

Network Working Group  
Request for Comments: 2496  
Obsoletes: 1407  
Category: Standards Track

D. Fowler, Editor  
Newbridge Networks  
January 1999

## Definitions of Managed Objects for the DS3/E3 Interface Type

### 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 DS3 and E3 interfaces. This document is a companion document with Definitions of Managed Objects for the DS0 (RFC 2494 [25]), DS1/E1/DS2/E2 (RFC 2495 [17]), 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 RFC1407 .....                        | 3  |
| 2 Overview .....                                      | 4  |
| 2.1 Use of ifTable for DS3 Layer .....                | 5  |
| 2.2 Usage Guidelines .....                            | 5  |
| 2.2.1 Usage of ifStackTable .....                     | 5  |
| 2.2.2 Usage of Channelization for DS3, DS1, DS0 ..... | 7  |
| 2.2.3 Usage of Channelization for DS3, DS2, DS1 ..... | 7  |
| 2.2.4 Usage of Loopbacks .....                        | 8  |
| 2.3 Objectives of this MIB Module .....               | 9  |
| 2.4 DS3/E3 Terminology .....                          | 9  |
| 2.4.1 Error Events .....                              | 10 |
| 2.4.2 Performance Parameters .....                    | 10 |

|  |    |
|--|----|
| 2.4.3 Performance Defects .....                                | 13 |
| 2.4.4 Other Terms .....  | 15 |
| 3 Object Definitions .....                                     | 15 |
| 3.1 The DS3/E3 Near End Group .....                            | 16 |
| 3.1.1 The DS3/E3 Configuration Table .....                     | 16 |
| 3.1.2 The DS3/E3 Current Table .....                           | 25 |
| 3.1.3 The DS3/E3 Interval Table .....                          | 28 |
| 3.1.4 The DS3/E3 Total .....                                   | 31 |
| 3.2 The DS3 Far End Group .....                                | 34 |
| 3.2.1 The DS3 Far End Configuration .....                      | 35 |
| 3.2.2 The DS3 Far End Current .....                            | 37 |
| 3.2.3 The DS3 Far End Interval Table .....                     | 39 |
| 3.2.4 The DS3 Far End Total .....                              | 41 |
| 3.3 The DS3/E3 Fractional Table .....                          | 43 |
| 3.4 The DS3 Trap Group .....                                   | 46 |
| 3.5 Conformance Groups .....                                   | 46 |
| 4 Appendix A - Use of dsx3IfIndex and dsx3LineIndex .....      | 51 |
| 5 Appendix B - The delay approach to Unavailable Seconds. .... | 54 |
| 6 Intellectual Property .....                                  | 56 |
| 7 Acknowledgments .....  | 56 |
| 8 References .....   | 56 |
| 9 Security Considerations .....                                | 58 |
| 10 Author's Address .....                                      | 59 |
| 11 Full Copyright Statement .....                              | 60 |

## 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 RFC1407

This MIB obsoletes RFC1407. The changes from RFC1407 are the following:

- (1) The Fractional Table has been deprecated
- (2) This document uses SMIV2
- (3) Values are given for ifTable and ifXTable
- (4) Example usage of ifStackTable is included
- (5) dsx3IfIndex has been deprecated
- (6) 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.
- (7) An inward loopback has been added.
- (8) Additional lineStatus bits have been added for Near End in Unavailable Signal State, Carrier Equipment Out of Service, DS@ Payload AIS, and DS@ Performance Threshold

- (9) A read-write line Length object has been added.
- (10) Added a lineStatus last change, trap and enabler.
- (11) Textual Conventions for statistics objects have been used.
- (12) A new object, dsx3LoopbackStatus, has been introduced to reflect the loopbacks established on a DS3/E3 interface and the source to the requests. dsx3LoopbackConfig continues to be the desired loopback state while dsx3LoopbackStatus reflects the actual state.
- (13) A dual loopback has been added to allow the setting of an inward loopback and a line loopback at the same time.
- (14) An object has been added to indicate whether or not this is a channelized DS3/E3.
- (15) A new object has been added to indicate which DS1 is to set for remote loopback.

## 2. Overview

These objects are used when the particular media being used to realize an interface is a DS3/E3 interface. At present, this applies to these values of the ifType variable in the Internet-standard MIB:

ds3 (30)

The DS3 definitions contained herein are based on the DS3 specifications in ANSI T1.102-1987, ANSI T1.107-1988, ANSI T1.107a-1990, and ANSI T1.404-1989 [8,9,9a,10]. The E3 definitions contained herein are based on the E3 specifications in CCITT G.751 [12].

## 2.1. Use of ifTable for DS3 Layer

Only the ifGeneralGroup needs to be supported.

| ifTable Object         | Use for DS3 Layer  |
|------------------------|--|
| ifIndex                | Interface index.   |
| ifDescr                | See interfaces MIB [5]   |
| ifType                 | ds3(30)  |
| ifSpeed                | Speed of line rate<br>DS3 - 44736000<br>E3 - 34368000  |
| ifPhysAddress          | The value of the Circuit Identifier.<br>If no Circuit Identifier has been assigned<br>this object should have an octet string<br>with zero length. |
| ifAdminStatus          | See interfaces MIB [5]   |
| ifOperStatus           | See interfaces MIB [5]   |
| ifLastChange           | See interfaces MIB [5]   |
| ifName                 | See interfaces MIB [5]   |
| ifLinkUpDownTrapEnable | Set to enabled(1).   |
| ifHighSpeed            | Speed of line in Mega-bits per second<br>(either 45 or 34)   |
| ifConnectorPresent     | Set to true(1) normally, except for<br>cases such as DS3/E3 over AAL1/ATM where<br>false(2) is appropriate   |

## 2.2. Usage Guidelines

### 2.2.1. Usage of ifStackTable

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 DS3 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 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. When this object is set to enabledDs1, 28 ifEntries of type DS1 will be created by the agent. If dsx3Channelization is set to disabled, then the DS1s are destroyed.

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. There will be 24 DS0s in the ifTable for each DS1. 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                    |

### 2.2.3. 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 | <code>dsx1Ds1ChannelNumber</code> | <code>dsx1ChanMappedIfIndex</code> |
|---------|-----------------------------------|------------------------------------|
| 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 | <code>dsx1Ds1ChannelNumber</code> | <code>dsx1ChanMappedIfIndex</code> |
|---------|-----------------------------------|------------------------------------|
| 2       | 1                                 | 9                                  |
| 2       | 2                                 | 10                                 |
| 2       | 3                                 | 11                                 |
| 2       | 4                                 | 12                                 |
| 3       | 1                                 | 13                                 |
| 3       | 2                                 | 14                                 |
| ...     |                                   |                                    |
| 8       | 4                                 | 36                                 |

#### 2.2.4. Usage of Loopbacks

This section discusses the behaviour of objects related to loopbacks.

The object `dsx3LoopbackConfig` 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 DS3 interface, the object `dsx3LoopbackStatus` 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 DS3/E3 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 DS3/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 DS3/E3 devices that are currently deployed.

### 2.4. DS3/E3 Terminology

The terminology used in this document to describe error conditions on a DS3 interface as monitored by a DS3 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 bipolar violation error event, for B3ZS(HDB3)-coded signals, is the occurrence of a pulse of the same polarity as the previous pulse without being part of the zero substitution code, B3ZS(HDB3). For B3ZS(HDB3)-coded signals, a bipolar violation error event may also include other error patterns such as: three(four) or more consecutive zeros and incorrect polarity. (See T1.231 section 7.1.1.1.1)

##### Excessive Zeros (EXZ) Error Event

An EXZ is the occurrence of any zero string length equal to or greater than 3 for B3ZS, or greater than 4 for HDB3. (See T1.231 section 7.1.1.1.2)

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

This parameter is a count of both BPVs and EXZs occurring over the accumulation period. An EXZ increments the LCV by one regardless of the length of the zero string. (Also known as CV-L. See T1.231 section 7.4.1.1)

##### P-bit Coding Violation (PCV) Error Event

For all DS3 applications, a coding violation error event is a P-bit Parity Error event. A P-bit Parity Error event is the occurrence of a received P-bit code on the DS3 M-frame that is not identical to the corresponding locally- calculated code. (See T1.231 section 7.1.1.2.1)

##### C-bit Coding Violation (CCV) Error Event

For C-bit Parity and SYNTRAN DS3 applications, this is the count of coding violations reported via the C-bits. For C-bit Parity, it is a count of CP-bit parity errors occurring in the accumulation interval. For SYNTRAN, it is a count of CRC-9 errors occurring in the accumulation interval. (See T1.231 section 7.1.1.2.2)

#### 2.4.2. 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.

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 CV occurred OR one or more LOS defects. (Also known as ES-L. See T1.231 section 7.4.1.2)

P-bit Errored Seconds (PES)

An PES is a second with one or more PCVs OR one or more Out of Frame defects OR a detected incoming AIS. This gauge is not incremented when UASs are counted. (Also known as ESP-P. See T1.231 section 7.4.2.2)

P-bit Severely Errored Seconds (PSES)

A PSES is a second with 44 or more PCVs OR one or more Out of Frame defects OR a detected incoming AIS. This gauge is not incremented when UASs are counted. (Also known as SESP-P. See T1.231 section 7.4.2.5)

C-bit Errored Seconds (CES)

An CES is a second with one or more CCVs OR one or more Out of Frame defects OR a detected incoming AIS. This count is only for the SYNTRAN and C-bit Parity DS3 applications. This gauge is not incremented when UASs are counted. (Also known as ESCP-P. See T1.231 section 7.4.2.2)

C-bit Severely Errored Seconds (CSES)

A CSES is a second with 44 or more CCVs OR one or more Out of Frame defects OR a detected incoming AIS. This count is only for the SYNTRAN and C-bit Parity DS3 applications. This gauge is not incremented when UASs are counted. (Also known as SESCP-P. See T1.231 section 7.4.2.5)

Severely Errored Framing Seconds (SEFS)

A SEFS is a second with one or more Out of Frame defects OR a detected incoming AIS. This item is not incremented during unavailable seconds. (Also known as SAS-P. See T1.231 section 7.4.2.6)

### Unavailable Seconds (UAS)

UAS are calculated by counting the number of seconds that the interface is unavailable. The DS3 interface is said to be unavailable from the onset of 10 contiguous PSESSs, 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 PSESSs, then the DS3 interface unavailability starts from the onset of these PSESSs. Once unavailable, and if no failure is present, the DS3 interface becomes available at the onset of 10 contiguous seconds with no PSESSs. Once unavailable, and if a failure is present, the DS3 interface becomes available at the onset of 10 contiguous seconds with no PSESSs, if the failure clearing time is less than or equal to 10 seconds. If the failure clearing time is more than 10 seconds, the DS3 interface becomes available at the onset of 10 contiguous seconds with no PSESSs, or the onset period leading to the successful clearing condition, whichever occurs later. With respect to the DS3 error counts, all counters are incremented while the DS3 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 PES, PSES, CES, and CSES 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, CCV, and SEFS 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 PES, CES, PCV, and CCV 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 PCV, CCV, PES, CES, PSES, CSEC, SEFS, and UAS counts for the PREVIOUS interval must be adjusted. In this case successive GETs of the affected dsx3IntervalPSESS and dsx3IntervalUASs 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 SESs. 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.

#### 2.4.3. Performance Defects

##### Failure States:

The Remote Alarm Indication (RAI) failure, in SYNTRAN applications, is declared after detecting the Yellow Alarm

Signal on the alarm channel. See ANSI T1.107a-1990 [9a]. The Remote Alarm Indication failure, in C-bit Parity DS3 applications, is declared as soon as the presence of either one or two alarm signals are detected on the Far End Alarm Channel. See [9]. The Remote Alarm Indication failure may also be declared after detecting the far-end SEF/AIS defect (aka yellow). The Remote Alarm Indication failure is cleared as soon as the presence of the any of the above alarms are removed.

Also, the incoming failure state is declared when a defect persists for at least 2-10 seconds. The defects are the following: Loss of Signal (LOS), an Out of Frame (OOF) or an incoming Alarm Indication Signal (AIS). The Failure State is cleared when the defect is absent for less than or equal to 20 seconds.

#### Far End SEF/AIS defect (aka yellow)

A Far End SEF/AIS defect is the occurrence of the two X-bits in a M-frame set to zero. The Far End SEF/AIS defect is terminated when the two X-bits in a M-frame are set to one. (Also known as SASCP-PFE. See T1.231 section 7.4.4.2.6)

#### Out of Frame (OOF) defect

A DS3 OOF defect is detected when any three or more errors in sixteen or fewer consecutive F-bits occur within a DS3 M-frame. An OOF defect may also be called a Severely Errored Frame (SEF) defect. An OOF defect is cleared when reframe occurs. A DS3 Loss of Frame (LOF) failure is declared when the DS3 OOF defect is consistent for 2 to 10 seconds. The DS3 OOF defect ends when reframe occurs. The DS3 LOF failure is cleared when the DS3 OOF defect is absent for 10 to 20 seconds. (See T1.231 section 7.1.2.2.1)

An E3 OOF defect is detected when four consecutive frame alignment signals have been incorrectly received in there predicted positions in an E3 signal. E3 frame alignment occurs when the presence of three consecutive frame alignment signals have been detected.

#### Loss of Signal (LOS) defect

The DS3 LOS defect is declared upon observing 175 +/- 75 contiguous pulse positions with no pulses of either positive or negative polarity. The DS3 LOS defect is terminated upon observing an average pulse density of at least 33% over a period of 175 +/- 75 contiguous pulse positions starting with the receipt of a pulse. (See T1.231 section 7.1.2.1.1)

#### Alarm Indication Signal (AIS) defect

The DS3 AIS is framed with "stuck stuffing." This implies that it has a valid M-subframe alignment bits, M-frame alignment bits, and P bits. The information bits are set to a 1010... sequence, starting with a one (1) after each M-subframe alignment bit, M-frame alignment bit, X bit, P bit, and C bit. The C bits are all set to zero giving what is called "stuck stuffing." The X bits are set to one. The DS3 AIS defect is declared after DS3 AIS is present in contiguous M-frames for a time equal to or greater than T, where  $0.2 \text{ ms} \leq T \leq 100 \text{ ms}$ . The DS3 AIS defect is terminated after AIS is absent in contiguous M-frames for a time equal to or greater than T. (See T1.231 section 7.1.2.2.3)

The E3 binary content of the AIS is nominally a continuous stream of ones. AIS detection and the application of consequent actions, should be completed within a time limit of 1 ms.

#### 2.4.4. 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

```
DS3-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                       | FROM IF-MIB           |
| PerfCurrentCount, PerfIntervalCount, |                       |
| PerfTotalCount                       | FROM PerfHist-TC-MIB; |

## ds3 MODULE-IDENTITY

LAST-UPDATED "9808012130Z"

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 is the MIB module that describes  
 DS3 and E3 interfaces objects."

::= { transmission 30 }

-- The DS3/E3 Near End Group

-- The DS3/E3 Near End Group consists of four tables:

-- DS3/E3 Configuration

-- DS3/E3 Current

-- DS3/E3 Interval

-- DS3/E3 Total

-- the DS3/E3 Configuration Table

## dsx3ConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx3ConfigEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The DS3/E3 Configuration table."

::= { ds3 5 }

## dsx3ConfigEntry OBJECT-TYPE

SYNTAX Dsx3ConfigEntry

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the DS3/E3 Configuration table."
INDEX { dsx3LineIndex }
 ::= { dsx3ConfigTable 1 }

```

```

Dsx3ConfigEntry ::=
SEQUENCE {
    dsx3LineIndex                InterfaceIndex,
    dsx3IfIndex                  InterfaceIndex,
    dsx3TimeElapsed               INTEGER,
    dsx3ValidIntervals            INTEGER,
    dsx3LineType                  INTEGER,
    dsx3LineCoding                INTEGER,
    dsx3SendCode                  INTEGER,
    dsx3CircuitIdentifier          DisplayString,
    dsx3LoopbackConfig            INTEGER,
    dsx3LineStatus                INTEGER,
    dsx3TransmitClockSource        INTEGER,
    dsx3InvalidIntervals          INTEGER,
    dsx3LineLength                INTEGER,
    dsx3LineStatusLastChange      TimeStamp,
    dsx3LineStatusChangeTrapEnable INTEGER,
    dsx3LoopbackStatus            INTEGER,
    dsx3Channelization            INTEGER,
    dsx3Ds1ForRemoteLoop          INTEGER
}

```

```

dsx3LineIndex 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.

```

Previously, this object was the identifier of a DS3/E3 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS3/E3 interface, it should have the same value as ifIndex. Otherwise, number the dsx3LineIndices 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."
 ::= { dsx3ConfigEntry 1 }

dsx3IfIndex 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)."
 ::= { dsx3ConfigEntry 2 }

dsx3TimeElapsed 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."

 ::= { dsx3ConfigEntry 3 }

dsx3ValidIntervals 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."
 ::= { dsx3ConfigEntry 4 }

dsx3LineType OBJECT-TYPE
    SYNTAX  INTEGER {
        dsx3Other(1),
```

```

        dsx3M23(2),
        dsx3SYNTRAN(3),
        dsx3CbitParity(4),
        dsx3ClearChannel(5),
        e3Other(6),
        e3Framed(7),
        e3Plcp(8)
    }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "This variable indicates the variety of DS3 C-bit
    or E3 application implementing this interface. The
    type of interface affects the interpretation of
    the usage and error statistics. The rate of DS3
    is 44.736 Mbps and E3 is 34.368 Mbps. The
    dsx3ClearChannel value means that the C-bits are
    not used except for sending/receiving AIS.
    The values, in sequence, describe:

    TITLE:                SPECIFICATION:
    dsx3M23                ANSI T1.107-1988 [9]
    dsx3SYNTRAN            ANSI T1.107-1988 [9]
    dsx3CbitParity         ANSI T1.107a-1990 [9a]
    dsx3ClearChannel       ANSI T1.102-1987 [8]
    e3Framed               CCITT G.751 [12]
    e3Plcp                 ETSI T/NA(91)18 [13]."
```

::= { dsx3ConfigEntry 5 }

```

dsx3LineCoding OBJECT-TYPE
    SYNTAX INTEGER {
        dsx3Other(1),
        dsx3B3ZS(2),
        e3HDB3(3)
    }
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.

    dsx3B3ZS and e3HDB3 refer to the use of specified
    patterns of normal bits and bipolar violations
    which are used to replace sequences of zero bits
    of a specified length."
::= { dsx3ConfigEntry 6 }
```

## dsx3SendCode OBJECT-TYPE

```

SYNTAX  INTEGER {
    dsx3SendNoCode(1),
    dsx3SendLineCode(2),
    dsx3SendPayloadCode(3),
    dsx3SendResetCode(4),
    dsx3SendDS1LoopCode(5),
    dsx3SendTestPattern(6)
}

```

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This variable indicates what type of code is being sent across the DS3/E3 interface by the device. (These are optional for E3 interfaces.) Setting this variable causes the interface to begin sending the code requested. The values mean:

dsx3SendNoCode  
sending looped or normal data

dsx3SendLineCode  
sending a request for a line loopback

dsx3SendPayloadCode  
sending a request for a payload loopback  
(i.e., all DS1/E1s in a DS3/E3 frame)

dsx3SendResetCode  
sending a loopback deactivation request

dsx3SendDS1LoopCode  
requesting to loopback a particular DS1/E1  
within a DS3/E3 frame. The DS1/E1 is  
indicated in dsx3Ds1ForRemoteLoop.

dsx3SendTestPattern  
sending a test pattern."

::= { dsx3ConfigEntry 7 }

## dsx3CircuitIdentifier 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."  
 ::= { dsx3ConfigEntry 8 }

dsx3LoopbackConfig OBJECT-TYPE

SYNTAX INTEGER {  
     dsx3NoLoop(1),  
     dsx3PayloadLoop(2),  
     dsx3LineLoop(3),  
     dsx3OtherLoop(4),  
     dsx3InwardLoop(5),  
     dsx3DualLoop(6)  
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This variable represents the desired loopback configuration of the DS3/E3 interface.

The values mean:

dsx3NoLoop

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.

dsx3PayloadLoop

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.

dsx3LineLoop

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

dsx3OtherLoop

Loopbacks that are not defined here.

dsx3InwardLoop

The sent signal at this interface is looped back through the device.

dsx3DualLoop

Both dsx1LineLoop and dsx1InwardLoop will be active simultaneously."

::= { dsx3ConfigEntry 9 }

## dsx3LineStatus OBJECT-TYPE

SYNTAX INTEGER (1..4095)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This variable indicates the Line Status of the interface. It contains loopback state information and failure state information. The dsx3LineStatus is a bit map represented as a sum, therefore, it can represent multiple failures and a loopback (see dsx3LoopbackConfig object for the type of loopback) simultaneously. The dsx3NoAlarm must be set if and only if no other flag is set.

If the dsx3loopbackState bit is set, the loopback in effect can be determined from the dsx3loopbackConfig object.

The various bit positions are:

|                            |                     |   |
|----------------------------|---------------------|---|
| 1                          | dsx3NoAlarm         | No alarm present                            |
| 2                          | dsx3RcvRAIFailure   | Receiving Yellow/Remote Alarm Indication    |
| 4                          | dsx3XmitRAIAlarm    | Transmitting Yellow/Remote Alarm Indication |
| 8                          | dsx3RcvAIS          | Receiving AIS failure state                 |
| 16                         | dsx3XmitAIS         | Transmitting AIS                            |
| 32                         | dsx3LOF             | Receiving LOF failure state                 |
| 64                         | dsx3LOS             | Receiving LOS failure state                 |
| 128                        | dsx3LoopbackState   | Looping the received signal                 |
| 256                        | dsx3RcvTestCode     | Receiving a Test Pattern                    |
| 512                        | dsx3OtherFailure    | any line status not defined here            |
| 1024                       | dsx3UnavailSigState | Near End in Unavailable Signal State        |
| 2048                       | dsx3NetEquipOOS     | Carrier Equipment Out of Service"           |
| ::= { dsx3ConfigEntry 10 } |                     |   |

## dsx3TransmitClockSource 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 that an external clock is attached to the box containing the interface.

throughTiming indicates that transmit clock is derived from the recovered receive clock of another DS3 interface."

::= { dsx3ConfigEntry 11 }

dsx3InvalidIntervals OBJECT-TYPE

SYNTAX INTEGER (0..96)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of intervals in the range from 0 to dsx3ValidIntervals 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)."

::= { dsx3ConfigEntry 12 }

dsx3LineLength OBJECT-TYPE

SYNTAX INTEGER (0..64000)

UNITS "meters"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The length of the ds3 line in meters. This object provides information for line build out circuitry if it exists and can use this object to adjust the line build out."

::= { dsx3ConfigEntry 13 }

dsx3LineStatusLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of MIB II's sysUpTime object at the time this DS3/E3 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."

::= { dsx3ConfigEntry 14 }

dsx3LineStatusChangeTrapEnable OBJECT-TYPE

```

SYNTAX      INTEGER {
                enabled(1),
                disabled(2)
            }
MAX-ACCESS   read-write
STATUS       current
DESCRIPTION
    "Indicates whether dsx3LineStatusChange traps
    should be generated for this interface."
DEFVAL { disabled }
 ::= { dsx3ConfigEntry 15 }

```

dsx3LoopbackStatus OBJECT-TYPE

```

SYNTAX      INTEGER (1..127)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION

```

"This variable represents the current state of the loopback on the DS3 interface. It contains information about loopbacks established by a manager and remotely from the far end.

The dsx3LoopbackStatus is a bit map represented as a sum, therefore is can represent multiple loopbacks simultaneously.

The various bit positions are:

- 1 dsx3NoLoopback
- 2 dsx3NearEndPayloadLoopback
- 4 dsx3NearEndLineLoopback
- 8 dsx3NearEndOtherLoopback
- 16 dsx3NearEndInwardLoopback
- 32 dsx3FarEndPayloadLoopback
- 64 dsx3FarEndLineLoopback"

```
 ::= { dsx3ConfigEntry 16 }
```

dsx3Channelization OBJECT-TYPE

```

SYNTAX      INTEGER {
                disabled(1),
                enabledDs1(2),
                enabledDs2(3)
            }
MAX-ACCESS   read-write
STATUS       current
DESCRIPTION

```

"Indicates whether this ds3/e3 is channelized or unchannelized. The value of enabledDs1 indicates

that this is a DS3 channelized into DS1s. The value of enabledDs3 indicated that this is a DS3 channelized into DS2s. Setting this object will cause the creation or deletion of DS2 or DS1 entries in the ifTable. "

```
::= { dsx3ConfigEntry 17 }
```

```
dsx3Ds1ForRemoteLoop OBJECT-TYPE
```

```
SYNTAX      INTEGER (0..29)
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"Indicates which ds1/el on this ds3/e3 will be indicated in the remote ds1 loopback request. A value of 0 means no DS1 will be looped. A value of 29 means all ds1s/els will be looped."

```
::= { dsx3ConfigEntry 18 }
```

```
-- the DS3/E3 Current Table
```

```
dsx3CurrentTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF Dsx3CurrentEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"The DS3/E3 current table contains various statistics being collected for the current 15 minute interval."

```
::= { ds3 6 }
```

```
dsx3CurrentEntry OBJECT-TYPE
```

```
SYNTAX      Dsx3CurrentEntry
```

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

```
STATUS      current
```

```
DESCRIPTION
```

"An entry in the DS3/E3 Current table."

```
INDEX      { dsx3CurrentIndex }
```

```
::= { dsx3CurrentTable 1 }
```

```
Dsx3CurrentEntry ::=
```

```
SEQUENCE {
```

```
    dsx3CurrentIndex
```

```
    dsx3CurrentPESS
```

```
    dsx3CurrentPSESS
```

```
    dsx3CurrentSEFSS
```

```
    InterfaceIndex,
```

```
    PerfCurrentCount,
```

```
    PerfCurrentCount,
```

```
    PerfCurrentCount,
```

|                 |                   |
|-----------------|-------------------|
| dsx3CurrentUASS | PerfCurrentCount, |
| dsx3CurrentLCVs | PerfCurrentCount, |
| dsx3CurrentPCVs | PerfCurrentCount, |
| dsx3CurrentLESS | PerfCurrentCount, |
| dsx3CurrentCCVs | PerfCurrentCount, |
| dsx3CurrentCESS | PerfCurrentCount, |
| dsx3CurrentCESs | PerfCurrentCount  |

}

## dsx3CurrentIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index value which uniquely identifies the DS3/E3 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 dsx3LineIndex object instance."

::= { dsx3CurrentEntry 1 }

## dsx3CurrentPESS OBJECT-TYPE

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of P-bit Errored Seconds."

::= { dsx3CurrentEntry 2 }

## dsx3CurrentPSESS OBJECT-TYPE

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of P-bit Severely Errored Seconds."

::= { dsx3CurrentEntry 3 }

## dsx3CurrentSEFSS OBJECT-TYPE

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Severely Errored Framing Seconds."

::= { dsx3CurrentEntry 4 }

```
dsx3CurrentUASS OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of
        Unavailable Seconds."
    ::= { dsx3CurrentEntry 5 }

dsx3CurrentLCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Line
        Coding Violations."
    ::= { dsx3CurrentEntry 6 }

dsx3CurrentPCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of P-bit
        Coding Violations."
    ::= { dsx3CurrentEntry 7 }

dsx3CurrentLESS OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Errored Seconds."
    ::= { dsx3CurrentEntry 8 }

dsx3CurrentCCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of C-bit Coding Violations."
    ::= { dsx3CurrentEntry 9 }

dsx3CurrentCESS OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
```

```

        "The number of C-bit Errored Seconds."
 ::= { dsx3CurrentEntry 10 }

dsx3CurrentCSESS OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of C-bit Severely Errored Seconds."
 ::= { dsx3CurrentEntry 11 }

-- the DS3/E3 Interval Table

dsx3IntervalTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx3IntervalEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The DS3/E3 Interval Table contains various
        statistics collected by each DS3/E3 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 dsx3IntervalNumber)
        and for one specific interface (identified by
        dsx3IntervalIndex)."
```

```

 ::= { ds3 7 }

dsx3IntervalEntry OBJECT-TYPE
    SYNTAX  Dsx3IntervalEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An entry in the DS3/E3 Interval table."
    INDEX   { dsx3IntervalIndex, dsx3IntervalNumber }
 ::= { dsx3IntervalTable 1 }

Dsx3IntervalEntry ::=
    SEQUENCE {
        dsx3IntervalIndex          InterfaceIndex,
        dsx3IntervalNumber         INTEGER,
        dsx3IntervalPESs          PerfIntervalCount,
        dsx3IntervalPSESSs        PerfIntervalCount,
        dsx3IntervalSEFSSs        PerfIntervalCount,
        dsx3IntervalUASSs         PerfIntervalCount,
        dsx3IntervalLCVs          PerfIntervalCount,
        dsx3IntervalPCVs          PerfIntervalCount,
```

```

        dsx3IntervalLESSs      PerfIntervalCount,
        dsx3IntervalCCVs       PerfIntervalCount,
        dsx3IntervalCESSs      PerfIntervalCount,
        dsx3IntervalCSESSs     PerfIntervalCount,
        dsx3IntervalValidData   TruthValue
    }

dsx3IntervalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the
        DS3/E3 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
        dsx3LineIndex object instance."
    ::= { dsx3IntervalEntry 1 }

dsx3IntervalNumber 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."
    ::= { dsx3IntervalEntry 2 }

dsx3IntervalPESs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of P-bit
        Errored Seconds."
    ::= { dsx3IntervalEntry 3 }

dsx3IntervalPSESSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of P-bit
        Severely Errored Seconds."
    ::= { dsx3IntervalEntry 4 }

```

dsx3IntervalSEFSs OBJECT-TYPE  
SYNTAX PerfIntervalCount  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The counter associated with the number of  
        Severely Errored Framing Seconds."  
 ::= { dsx3IntervalEntry 5 }

dsx3IntervalUASs OBJECT-TYPE  
SYNTAX PerfIntervalCount  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The counter associated with the number of  
        Unavailable Seconds. This object may decrease if  
        the occurrence of unavailable seconds occurs across  
        an interval boundary."  
 ::= { dsx3IntervalEntry 6 }

dsx3IntervalLCVs OBJECT-TYPE  
SYNTAX PerfIntervalCount  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The counter associated with the number of Line  
        Coding Violations."  
 ::= { dsx3IntervalEntry 7 }

dsx3IntervalPCVs OBJECT-TYPE  
SYNTAX PerfIntervalCount  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The counter associated with the number of P-bit  
        Coding Violations."  
 ::= { dsx3IntervalEntry 8 }

dsx3IntervalLESSs OBJECT-TYPE  
SYNTAX PerfIntervalCount  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The number of Line Errored Seconds (BPVs or  
        illegal zero sequences)."  
 ::= { dsx3IntervalEntry 9 }

dsx3IntervalCCVs OBJECT-TYPE

```
SYNTAX PerfIntervalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of C-bit Coding Violations."
 ::= { dsx3IntervalEntry 10 }

dsx3IntervalCESS OBJECT-TYPE
SYNTAX PerfIntervalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of C-bit Errored Seconds."
 ::= { dsx3IntervalEntry 11 }

dsx3IntervalCSESS OBJECT-TYPE
SYNTAX PerfIntervalCount
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of C-bit Severely Errored Seconds."
 ::= { dsx3IntervalEntry 12 }

dsx3IntervalValidData OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This variable indicates if the data for this
     interval is valid."
 ::= { dsx3IntervalEntry 13 }

-- the DS3/E3 Total

dsx3TotalTable OBJECT-TYPE
SYNTAX SEQUENCE OF Dsx3TotalEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "The DS3/E3 Total Table contains the cumulative
     sum of the various statistics for the 24 hour
     period preceding the current interval."
 ::= { ds3 8 }

dsx3TotalEntry OBJECT-TYPE
SYNTAX Dsx3TotalEntry
MAX-ACCESS not-accessible
STATUS current
```

## DESCRIPTION

"An entry in the DS3/E3 Total table."

INDEX { dsx3TotalIndex }  
 ::= { dsx3TotalTable 1 }

Dsx3TotalEntry ::=

```
SEQUENCE {
    dsx3TotalIndex      InterfaceIndex,
    dsx3TotalPESs      PerfTotalCount,
    dsx3TotalPSESSs    PerfTotalCount,
    dsx3TotalSEFSSs    PerfTotalCount,
    dsx3TotalUASSs     PerfTotalCount,
    dsx3TotalLCVs      PerfTotalCount,
    dsx3TotalPCVs      PerfTotalCount,
    dsx3TotalLESSs     PerfTotalCount,
    dsx3TotalCCVs      PerfTotalCount,
    dsx3TotalCESSs     PerfTotalCount,
    dsx3TotalCSESSs    PerfTotalCount
}
```

dsx3TotalIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index value which uniquely identifies the DS3/E3 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 dsx3LineIndex object instance."

::= { dsx3TotalEntry 1 }

dsx3TotalPESs OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of P-bit Errored Seconds, encountered by a DS3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

::= { dsx3TotalEntry 2 }

dsx3TotalPSESSs OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The counter associated with the number of P-bit Severely Errored Seconds, encountered by a DS3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

::= { dsx3TotalEntry 3 }

## dsx3TotalSEFSS OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The counter associated with the number of Severely Errored Framing Seconds, encountered by a DS3/E3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

::= { dsx3TotalEntry 4 }

## dsx3TotalUASS OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The counter associated with the number of Unavailable Seconds, encountered by a DS3 interface in the previous 24 hour interval.

Invalid 15 minute intervals count as 0."

::= { dsx3TotalEntry 5 }

## dsx3TotalLCVs OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The counter associated with the number of Line Coding Violations encountered by a DS3/E3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

::= { dsx3TotalEntry 6 }

## dsx3TotalPCVs OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The counter associated with the number of P-bit Coding Violations, encountered by a DS3 interface

```

        in the previous 24 hour interval. Invalid 15
        minute intervals count as 0."
 ::= { dsx3TotalEntry 7 }

dsx3TotalLESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of Line Errored Seconds (BPVs or
        illegal zero sequences) encountered by a DS3/E3
        interface in the previous 24 hour interval.
        Invalid 15 minute intervals count as 0."
 ::= { dsx3TotalEntry 8 }

dsx3TotalCCVs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of C-bit Coding Violations encountered
        by a DS3 interface in the previous 24 hour
        interval. Invalid 15 minute intervals count as 0."
 ::= { dsx3TotalEntry 9 }

dsx3TotalCESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of C-bit Errored Seconds encountered
        by a DS3 interface in the previous 24 hour
        interval. Invalid 15 minute intervals count as 0."
 ::= { dsx3TotalEntry 10 }

dsx3TotalCSESSs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of C-bit Severely Errored Seconds
        encountered by a DS3 interface in the previous 24
        hour interval. Invalid 15 minute intervals count
        as 0."
 ::= { dsx3TotalEntry 11 }

-- The DS3 Far End Group
```

```

-- The DS3 Far End Group consists of four tables :
--   DS3 Far End Configuration
--   DS3 Far End Current
--   DS3 Far End Interval
--   DS3 Far End Total

-- The DS3 Far End Configuration Table

dsx3FarEndConfigTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx3FarEndConfigEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "The DS3 Far End Configuration Table contains
        configuration information reported in the C-bits
        from the remote end."
    ::= { ds3 9 }

dsx3FarEndConfigEntry OBJECT-TYPE
    SYNTAX  Dsx3FarEndConfigEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "An entry in the DS3 Far End Configuration table."
    INDEX   { dsx3FarEndLineIndex }
    ::= { dsx3FarEndConfigTable 1 }

Dsx3FarEndConfigEntry ::=
    SEQUENCE {
        dsx3FarEndLineIndex          InterfaceIndex,
        dsx3FarEndEquipCode           DisplayString,
        dsx3FarEndLocationIDCode      DisplayString,
        dsx3FarEndFrameIDCode         DisplayString,
        dsx3FarEndUnitCode            DisplayString,
        dsx3FarEndFacilityIDCode      DisplayString
    }

dsx3FarEndLineIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS3
        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 dsx3LineIndex object instance."

```

```
 ::= { dsx3FarEndConfigEntry 1 }

dsx3FarEndEquipCode OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..10))
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "This is the Far End Equipment Identification code
        that describes the specific piece of equipment.
        It is sent within the Path Identification
        Message."
    ::= { dsx3FarEndConfigEntry 2 }

dsx3FarEndLocationIDCode OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..11))
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "This is the Far End Location Identification code
        that describes the specific location of the
        equipment. It is sent within the Path
        Identification Message."
    ::= { dsx3FarEndConfigEntry 3 }

dsx3FarEndFrameIDCode OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..10))
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "This is the Far End Frame Identification code
        that identifies where the equipment is located
        within a building at a given location. It is sent
        within the Path Identification Message."
    ::= { dsx3FarEndConfigEntry 4 }

dsx3FarEndUnitCode OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..6))
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "This is the Far End code that identifies the
        equipment location within a bay. It is sent
        within the Path Identification Message."
    ::= { dsx3FarEndConfigEntry 5 }

dsx3FarEndFacilityIDCode OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..38))
    MAX-ACCESS  read-write
```

```

STATUS    current
DESCRIPTION
    "This code identifies a specific Far End DS3 path.
    It is sent within the Path Identification
    Message."
 ::= { dsx3FarEndConfigEntry 6 }

```

-- The DS3 Far End Current

```

dsx3FarEndCurrentTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx3FarEndCurrentEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "The DS3 Far End Current table contains various
        statistics being collected for the current 15
        minute interval. The statistics are collected
        from the far end block error code within the C-
        bits."
    ::= { ds3 10 }

```

```

dsx3FarEndCurrentEntry OBJECT-TYPE
    SYNTAX  Dsx3FarEndCurrentEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
        "An entry in the DS3 Far End Current table."
    INDEX   { dsx3FarEndCurrentIndex }
    ::= { dsx3FarEndCurrentTable 1 }

```

```

Dsx3FarEndCurrentEntry ::=
    SEQUENCE {
        dsx3FarEndCurrentIndex      InterfaceIndex,
        dsx3FarEndTimeElapsed        INTEGER,
        dsx3FarEndValidIntervals    INTEGER,
        dsx3FarEndCurrentCESS       PerfCurrentCount,
        dsx3FarEndCurrentCESs       PerfCurrentCount,
        dsx3FarEndCurrentCCVs       PerfCurrentCount,
        dsx3FarEndCurrentUASS       PerfCurrentCount,
        dsx3FarEndInvalidIntervals  INTEGER
    }

```

```

dsx3FarEndCurrentIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
        "The index value which uniquely identifies the DS3

```

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 dsx3LineIndex."

::= { dsx3FarEndCurrentEntry 1 }

dsx3FarEndTimeElapsed 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."

::= { dsx3FarEndCurrentEntry 2 }

dsx3FarEndValidIntervals 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."

::= { dsx3FarEndCurrentEntry 3 }

dsx3FarEndCurrentCESSs OBJECT-TYPE

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Far Far End C-bit Errored Seconds."

::= { dsx3FarEndCurrentEntry 4 }

dsx3FarEndCurrentCSESSs OBJECT-TYPE

SYNTAX PerfCurrentCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Far End C-bit Severely Errored Seconds."

```

 ::= { dsx3FarEndCurrentEntry 5 }

dsx3FarEndCurrentCCVs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
        C-bit Coding Violations reported via the far end
        block error count."
    ::= { dsx3FarEndCurrentEntry 6 }

dsx3FarEndCurrentUASSs OBJECT-TYPE
    SYNTAX  PerfCurrentCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
        unavailable seconds."
    ::= { dsx3FarEndCurrentEntry 7 }

dsx3FarEndInvalidIntervals OBJECT-TYPE
    SYNTAX  INTEGER (0..96)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of intervals in the range from 0 to
        dsx3FarEndValidIntervals 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)."
    ::= { dsx3FarEndCurrentEntry 8 }

-- The DS3 Far End Interval Table

dsx3FarEndIntervalTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx3FarEndIntervalEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The DS3 Far End Interval Table contains various
        statistics collected by each DS3 interface over
        the previous 24 hours of operation. The past 24
        hours are broken into 96 completed 15 minute
        intervals."
    ::= { ds3 11 }

dsx3FarEndIntervalEntry OBJECT-TYPE

```

```

SYNTAX  Dsx3FarEndIntervalEntry
MAX-ACCESS  not-accessible
STATUS  current
DESCRIPTION
    "An entry in the DS3 Far End Interval table."
INDEX     { dsx3FarEndIntervalIndex,
             dsx3FarEndIntervalNumber }
 ::= { dsx3FarEndIntervalTable 1 }

```

```

Dsx3FarEndIntervalEntry ::=
    SEQUENCE {
        dsx3FarEndIntervalIndex      InterfaceIndex,
        dsx3FarEndIntervalNumber      INTEGER,
        dsx3FarEndIntervalCESS        PerfIntervalCount,
        dsx3FarEndIntervalCSESS       PerfIntervalCount,
        dsx3FarEndIntervalCCVs        PerfIntervalCount,
        dsx3FarEndIntervalUASS         PerfIntervalCount,
        dsx3FarEndIntervalValidData    TruthValue
    }

```

```

dsx3FarEndIntervalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS3
        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 dsx3LineIndex."
    ::= { dsx3FarEndIntervalEntry 1 }

```

```

dsx3FarEndIntervalNumber 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."
    ::= { dsx3FarEndIntervalEntry 2 }

```

```

dsx3FarEndIntervalCESS OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION

```

```

        "The counter associated with the number of Far End
        C-bit Errored Seconds encountered by a DS3
        interface in one of the previous 96, individual 15
        minute, intervals. In the case where the agent is
        a proxy and data is not available, return
        noSuchInstance."
 ::= { dsx3FarEndIntervalEntry 3 }

dsx3FarEndIntervalCSESSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
        C-bit Severely Errored Seconds."
 ::= { dsx3FarEndIntervalEntry 4 }

dsx3FarEndIntervalCCVs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
        C-bit Coding Violations reported via the far end
        block error count."
 ::= { dsx3FarEndIntervalEntry 5 }

dsx3FarEndIntervalUASSs OBJECT-TYPE
    SYNTAX  PerfIntervalCount
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
        unavailable seconds."
 ::= { dsx3FarEndIntervalEntry 6 }

dsx3FarEndIntervalValidData OBJECT-TYPE
    SYNTAX  TruthValue
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "This variable indicates if the data for this
        interval is valid."
 ::= { dsx3FarEndIntervalEntry 7 }

-- The DS3 Far End Total

```

```

dsx3FarEndTotalTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF Dsx3FarEndTotalEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "The DS3 Far End Total Table contains the
         cumulative sum of the various statistics for the
         24 hour period preceding the current interval."
    ::= { ds3 12 }

dsx3FarEndTotalEntry OBJECT-TYPE
    SYNTAX  Dsx3FarEndTotalEntry
    MAX-ACCESS not-accessible
    STATUS  current
    DESCRIPTION
        "An entry in the DS3 Far End Total table."
    INDEX   { dsx3FarEndTotalIndex }
    ::= { dsx3FarEndTotalTable 1 }

Dsx3FarEndTotalEntry ::=
    SEQUENCE {
        dsx3FarEndTotalIndex      InterfaceIndex,
        dsx3FarEndTotalCESs       PerfTotalCount,
        dsx3FarEndTotalCSESSs     PerfTotalCount,
        dsx3FarEndTotalCCVs       PerfTotalCount,
        dsx3FarEndTotalUASS       PerfTotalCount
    }

dsx3FarEndTotalIndex OBJECT-TYPE
    SYNTAX  InterfaceIndex
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "The index value which uniquely identifies the DS3
         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 dsx3LineIndex."
    ::= { dsx3FarEndTotalEntry 1 }

dsx3FarEndTotalCESs OBJECT-TYPE
    SYNTAX  PerfTotalCount
    MAX-ACCESS read-only
    STATUS  current
    DESCRIPTION
        "The counter associated with the number of Far End
         C-bit Errored Seconds encountered by a DS3
         interface in the previous 24 hour interval."

```

```

        Invalid 15 minute intervals count as 0."
 ::= { dsx3FarEndTotalEntry 2 }

```

dsx3FarEndTotalCSESS OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Far End C-bit Severely Errored Seconds encountered by a DS3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

```

 ::= { dsx3FarEndTotalEntry 3 }

```

dsx3FarEndTotalCCVs OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Far End C-bit Coding Violations reported via the far end block error count encountered by a DS3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

```

 ::= { dsx3FarEndTotalEntry 4 }

```

dsx3FarEndTotalUASS OBJECT-TYPE

SYNTAX PerfTotalCount

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The counter associated with the number of Far End unavailable seconds encountered by a DS3 interface in the previous 24 hour interval. Invalid 15 minute intervals count as 0."

```

 ::= { dsx3FarEndTotalEntry 5 }

```

-- the DS3/E3 Fractional Table

-- This table is deprecated.

dsx3FracTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dsx3FracEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This table is deprecated in favour of using

ifStackTable.

Implementation of this table was optional. It was designed for those systems dividing a DS3/E3 into channels containing different data streams that are of local interest.

The DS3/E3 fractional table identifies which DS3/E3 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 a DS3 device with 4 high speed links carrying router traffic, a feed for voice, a feed for video, and a synchronous channel for a non-routed protocol. We might describe the allocation of channels, in the dsx3FracTable, as follows:

```
dsx3FracIfIndex.2. 1 = 3   dsx3FracIfIndex.2.15 = 4
dsx3FracIfIndex.2. 2 = 3   dsx3FracIfIndex.2.16 = 6
dsx3FracIfIndex.2. 3 = 3   dsx3FracIfIndex.2.17 = 6
dsx3FracIfIndex.2. 4 = 3   dsx3FracIfIndex.2.18 = 6
dsx3FracIfIndex.2. 5 = 3   dsx3FracIfIndex.2.19 = 6
dsx3FracIfIndex.2. 6 = 3   dsx3FracIfIndex.2.20 = 6
dsx3FracIfIndex.2. 7 = 4   dsx3FracIfIndex.2.21 = 6
dsx3FracIfIndex.2. 8 = 4   dsx3FracIfIndex.2.22 = 6
dsx3FracIfIndex.2. 9 = 4   dsx3FracIfIndex.2.23 = 6
dsx3FracIfIndex.2.10 = 4   dsx3FracIfIndex.2.24 = 6
dsx3FracIfIndex.2.11 = 4   dsx3FracIfIndex.2.25 = 6
dsx3FracIfIndex.2.12 = 5   dsx3FracIfIndex.2.26 = 6
dsx3FracIfIndex.2.13 = 5   dsx3FracIfIndex.2.27 = 6
dsx3FracIfIndex.2.14 = 5   dsx3FracIfIndex.2.28 = 6
```

For dsx3M23, dsx3 SYNTRAN, dsx3CbitParity, and dsx3ClearChannel there are 28 legal channels, numbered 1 through 28.

For e3Framed there are 16 legal channels, numbered 1 through 16. The channels (1..16) correspond directly to the equivalently numbered time-slots."

```
::= { ds3 13 }
```

dsx3FracEntry OBJECT-TYPE

SYNTAX Dsx3FracEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"An entry in the DS3 Fractional table."

```

INDEX    { dsx3FracIndex, dsx3FracNumber }
 ::= { dsx3FracTable 1 }

```

```

Dsx3FracEntry ::=
    SEQUENCE {
        dsx3FracIndex      INTEGER,
        dsx3FracNumber     INTEGER,
        dsx3FracIfIndex    INTEGER
    }

```

```

dsx3FracIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..'7fffffff'h)
    MAX-ACCESS  read-only
    STATUS  deprecated
    DESCRIPTION
        "The index value which uniquely identifies the
        DS3 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 dsx3LineIndex
        object instance."
    ::= { dsx3FracEntry 1 }

```

```

dsx3FracNumber OBJECT-TYPE
    SYNTAX  INTEGER (1..31)
    MAX-ACCESS  read-only
    STATUS  deprecated
    DESCRIPTION
        "The channel number for this entry."
    ::= { dsx3FracEntry 2 }

```

```

dsx3FracIfIndex 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."
    ::= { dsx3FracEntry 3 }

```

```
-- Ds3 TRAPS
```

```
ds3Traps OBJECT IDENTIFIER ::= { ds3 15 }
```

```
dsx3LineStatusChange NOTIFICATION-TYPE
```

```
  OBJECTS { dsx3LineStatus,
            dsx3LineStatusLastChange }
```

```
  STATUS current
```

```
  DESCRIPTION
```

```
    "A dsx3LineStatusChange trap is sent when the
    value of an instance of dsx3LineStatus changes. It
    can be utilized by an NMS to trigger polls. When
    the line status change results in a lower level
    line status change (i.e. ds1), then no traps for
    the lower level are sent."
```

```
    ::= { ds3Traps 0 1 }
```

```
-- conformance information
```

```
ds3Conformance OBJECT IDENTIFIER ::= { ds3 14 }
```

```
ds3Groups          OBJECT IDENTIFIER ::= {
ds3Conformance 1 } ds3Compliances OBJECT
IDENTIFIER ::= { ds3Conformance 2 }
```

```
-- compliance statements
```

```
ds3Compliance MODULE-COMPLIANCE
```

```
  STATUS current
```

```
  DESCRIPTION
```

```
    "The compliance statement for DS3/E3
    interfaces."
```

```
MODULE -- this module
```

```
  MANDATORY-GROUPS { ds3NearEndConfigGroup,
                     ds3NearEndStatisticsGroup }
```

```
GROUP ds3FarEndGroup
```

```
DESCRIPTION
```

```
  "Implementation of this group is optional for all
  systems that attach to a DS3 Interface. However,
  only C-bit Parity and SYNTRAN DS3 applications
  have the capability (option) of providing this
  information."
```

GROUP ds3NearEndOptionalConfigGroup  
DESCRIPTION  
"Implementation of this group is optional for all systems that attach to a DS3 interface."

OBJECT dsx3LineType  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the line type is not required."

OBJECT dsx3LineCoding  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the line coding is not required."

OBJECT dsx3SendCode  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the send code is not required."

OBJECT dsx3LoopbackConfig  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for loopbacks is not required."

OBJECT dsx3TransmitClockSource  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the transmit clock source is not required."

OBJECT dsx3LineLength  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the line length is not required."

OBJECT dsx3Channelization  
MIN-ACCESS read-only  
DESCRIPTION  
"Write access for the channelization is not required."

::= { ds3Compliances 1 }

-- units of conformance

```
ds3NearEndConfigGroup OBJECT-GROUP
    OBJECTS { dsx3LineIndex,
               dsx3TimeElapsed,
               dsx3ValidIntervals,
               dsx3LineType,
               dsx3LineCoding,
               dsx3SendCode,
               dsx3CircuitIdentifier,
               dsx3LoopbackConfig,
               dsx3LineStatus,
               dsx3TransmitClockSource,
               dsx3InvalidIntervals,
               dsx3LineLength,
               dsx3LoopbackStatus,
               dsx3Channelization,
               dsx3DslForRemoteLoop }
    STATUS current
    DESCRIPTION
        "A collection of objects providing configuration
        information applicable to all DS3/E3 interfaces."
    ::= { ds3Groups 1 }

ds3NearEndStatisticsGroup OBJECT-GROUP
    OBJECTS { dsx3CurrentIndex,
               dsx3CurrentPESs,
               dsx3CurrentPSESSs,
               dsx3CurrentSEFSs,
               dsx3CurrentUASs,
               dsx3CurrentLCVs,
               dsx3CurrentPCVs,
               dsx3CurrentLESSs,
               dsx3CurrentCCVs,
               dsx3CurrentCESSs,
               dsx3CurrentCSESSs,
               dsx3IntervalIndex,
               dsx3IntervalNumber,
               dsx3IntervalPESs,
               dsx3IntervalPSESSs,
               dsx3IntervalSEFSs,
               dsx3IntervalUASs,
               dsx3IntervalLCVs,
               dsx3IntervalPCVs,
               dsx3IntervalLESSs,
               dsx3IntervalCCVs,
               dsx3IntervalCESSs,
               dsx3IntervalCSESSs,
               dsx3IntervalValidData,
               dsx3TotalIndex,
```

```

        dsx3TotalPESs,
        dsx3TotalPSEsEs,
        dsx3TotalSEFSs,
        dsx3TotalUASs,
        dsx3TotalLCVs,
        dsx3TotalPCVs,
        dsx3TotalLEsEs,
        dsx3TotalCCVs,
        dsx3TotalCESs,
        dsx3TotalCSEsEs }
STATUS    current
DESCRIPTION
    "A collection of objects providing statistics
    information applicable to all DS3/E3 interfaces."
 ::= { ds3Groups 2 }

ds3FarEndGroup  OBJECT-GROUP
    OBJECTS { dsx3FarEndLineIndex,
               dsx3FarEndEquipCode,
               dsx3FarEndLocationIDCode,
               dsx3FarEndFrameIDCode,
               dsx3FarEndUnitCode,
               dsx3FarEndFacilityIDCode,
               dsx3FarEndCurrentIndex,
               dsx3FarEndTimeElapsed,
               dsx3FarEndValidIntervals,
               dsx3FarEndCurrentCESs,
               dsx3FarEndCurrentCSEsEs,
               dsx3FarEndCurrentCCVs,
               dsx3FarEndCurrentUASs,
               dsx3FarEndInvalidIntervals,
               dsx3FarEndIntervalIndex,
               dsx3FarEndIntervalNumber,
               dsx3FarEndIntervalCESs,
               dsx3FarEndIntervalCSEsEs,
               dsx3FarEndIntervalCCVs,
               dsx3FarEndIntervalUASs,
               dsx3FarEndIntervalValidData,
               dsx3FarEndTotalIndex,
               dsx3FarEndTotalCESs,
               dsx3FarEndTotalCSEsEs,
               dsx3FarEndTotalCCVs,
               dsx3FarEndTotalUASs }
STATUS    current
DESCRIPTION
    "A collection of objects providing remote
    configuration and statistics information
    applicable to C-bit Parity and SYNTRAN DS3

```

```
        interfaces."
 ::= { ds3Groups 3 }

ds3DeprecatedGroup OBJECT-GROUP
  OBJECTS { dsx3IfIndex,
            dsx3FracIndex,
            dsx3FracNumber,
            dsx3FracIfIndex }
  STATUS deprecated
  DESCRIPTION
    "A collection of obsolete objects that may be
    implemented for backwards compatibility."
 ::= { ds3Groups 4 }

ds3NearEndOptionalConfigGroup OBJECT-GROUP
  OBJECTS { dsx3LineStatusLastChange,
            dsx3LineStatusChangeTrapEnable }

  STATUS current
  DESCRIPTION
    "A collection of objects that may be implemented
    on DS3/E3 interfaces."
 ::= { ds3Groups 5 }

ds3NearEndOptionalTrapGroup NOTIFICATION-GROUP
  NOTIFICATIONS { dsx3LineStatusChange }
  STATUS current
  DESCRIPTION
    "A collection of notifications that may be
    implemented on DS3/E3 interfaces."
 ::= { ds3Groups 6 }

END
```

#### 4. Appendix A - Use of dsx3IfIndex and dsx3LineIndex

This Appendix exists to document the previous use of dsx3IfIndex and dsx3LineIndex and to clarify the relationship of dsx3LineIndex as defined in rfc1407 with the dsx3LineIndex 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.

[Old Definition]: "this object is the identifier of a DS3/E3 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS3/E3 interface, it should have the same value as ifIndex. Otherwise, number the dsx3LineIndices 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, my understanding was that 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 rfc1407:

Different physical configurations for the support of SNMP with DS3/E3 equipment exist. To accommodate these scenarios, two different indices for DS3/E3 interfaces are introduced in this MIB. These indices are dsx3IfIndex and dsx3LineIndex.

External interface scenario: the SNMP Agent represents all managed DS3/E3 lines as external interfaces (for example, an Agent residing on the device supporting DS3/E3 interfaces directly):

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

ifIndex=dsx3IfIndex=dsx3LineIndex for all interfaces.

The dsx3IfIndex column of the DS3/E3 Configuration table relates each DS3/E3 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 DS3/E3 interfaces (e.g., a router). The Agent represents both the host and the DS3/E3 device. The index dsx3LineIndex is used to not only represent the DS3/E3 interfaces external from the host/DS3/E3-device combination, but also the DS3/E3 interfaces connecting the host and the DS3/E3 device. The index dsx3IfIndex 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:

|                                      |                            |             |         |                  |                  |
|--------------------------------------|----------------------------|-------------|---------|------------------|------------------|
| E<br>t<br>h<br>e<br>r<br>n<br>e<br>t | R<br>O<br>U<br>T<br>E<br>R | +-----+     |         |                  |                  |
|                                      |                            | 44.736 MBPS | ds3 M13 | Line#A           | ds3 C-bit Parity |
|                                      |                            | -----       | -       | -                | ----->           |
|                                      |                            | 44.736 MBPS | ds3 M13 | Line#B           | ds3 C-bit Parity |
|                                      |                            | -----       | -       | -                | ----->           |
| -----                                | E<br>R                     | 44.736 MBPS | ds3 M13 | Line#C           | ds3 C-bit Parity |
|                                      |                            | -----       | -       | -                | ----->           |
| 44.736 MBPS                          |                            | ds3 M13     | Line#D  | ds3 C-bit Parity |                  |
| -----                                |                            | -           | -       | ----->           |                  |
| +-----+                              |                            |             |         |                  |                  |

The assignment of the index values could for example be:

| ifIndex (= dsx3IfIndex) |        |              | dsx3LineIndex |
|-------------------------|--------|--------------|---------------|
| 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            |

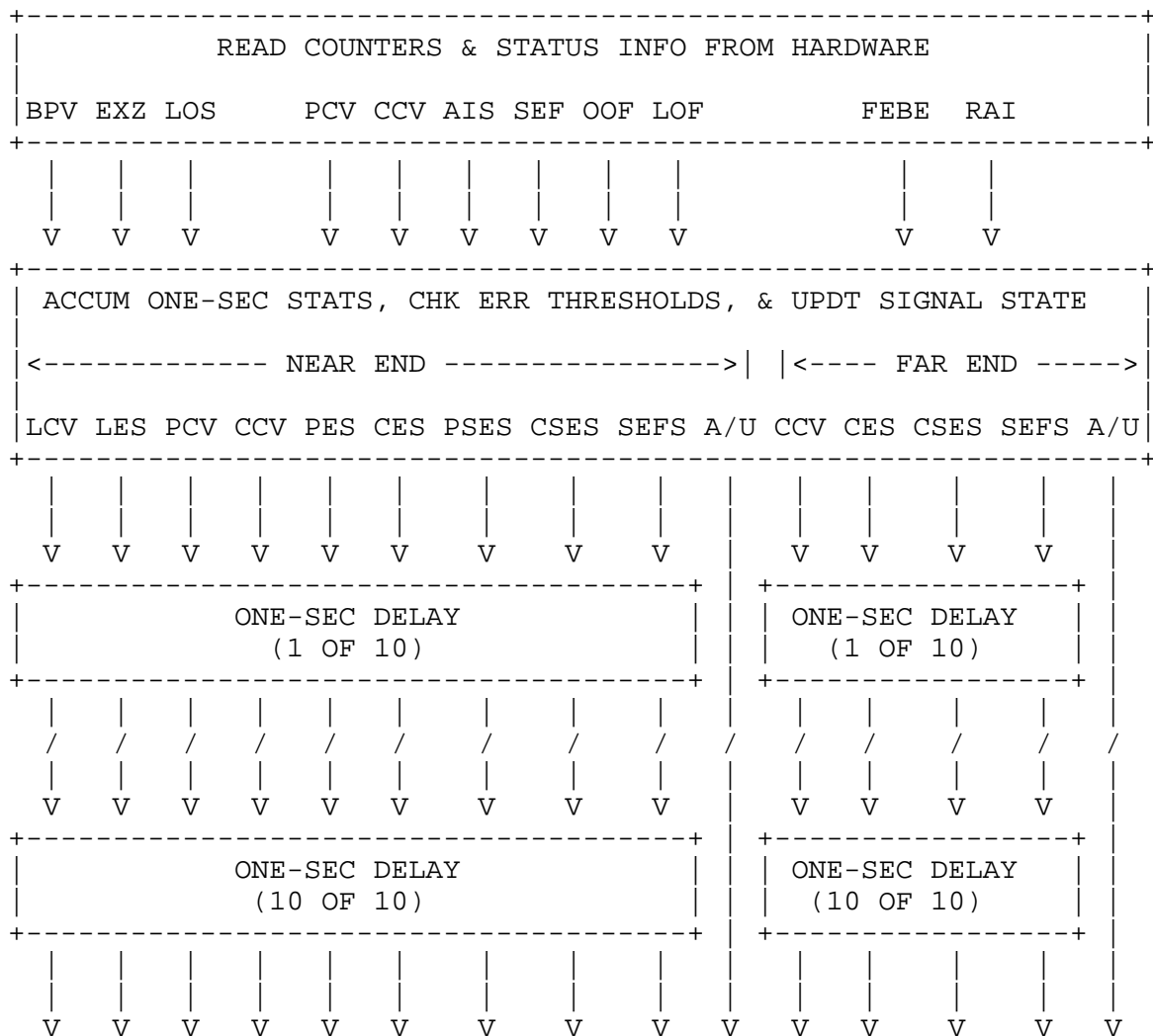
For this example, ifNumber is equal to 5. Note the following description of dsx3LineIndex: the dsx3LineIndex identifies a DS3/E3 Interface on a managed device. If there is an ifEntry that is directly associated with this and only this DS3/E3 interface, it should have the same value as ifIndex. Otherwise, number the dsx3LineIndices 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 (= dsx3IfIndex) |        |              | dsx3LineIndex |
|-------------------------|--------|--------------|---------------|
| 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 DS3 C-Bit parity application. Similar rules would apply for other interfaces covered by this MIB. 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 CCV PES CES PSES CSES SEFS UAS CCV CES CSES SEFS UAS|
+-----+

```

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] Fowler D., "Definitions of Managed Objects for the DS1 and E1 Interface Types", RFC 2495, January 1999.
- [18] Brown, T., and Tesink, K., "Definitions of Managed Objects for the SONET/SDH Interface Type", Work in Progress.
- [19] American National Standard for telecommunications - digital hierarchy - electrical interfaces, ANSI T1.102- 1987.
- [20] American National Standard for telecommunications - digital hierarchy - formats specification, ANSI T1.107- 1988.

[20a]ANSI T1.107a-1990.

[21] American National Standard for telecommunications - Carrier-to-Customer Installation - DS3 Metallic Interface, ANSI T1.404-1989.

[22] American National Standard for Telecommunications -- Layer 1 In-Service Digital Transmission Performance Monitoring T1.231, Sept 1993.

[23] CCITT - Digital Multiplex Equipment Operating at the Third Order Bit Rate of 34 368 Kbit/s and the Forth Order Bit Rate of 139 264 Kbit/s and Using Positive Justification, G.751

[24] European Telecommunications Standards Institute -- ETS "34M" -- Metropolitan Area Network Physical Convergence Layer Procedure for 34.368 Megabits per Second, T/NA(91)18, May 1991.

[25] Fowler, D., "Definitions of Managed Objects for the Ds0 and DS0Bundle Interface Types", RFC 2494, January 1999.

[26] Tesink, K., "Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals", RFC 2493, 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 dsx3LineType, for example, both ends of a ds3/e3 must have the same value in order for traffic to flow. In the case of dsx3SendCode and dsx3LoopbackConfig, for another example, traffic may stop transmitting when particular loopbacks are applied.

dsx3LineType  
dsx3LineCoding  
dsx3SendCode  
dsx3LoopbackConfig  
dsx3TransmitClockSource  
dsx3LineLength  
dsx3Channelization

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

dsx3CircuitIdentifier

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

dsx3LineStatusChangeTrabEnable

#### 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.

