsx1FarEndCurrentIndex,
          dsx1FarEndTimeElapsed,

```

```

    dsx1FarEndValidIntervals,
    dsx1FarEndCurrentESS,
    dsx1FarEndCurrentSESS,
    dsx1FarEndCurrentSEFSS,
    dsx1FarEndCurrentUASS,
    dsx1FarEndCurrentCSSs,
    dsx1FarEndCurrentLESS,
    dsx1FarEndCurrentPCVs,
    dsx1FarEndCurrentBESS,
    dsx1FarEndCurrentDMs,
    dsx1FarEndInvalidIntervals,
    dsx1FarEndIntervalIndex,
    dsx1FarEndIntervalNumber,
    dsx1FarEndIntervalESS,
    dsx1FarEndIntervalSESS,
    dsx1FarEndIntervalSEFSS,
    dsx1FarEndIntervalUASS,
    dsx1FarEndIntervalCSSs,
    dsx1FarEndIntervalLESS,
    dsx1FarEndIntervalPCVs,
    dsx1FarEndIntervalBESS,
    dsx1FarEndIntervalDMs,
    dsx1FarEndIntervalValidData,
    dsx1FarEndTotalIndex,
    dsx1FarEndTotalESS,
    dsx1FarEndTotalSESS,
    dsx1FarEndTotalSEFSS,
    dsx1FarEndTotalUASS,
    dsx1FarEndTotalCSSs,
    dsx1FarEndTotalLESS,
    dsx1FarEndTotalPCVs,
    dsx1FarEndTotalBESS,
    dsx1FarEndTotalDMs }
STATUS    current
DESCRIPTION
    "A collection of objects providing remote
    configuration and statistics information."
 ::= { ds1Groups 3 }

```

ds1DeprecatedGroup OBJECT-GROUP

```

    OBJECTS { dsx1IfIndex,
              dsx1FracIndex,
              dsx1FracNumber,
              dsx1FracIfIndex }

```

STATUS deprecated

DESCRIPTION

"A collection of obsolete objects that may be implemented for backwards compatibility."

```

 ::= { ds1Groups 4 }

ds1NearEndOptionalConfigGroup OBJECT-GROUP
    OBJECTS { dsx1LineStatusLastChange,
               dsx1LineStatusChangeTrapEnable }

    STATUS      current
    DESCRIPTION
        "A collection of objects that may be implemented
         on DS1 and DS2 interfaces."
    ::= { ds1Groups 5 }

ds1DS2Group OBJECT-GROUP
    OBJECTS { dsx1LineIndex,
               dsx1LineType,
               dsx1LineCoding,
               dsx1SendCode,
               dsx1LineStatus,
               dsx1SignalMode,
               dsx1TransmitClockSource,
               dsx1Channelization }

    STATUS      current
    DESCRIPTION
        "A collection of objects providing information
         about DS2 (6,312 kbps) and E2 (8,448 kbps)
         systems."
    ::= { ds1Groups 6 }

ds1TransStatsGroup OBJECT-GROUP
    OBJECTS { dsx1CurrentESSs,
               dsx1CurrentSESSs,
               dsx1CurrentUASSs,
               dsx1IntervalESSs,
               dsx1IntervalSESSs,
               dsx1IntervalUASSs,
               dsx1TotalESSs,
               dsx1TotalSESSs,
               dsx1TotalUASSs }

    STATUS      current
    DESCRIPTION
        "A collection of objects which are the
         statistics which can be collected from a ds1
         interface that is running transparent or unframed
         lineType. Statistics not in this list should
         return noSuchInstance."
    ::= { ds1Groups 7 }

ds1NearEndOptionalTrapGroup NOTIFICATION-GROUP

```

```
NOTIFICATIONS { dsx1LineStatusChange }
STATUS      current
DESCRIPTION
    "A collection of notifications that may be
    implemented on DS1 and DS2 interfaces."
::= { dslGroups 8 }

dslChanMappingGroup OBJECT-GROUP
OBJECTS { dsx1ChanMappedIfIndex }
STATUS      current
DESCRIPTION
    "A collection of objects that give an mapping of
    DS3 Channel (dslChannelNumber) to ifIndex."
::= { dslGroups 9 }

END
```

#### 4. Appendix A - Use of dsx1IfIndex and dsx1LineIndex

This Appendix exists to document the previous use of dsx1IfIndex and dsx1LineIndex and to clarify the relationship of dsx1LineIndex as defined in rfc1406 with the dsx1LineIndex as defined in this document.

The following shows the old and new definitions and the relationship:

[New Definition]: "This object should be made equal to ifIndex. The next paragraph describes its previous usage. Making the object equal to ifIndex allows proper use of ifStackTable and ds0/ds0bundle mibs.

[Old Definition]: "This object is the identifier of a DS1 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS1 interface, it should have the same value as ifIndex. Otherwise, number the dsx1LineIndices with a unique identifier following the rules of choosing a number that is greater than ifNumber and numbering the inside interfaces (e.g., equipment side) with even numbers and outside interfaces (e.g., network side) with odd numbers."

When the "Old Definition" was created, it was described this way to allow a manager to treat the value \_as if\_ it were and ifIndex, i.e. the value would either be: 1) an ifIndex value or 2) a value that was guaranteed to be different from all valid ifIndex values.

The new definition is a subset of that definition, i.e. the value is always an ifIndex value.

The following is Section 3.1 from rfc1406:

Different physical configurations for the support of SNMP with DS1 equipment exist. To accommodate these scenarios, two different indices for DS1 interfaces are introduced in this MIB. These indices are dsx1IfIndex and dsx1LineIndex.

External interface scenario: the SNMP Agent represents all managed DS1 lines as external interfaces (for example, an Agent residing on the device supporting DS1 interfaces directly):

For this scenario, all interfaces are assigned an integer value equal to ifIndex, and the following applies:

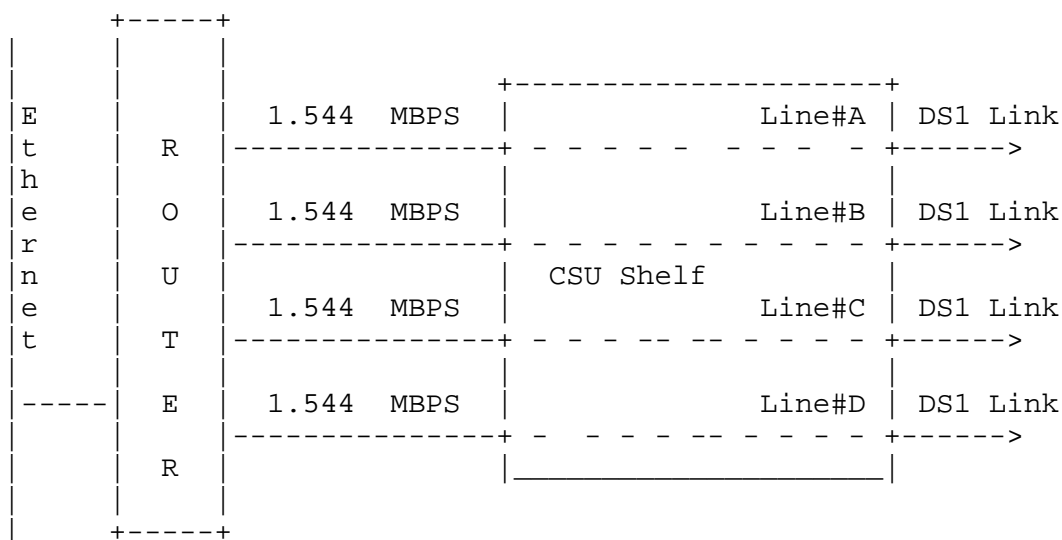
ifIndex=dsx1IfIndex=dsx1LineIndex for all interfaces.

The dsx1IfIndex column of the DS1 Configuration table relates each DS1 interface to its corresponding interface (ifIndex) in the Internet-standard MIB (MIB-II STD 17, RFC1213).

External&Internal interface scenario: the SNMP Agents resides on an host external from the device supporting DS1 interfaces (e.g., a router). The Agent represents both the host and the DS1 device. The index dsx1LineIndex is used to not only represent the DS1 interfaces external from the host/DS1-device combination, but also the DS1 interfaces connecting the host and the DS1 device. The index dsx1IfIndex is always equal to ifIndex.

Example:

A shelf full of CSUs connected to a Router. An SNMP Agent residing on the router proxies for itself and the CSU. The router has also an Ethernet interface:



The assignment of the index values could for example be:

| ifIndex (= dsx1IfIndex) |                     | dsx1LineIndex |
|-------------------------|---------------------|---------------|
| 1                       | NA                  | NA (Ethernet) |
| 2                       | Line#A Router Side  | 6             |
| 2                       | Line#A Network Side | 7             |
| 3                       | Line#B Router Side  | 8             |
| 3                       | Line#B Network Side | 9             |
| 4                       | Line#C Router Side  | 10            |

|   |        |              |    |
|---|--------|--------------|----|
| 4 | Line#C | Network Side | 11 |
| 5 | Line#D | Router Side  | 12 |
| 5 | Line#D | Network Side | 13 |

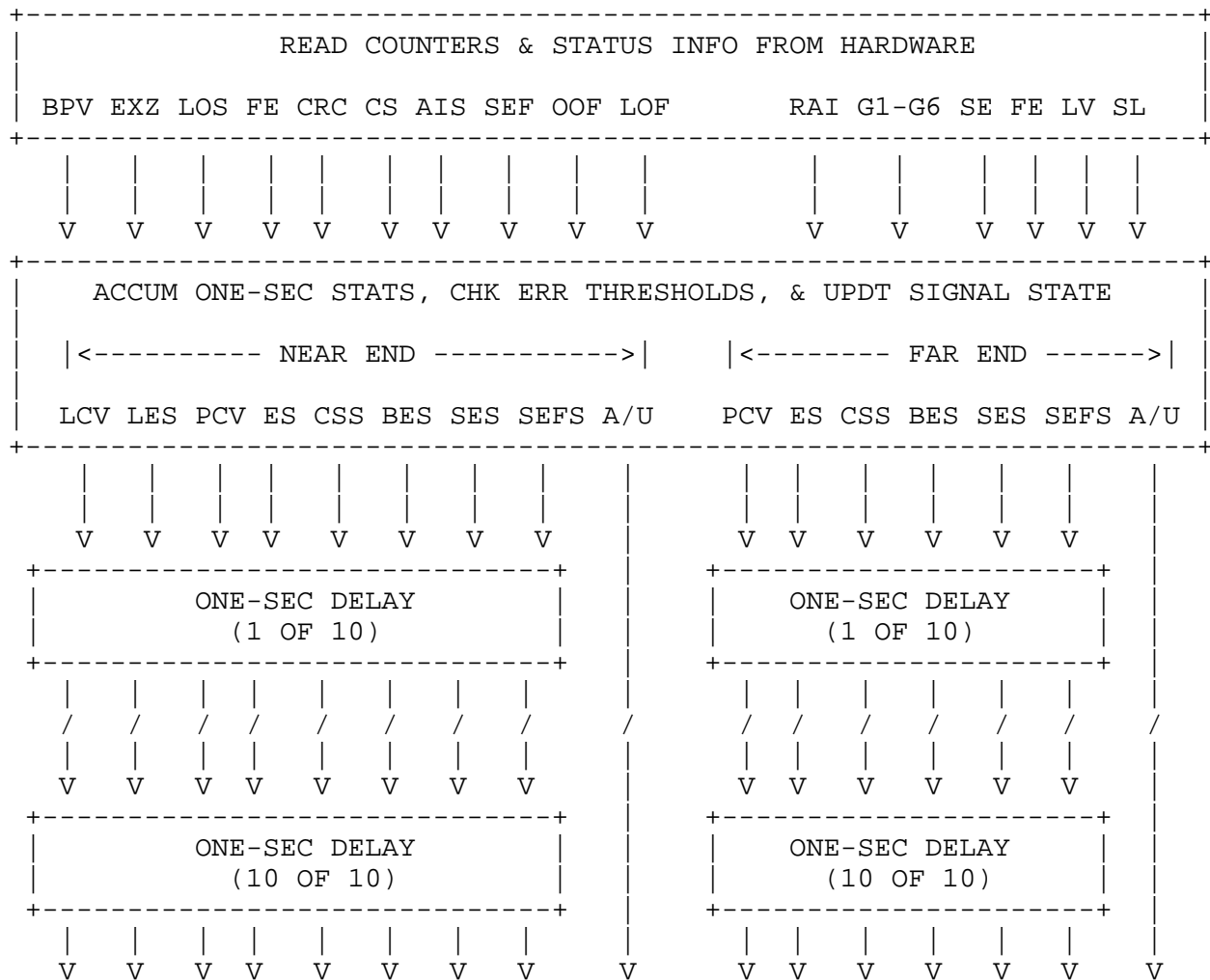
For this example, ifNumber is equal to 5. Note the following description of dsxlLineIndex: the dsxlLineIndex identifies a DS1 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS1 interface, it should have the same value as ifIndex. Otherwise, number the dsxlLineIndices with a unique identifier following the rules of choosing a number greater than ifNumber and numbering inside interfaces (e.g., equipment side) with even numbers and outside interfaces (e.g., network side) with odd numbers.

If the CSU shelf is managed by itself by a local SNMP Agent, the situation would be:

| ifIndex (= dsxlIfIndex) |        |              | dsxlLineIndex |
|-------------------------|--------|--------------|---------------|
| 1                       | Line#A | Network Side | 1             |
| 2                       | Line#A | RouterSide   | 2             |
| 3                       | Line#B | Network Side | 3             |
| 4                       | Line#B | RouterSide   | 4             |
| 5                       | Line#C | Network Side | 5             |
| 6                       | Line#C | Router Side  | 6             |
| 7                       | Line#D | Network Side | 7             |
| 8                       | Line#D | Router Side  | 8             |

## 5. Appendix B - The delay approach to Unavailable Seconds.

This procedure is illustrated below for a DS1 ESF interface. Similar rules would apply for other DS1, DS2, and E1 interface variants. The procedure guarantees that the statistical counters are correctly updated at all times, although they lag real time by 10 seconds. At the end of each 15 minutes interval the current interval counts are transferred to the most recent interval entry and each interval is shifted up by one position, with the oldest being discarded if necessary in order to make room. The current interval counts then start over from zero. Note, however, that the signal state calculation does not start afresh at each interval boundary; rather, signal state information is retained across interval boundaries.



```

+-----+
|                                     |
|               UPDATE STATISTICS COUNTERS               |
|                                     |
|<----- NEAR END ----->| |<----- FAR END ----->|
|                                     |
|LCV LES PCV ES CSS BES SES SEFS UAS DM PCV ES CSS BES SES SEFS UAS DM|
|                                     |
+-----+

```

Note that if such a procedure is adopted there is no current interval data for the first ten seconds after a system comes up. noSuchInstance must be returned if a management station attempts to access the current interval counters during this time.

It is an implementation-specific matter whether an agent assumes that the initial state of the interface is available or unavailable.

## 6. 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.

## 7. Acknowledgments

This document was produced by the Trunk MIB Working Group.

## 8. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.
- [6] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [7] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] 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.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, January 1998.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, January 1998.

- [13] 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.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2273, January 1998.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, January 1998.
- [16] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2233, November 1997.
- [17] AT&T Information Systems, AT&T ESF DS1 Channel Service Unit User's Manual, 999-100-305, February 1988.
- [18] AT&T Technical Reference, Requirements for Interfacing Digital Terminal Equipment to Services Employing the Extended Superframe Format, Publication 54016, May 1988.
- [19] American National Standard for Telecommunications -- Carrier-to-Customer Installation - DS1 Metallic Interface, T1.403, February 1989.
- [20] CCITT Specifications Volume III, Recommendation G.703, Physical/Electrical Characteristics of Hierarchical Digital Interfaces, April 1991.
- [21] ITU-T G.704: Synchronous frame structures used at 1544, 6312, 2048, 8488 and 44 736 kbit/s Hierarchical Levels, July 1995.
- [22] American National Standard for Telecommunications -- Digital Hierarchy -- Layer 1 In-Service Digital Transmission Performance Monitoring, T1.231, Sept 1993.
- [23] CCITT Specifications Volume IV, Recommendation O.162, Equipment To Perform In Service Monitoring On 2048 kbit/s Signals, July 1988.
- [24] CCITT Specifications Volume III, Recommendation G.821, Error Performance Of An International Digital Connection Forming Part Of An Integrated Services Digital Network, July 1988.
- [25] AT&T Technical Reference, Technical Reference 62411, ACCUNET T1.5 Service Description And Interface Specification, December 1990.

- [26] CCITT Specifications Volume III, Recommendation G.706, Frame Alignment and Cyclic Redundancy Check (CRC) Procedures Relating to Basic Frame Structures Defined in Recommendation G.704, July 1988.
- [27] CCITT Specifications Volume III, Recommendation G.732, Characteristics Of Primary PCM Multiplex Equipment Operating at 2048 kbit/s, July 1988.
- [28] Fowler, D., "Definitions of Managed Objects for the DS3/E3 Interface Types", RFC 2496, January 1999.
- [29] Brown, T., and Tesink, K., "Definitions of Managed Objects for the SONET/SDH Interface Type", Work in Progress.
- [30] Fowler, D., "Definitions of Managed Objects for the DS0 and DS0Bundle Interface Types", RFC 2494, January 1999.
- [31] ITU-T G.775: Loss of signal (LOS) and alarm indication signal (AIS) defect detection and clearance criteria, May 1995.
- [32] ITU-T G.826: Error performance parameters and objectives for international, constant bit rate digital paths at or above the primary rate, November 1993.
- [33] American National Standard for Telecommunications -- Digital Hierarchy - Electrical Interfaces, T1.102, December 1993.
- [34] American National Standard for Telecommunications -- Digital Hierarchy - Format Specifications, T1.107, August 1988.
- [35] Tesink, K., "Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals", RFC XXXX, January 1999.

## 9. Security Considerations

SNMPv1 by itself is such an insecure 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 (read) 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 2274 [12] and the View-based Access Control Model RFC 2275 [15] 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 those objects only to those principals (users) that have legitimate rights to access them.

Setting any of the following objects to an inappropriate value can cause loss of traffic. The definition of inappropriate varies for each object. In the case of `dsx1LineType`, for example, both ends of a `ds1/e1` must have the same value in order for traffic to flow. In the case of `dsx1SendCode` and `dsx1LoopbackConfig`, for another example, traffic may stop transmitting when particular loopbacks are applied.

- `dsx1LineType`
- `dsx1LineCoding`
- `dsx1SendCode`
- `dsx1LoopbackConfig`
- `dsx1SignalMode`
- `dsx1TransmitClockSource`
- `dsx1Fdl`
- `dsx1LineLength`
- `dsx1Channelization`

Setting the following object is mischevious, but not harmful to traffic.

- `dsx1CircuitIdentifier`

Setting the following object can cause an increase in the number of traps received by the network management station.

- `dsx1LineStatusChangeTrabEnable`

## 10. Author's Address

David Fowler  
Newbridge Networks  
600 March Road  
Kanata, Ontario, Canada K2K 2E6

Phone: (613) 599-3600, ext 6559  
EMail: davef@newbridge.com

## 11. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

