                    Definitions of Managed Objects
                for the DS0 and DS0 Bundle 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

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 DS0 and DS0
   Bundle interfaces. This document is a companion document with
   Definitions of Managed Objects for the DS1/E1/DS2/E2 (RFC 2495 [17]),
   DS3/E3 (RFC 2496 [18]), 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

    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.

2. Overview

These objects are used when the particular media being used to
realize an interface is a DS0 interface. At present, this applies to
these values of the ifType variable in the Internet-standard MIB:

        ds0 (81)
        ds0Bundle (82)

2.1. BONDing Terminology

Please reference The BONDing Spec [20] for definitions of terms used
to describe bonding modes.

2.2. Use of ifTable for DS0 Layer

The following items are defined in RFC 2233 [16].

Only the ifGeneralInformationGroup and ifCounterDiscontinuityGroup
need to be supported.

ifTable Object	Use for DS0 Layer
ifIndex	Interface index.
ifDescr	See interfaces MIB [16].
ifType	ds0(81) or ds0Bundle(82).
ifSpeed	64000 for ds0 (regardless of the setting of robbed bit signalling) or N*64000 for ds0Bundle.
ifPhysAddress	The value of the Circuit Identifier. If no Circuit Identifier has been assigned this object should have an octet string with zero length.

            ifAdminStatus   See interfaces MIB [16].

            ifOperStatus   See interfaces MIB [16].

            ifLastChange   See interfaces MIB [16].

            ifName   See interfaces MIB [16].

            ifLinkUpDownTrapEnable   Set to disabled(2).
                                       Supports read-only access.

            ifHighSpeed   Set to rounded ifSpeed/1000000.

            ifConnectorPresent   Set to false(2).

2.3. Using ifStackTable

   This section describes by example how to use ifStackTable to
   represent the relationship of ds0 and ds0Bundles with ds1 interfaces.
   Implementors of the stack table for ds0 and ds0Bundle interfaces
   should look at the appropriate RFC for the service being stacked on
   ds0s and ds0Bundles. Examples given below are for illustration
   purposes only.

   Example: A Frame Relay Service is being carried on 4 ds0s of a ds1.
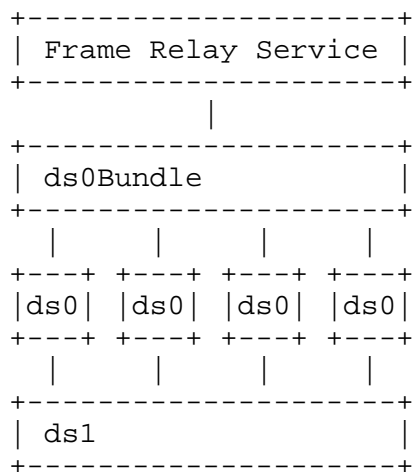
```
          +---------------------+
          | Frame Relay Service |
          +---------------------+
                    |
          +---------------------+
          | ds0Bundle           |
          +---------------------+
            |     |     |     |
          +---+ +---+ +---+ +---+
          |ds0| |ds0| |ds0| |ds0|
          +---+ +---+ +---+ +---+
            |     |     |     |
          +---------------------+
          | ds1                 |
          +---------------------+
```

   The assignment of the index values could for example be:

           ifIndex   Description
           1   FrameRelayService (type 44)
           2   ds0Bundle   (type 82)
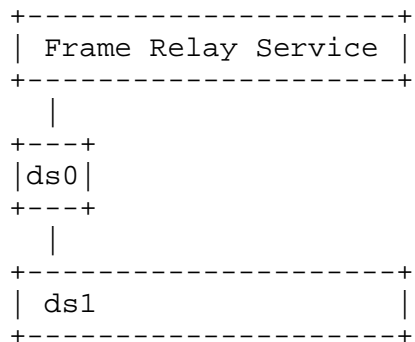   3   ds0 #1   (type 81)

```
        4          ds0 #2              (type 81)
        5          ds0 #3              (type 81)
        6          ds0 #4              (type 81)
        7          ds1                 (type 18)
```

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

```
        ifStackTable Entries

        HigherLayer    LowerLayer
        0              1
        1              2
        2              3
        2              4
        2              5
        2              6
        3              7
        4              7
        5              7
        6              7
        7              0
```

In the case where the frameRelayService is using a single ds0, then the ds0Bundle is not required.

```
        +--------------------+
        | Frame Relay Service |
        +--------------------+
           |
        +---+
        |ds0|
        +---+
           |
        +--------------------+
        | ds1                |
        +--------------------+
```

The assignment of the index values could for example be:

```
        ifIndex   Description
        1         FrameRelayService (type 44)
        2         ds0               (type 81)
        3         ds1               (type 18)
```

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

```
ifStackTable Entries

HigherLayer    LowerLayer
0              1
1              2
2              3
3              0
```

2.3.1. 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. Timeslot
   16 is not created for framing types that do not pass data over it.

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

```
dsx1ChanMappingTable Entries

ifIndex   dsx1Ds1ChannelNumber    dsx1ChanMappedIfIndex
1         1                       2
1         2                       3
......
1         28                      29
```

   In addition, the DS1s are channelized into DS0s. The object
   dsx1Channelization is set to enabledDs0 for each DS1. When this
   object is set to this value, 24 DS0s are created by the agent. There
   will be 24 DS0s in the ifTable for each DS1. If the
   dsx1Channelization is set to disabled, the 24 DS0s are destroyed.

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

```
        dsx0ChanMappingTable Entries

        ifIndex    dsx0Ds0ChannelNumber  dsx0ChanMappedIfIndex
        2          1                     30
        2          2                     31
        ......
        2          24                    53
```

2.3.2. Usage of ifIndex Mapping for DS0Bundle

   An example is given here to explain the ifIndex mapping objects in
   the DS0Bundle MIB to help the implementor use the objects correctly.

   Assume that a DS1 (with ifIndex 1) is channelized into DS0s. There
   will be 24 DS0s in the ifTable. Assume the entries in the ifTable
   for the DS0s are created in channel order and the ifIndex values are
   2 through 25. Now, assume that there are two bundles on the DS1.
   The first one uses channels 1 and 2. The second uses channels 3 and
   4. There will be two ifTable entries for these bundles, with values
   of 26 and 27 for ifIndex. There will be an entry in the
   dsx0BundleTable for each bundle. The entries will be as follows:

```
        dsx0BundleTable Entries

        dsx0BundleIndex    dsx0BundleIfIndex
        1                  26
        2                  27
```

   There will be an entry in the dsx0ConfigTable for each DS0. The
   entries will be as follows:

```
        dsx0ConfigTable Entries

        ifIndex    dsx0Ds0ChannelNumber   dsx0Ds0BundleMappedIfIndex
        2          1                      26
        3          2                      26
        4          3                      27
        5          4                      27
        6          5                      0
        7          6                      0
        ......
        25         24                     0
```

3. Overview of the MIB

   This document contains 2 MIB modules, the DS0 MIB and the DS0Bundle
   MIB.

3.1. DS0 MIB

   The DS0 MIB is used to represent individual DS0s in a DS1 or E1.
   Variables in this MIB would be created for each DS0 in the ifTable.
   This MIB contains the following group:

   The DS0 Config Group - This group contains configuration information
   about a particular DS0.

3.2. DS0Bundle MIB

   The DS0Bundle MIB is used to represent collections of DS0s that are
   used together to carry data within a DS1/E1 at speeds greater than
   that of a single DS0. DS0Bundles are created on top of DS0s and are
   represented that way in the ifStackTable. This MIB contains the
   following groups:

   The DS0 Bundle Group - This group contains objects used for creating
   new ds0Bundles. This group is mandatory.

   The DS0 Bonding Group - This group contains information about bonding
   for a ds0Bundle, if bonding is enabled. This group is optional.

4. Object Definitions for DS0

   DS0-MIB DEFINITIONS ::= BEGIN

   IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        transmission   FROM SNMPv2-SMI
        MODULE-COMPLIANCE, OBJECT-GROUP   FROM SNMPv2-CONF
        DisplayString, TruthValue   FROM SNMPv2-TC
        ifIndex, InterfaceIndex,
        InterfaceIndexOrZero   FROM IF-MIB;

   -- This is the MIB module for the DS0 Interface objects.

   ds0 MODULE-IDENTITY
        LAST-UPDATED "9807161630Z"
        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 3619

           E-mail: davef@newbridge.com"
        DESCRIPTION
             "The MIB module to describe
              DS0 interfaces objects."
        REVISION "9805242010Z"
        DESCRIPTION
             "Initial version of the DS0-MIB."

        ::= { transmission 81 }

   -- The DS0 Config Group

   -- Implementation of this group is mandatory for all
   -- systems that use a DS0 Interface.

   -- The DS0 Config Group consists of two tables:
   --   DS0 Configuration Table
   --   DS0 Channel Mapping Table

   -- The DS0 Configuration Table

   dsx0ConfigTable OBJECT-TYPE
       SYNTAX   SEQUENCE OF Dsx0ConfigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
               "The DS0 Configuration table."
       ::= { ds0 1 }

   dsx0ConfigEntry OBJECT-TYPE
       SYNTAX Dsx0ConfigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
               "An entry in the DS0 Configuration table. There
               is an entry in this table for each DS0 interface."
       INDEX   { ifIndex }
       ::= { dsx0ConfigTable 1 }

   Dsx0ConfigEntry ::=
       SEQUENCE {
           dsx0Ds0ChannelNumber   INTEGER,
           dsx0RobbedBitSignalling   TruthValue,
           dsx0CircuitIdentifier   DisplayString,
           dsx0IdleCode   INTEGER,
           dsx0SeizedCode   INTEGER,

```
            dsx0ReceivedCode              INTEGER,
            dsx0TransmitCodesEnable       TruthValue,
            dsx0Ds0BundleMappedIfIndex    InterfaceIndexOrZero
    }

    dsx0Ds0ChannelNumber OBJECT-TYPE
        SYNTAX   INTEGER(0..31)
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
                "This object indicates the channel number of the
                ds0 on its DS1/E1."
        ::= { dsx0ConfigEntry 1 }

    dsx0RobbedBitSignalling OBJECT-TYPE
        SYNTAX   TruthValue
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
                "This object indicates if Robbed Bit Signalling is
                turned on or off for a given ds0.  This only
                applies to DS0s on a DS1 link.  For E1 links the
                value is always off (false)."
        ::= { dsx0ConfigEntry 2 }

    dsx0CircuitIdentifier OBJECT-TYPE
        SYNTAX   DisplayString (SIZE (0..255))
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
                "This object contains the transmission vendor's
                circuit identifier, for the purpose of
                facilitating troubleshooting."
        ::= { dsx0ConfigEntry 3 }

    dsx0IdleCode OBJECT-TYPE
        SYNTAX   INTEGER(0..15)
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
                "This object contains the code transmitted in the
                ABCD bits when the ds0 is not connected and
                dsx0TransmitCodesEnable is enabled.  The object is
                a bitmap and the various bit positions are:
                        1       D bit
                        2       C bit
                        4       B bit
                        8       A bit"
```

```
          ::= { dsx0ConfigEntry 4 }

    dsx0SeizedCode OBJECT-TYPE
        SYNTAX  INTEGER(0..15)
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
               "This object contains the code transmitted in the
               ABCD bits when the ds0 is connected and
               dsx0TransmitCodesEnable is enabled.  The object is
               a bitmap and the various bit positions are:
                       1      D bit
                       2      C bit
                       4      B bit
                       8      A bit"
        ::= { dsx0ConfigEntry 5 }

    dsx0ReceivedCode OBJECT-TYPE
        SYNTAX  INTEGER(0..15)
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
               "This object contains the code being received in
               the ABCD bits.  The object is a bitmap and the
               various bit positions are:
                       1      D bit
                       2      C bit
                       4      B bit
                       8      A bit"
        ::= { dsx0ConfigEntry 6 }

    dsx0TransmitCodesEnable OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS  read-write
        STATUS  current
        DESCRIPTION
               "This object determines if the idle and seized
               codes are transmitted. If the value of this object
               is true then the codes are transmitted."
        ::= { dsx0ConfigEntry 7 }

    dsx0Ds0BundleMappedIfIndex OBJECT-TYPE
        SYNTAX InterfaceIndexOrZero
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
               "This object indicates the ifIndex value assigned
               by the agent for the ds0Bundle(82) ifEntry to
```

              which the given ds0(81) ifEntry may belong.

              If the given ds0(81) ifEntry does not belong to
              any ds0Bundle(82) ifEntry, then this object has a
              value of zero.

              While this object provides information that can
              also be found in the ifStackTable, it provides
              this same information with a single table lookup,
              rather than by walking the ifStackTable to find
              the possibly non-existent ds0Bundle(82) ifEntry
              that may be stacked above the given ds0(81)
              ifTable entry."
         ::= { dsx0ConfigEntry 8 }

    -- The DS0 Channel Mapping Table

    dsx0ChanMappingTable OBJECT-TYPE
         SYNTAX   SEQUENCE OF Dsx0ChanMappingEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
              "The DS0 Channel Mapping table. This table maps a
              DS0 channel number on a particular DS1/E1 into an
              ifIndex."
         ::= { ds0 3 }

    dsx0ChanMappingEntry OBJECT-TYPE
         SYNTAX Dsx0ChanMappingEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
              "An entry in the DS0 Channel Mapping table. There
              is an entry in this table corresponding to each
              ds0 ifEntry within any interface that is
              channelized to the individual ds0 ifEntry level.

              This table is intended to facilitate mapping from
              channelized interface / channel number to DS0
              ifEntry. (e.g. mapping (DS1 ifIndex, DS0 Channel
              Number) -> ifIndex)

              While this table provides information that can
              also be found in the ifStackTable and
              dsx0ConfigTable, it provides this same information
              with a single table lookup, rather than by walking
              the ifStackTable to find the various constituent
              ds0 ifTable entries, and testing various

```
            dsx0ConfigTable entries to check for the entry
            with the applicable DS0 channel number."
        INDEX   { ifIndex, dsx0Ds0ChannelNumber }
        ::= { dsx0ChanMappingTable 1 }

    Dsx0ChanMappingEntry ::=
        SEQUENCE {
            dsx0ChanMappedIfIndex  InterfaceIndex
    }

    dsx0ChanMappedIfIndex OBJECT-TYPE
        SYNTAX  InterfaceIndex
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
            "This object indicates the ifIndex value assigned
            by the agent for the individual ds0 ifEntry that
            corresponds to the given DS0 channel number
            (specified by the INDEX element
            dsx0Ds0ChannelNumber) of the given channelized
            interface (specified by INDEX element ifIndex)."
        ::= { dsx0ChanMappingEntry 1 }

    -- conformance information

    ds0Conformance OBJECT IDENTIFIER ::= { ds0 2 }

    ds0Groups       OBJECT IDENTIFIER ::= { ds0Conformance 1 }
    ds0Compliances OBJECT IDENTIFIER ::= { ds0Conformance 2 }



    -- compliance statements

    ds0Compliance MODULE-COMPLIANCE
        STATUS  current
        DESCRIPTION
            "The compliance statement for DS0 interfaces."
        MODULE  -- this module
            MANDATORY-GROUPS { ds0ConfigGroup }

            OBJECT dsx0RobbedBitSignalling
            MIN-ACCESS read-only
            DESCRIPTION
                "The ability to set RBS is not required."
            OBJECT dsx0CircuitIdentifier
            MIN-ACCESS read-only
            DESCRIPTION
```

```
                "The ability to set the circuit identifier is not
                required."

            OBJECT dsx0IdleCode
            MIN-ACCESS read-only
            DESCRIPTION
                "The ability to set the idle code is not
                required."

            OBJECT dsx0SeizedCode
            MIN-ACCESS read-only
            DESCRIPTION
                "The ability to set the seized code is not
                required."

            OBJECT dsx0TransmitCodesEnable
            MIN-ACCESS read-only
            DESCRIPTION
                "The ability to enable and disable the
                transmitting of idle and seized codes is not
                required."

        ::= { ds0Compliances 1 }

    -- units of conformance

    ds0ConfigGroup  OBJECT-GROUP
        OBJECTS { dsx0Ds0ChannelNumber,
                  dsx0RobbedBitSignalling,
                  dsx0CircuitIdentifier,
                  dsx0IdleCode,
                  dsx0SeizedCode,
                  dsx0ReceivedCode,
                  dsx0TransmitCodesEnable,
                  dsx0Ds0BundleMappedIfIndex,
                  dsx0ChanMappedIfIndex }
        STATUS  current
        DESCRIPTION
                "A collection of objects providing configuration
                information applicable to all DS0 interfaces."
        ::= { ds0Groups 1 }

    END
```

5. Object Definitions for DS0 Bundle

     DS0BUNDLE-MIB DEFINITIONS ::= BEGIN

     IMPORTS
          MODULE-IDENTITY, OBJECT-TYPE,
          transmission   FROM SNMPv2-SMI
          MODULE-COMPLIANCE, OBJECT-GROUP   FROM SNMPv2-CONF
          DisplayString, RowStatus,
          TestAndIncr   FROM SNMPv2-TC
          ifIndex, InterfaceIndex   FROM IF-MIB;

     --   This is the MIB module for the DS0Bundle Interface
     --   objects.

     ds0Bundle MODULE-IDENTITY
          LAST-UPDATED "9807161630Z"
          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 3619

             E-mail: davef@newbridge.com"
          DESCRIPTION
              "The MIB module to describe
               DS0 Bundle interfaces objects."
          REVISION "9805242010Z"
          DESCRIPTION
              "Initial version of the DS0BUNDLE-MIB."

           ::= { transmission 82 }

     --
     -- The DS0 Bundle Config Group
     --
     -- Implementation of this group is mandatory for all
     -- systems that use a DS0Bundle Interface.
     --
     -- The DS0 Bundle Config Group consists of one table:
     --   DS0 Bundle Table

     -- The DS0 Bundle Table

```
     dsx0BundleNextIndex OBJECT-TYPE
         SYNTAX   TestAndIncr
         MAX-ACCESS   read-write
         STATUS   current
         DESCRIPTION
                 "This object is used to assist the manager in
                 selecting a value for dsx0BundleIndex.  Because
                 this object is of syntax TestAndIncr (see the
                 SNMPv2-TC document, RFC 1903) it can also be used
                 to avoid race conditions with multiple managers
                 trying to create rows in the table.

                 If the result of the SET for dsx0BundleNextIndex
                 is not success, this means the value has been
                 changed from index (i.e. another manager used the
                 value), so a new value is required.

                 The algorithm is:
                 done = false
                 while done == false
                     index = GET (dsx0BundleNextIndex.0)
                     SET (dsx0BundleNextIndex.0=index)
                     if (set failed)
                       done = false
                     else
                       SET(dsx0BundleRowStatus.index=createAndGo)
                       if (set failed)
                         done = false
                       else
                         done = true
                         other error handling"
         ::= { ds0Bundle 2 }

     dsx0BundleTable OBJECT-TYPE
         SYNTAX   SEQUENCE OF Dsx0BundleEntry
         MAX-ACCESS   not-accessible
         STATUS   current
         DESCRIPTION
                 "There is an row in this table for each ds0Bundle
                 in the system.  This table can be used to
                 (indirectly) create rows in the ifTable with
                 ifType = 'ds0Bundle(82)'."
         ::= { ds0Bundle 3 }

     dsx0BundleEntry OBJECT-TYPE
         SYNTAX   Dsx0BundleEntry
         MAX-ACCESS   not-accessible
         STATUS   current
```

```
        DESCRIPTION
                "There is a row in entry in this table for each
                ds0Bundle interface."
        INDEX   { dsx0BundleIndex }
        ::= { dsx0BundleTable 1 }

    Dsx0BundleEntry ::=
        SEQUENCE {
            dsx0BundleIndex              INTEGER,
            dsx0BundleIfIndex            InterfaceIndex,
            dsx0BundleCircuitIdentifier  DisplayString,
            dsx0BundleRowStatus          RowStatus
    }

    dsx0BundleIndex OBJECT-TYPE
        SYNTAX   INTEGER (0..2147483647)
        MAX-ACCESS  not-accessible
        STATUS   current
        DESCRIPTION
                "A unique identifier for a ds0Bundle.  This is not
                the same value as ifIndex.  This table is not
                indexed by ifIndex because the manager has to
                choose the index in a createable row and the agent
                must be allowed to select ifIndex values."
        ::= { dsx0BundleEntry 1 }

    dsx0BundleIfIndex OBJECT-TYPE
        SYNTAX   InterfaceIndex
        MAX-ACCESS  read-only
        STATUS   current
        DESCRIPTION
                "The ifIndex value the agent selected for the
                (new) ds0Bundle interface."
        ::= { dsx0BundleEntry 2 }

    dsx0BundleCircuitIdentifier OBJECT-TYPE
         SYNTAX  DisplayString (SIZE (0..255))
         MAX-ACCESS  read-create
         STATUS   current
         DESCRIPTION
                "This variable contains the transmission vendor's
                circuit identifier, for the purpose of
                facilitating troubleshooting."
         ::= { dsx0BundleEntry 3 }

    dsx0BundleRowStatus OBJECT-TYPE
        SYNTAX   RowStatus
        MAX-ACCESS  read-create
```

```
        STATUS  current
        DESCRIPTION
                "This object is used to create and delete rows in
                this table."
        ::= { dsx0BundleEntry 4 }

    -- The DS0 Bonding Group

    -- Implementation of this group is optional for all
    -- systems that use a DS0Bundle Interface.

    -- The DS0 Bonding Group consists of one table:
    --    DS0 Bonding Table

    -- The DS0 Bonding Table

    dsx0BondingTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Dsx0BondingEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
                "The DS0 Bonding table."
        ::= { ds0Bundle 1 }

    dsx0BondingEntry OBJECT-TYPE
        SYNTAX  Dsx0BondingEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
                "An entry in the DS0 Bonding table.  There is a
                row in this table for each DS0Bundle interface."
        INDEX   { ifIndex }
        ::= { dsx0BondingTable 1 }

    Dsx0BondingEntry ::=
        SEQUENCE {
            dsx0BondMode            INTEGER,
            dsx0BondStatus          INTEGER,
            dsx0BondRowStatus       RowStatus
    }

    dsx0BondMode OBJECT-TYPE
        SYNTAX  INTEGER {
                    none(1),
                    other(2),
                    mode0(3),
                    mode1(4),
                    mode2(5),
```

```
                        mode3(6)
        }
        MAX-ACCESS  read-create
        STATUS  current
        DESCRIPTION
                "This object indicates which BONDing mode is used,
                if any, for a ds0Bundle.  Mode0 provides parameter
                and number exchange with no synchronization.  Mode
                1 provides parameter and number exchange.  Mode 1
                also provides synchronization during
                initialization but does not include inband
                monitoring.  Mode 2 provides all of the above plus
                inband monitoring.  Mode 2 also steals 1/64th of
                the bandwidth of each channel (thus not supporting
                n x 56/64 kbit/s data channels for most values of
                n). Mode 3 provides all of the above, but also
                provides n x 56/64 kbit/s data channels.  Most
                common implementations of Mode 3 add an extra
                channel to support the inband monitoring overhead.
                ModeNone should be used when the interface is not
                performing bandwidth-on-demand."
        ::= { dsx0BondingEntry 1 }

    dsx0BondStatus OBJECT-TYPE
        SYNTAX  INTEGER {
                    idle(1),
                    callSetup(2),
                    dataTransfer(3)
        }
        MAX-ACCESS  read-only
        STATUS  current
        DESCRIPTION
                "This object indicates the current status of the
                bonding call using this ds0Bundle. idle(1) should
                be used when the bonding mode is set to none(1)."
        ::= { dsx0BondingEntry 2 }

    dsx0BondRowStatus OBJECT-TYPE
        SYNTAX  RowStatus
        MAX-ACCESS  read-create
        STATUS  current
        DESCRIPTION
                "This object is used to create new rows in this
                table, modify existing rows, and to delete
                existing rows."
        ::= { dsx0BondingEntry 3 }
```

    -- conformance information

    ds0BundleConformance OBJECT IDENTIFIER ::= { ds0Bundle 4 }

    ds0BundleGroups   OBJECT IDENTIFIER
        ::= { ds0BundleConformance 1 }

    ds0BundleCompliances OBJECT IDENTIFIER
        ::= { ds0BundleConformance 2 }

    -- compliance statements

    ds0BundleCompliance MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
                "The compliance statement for DS0Bundle
                interfaces."
        MODULE -- this module
            MANDATORY-GROUPS {ds0BundleConfigGroup }

            GROUP   ds0BondingGroup
            DESCRIPTION
                "Implementation of this group is optional for all
                systems that attach to a DS0Bundle Interface."

            OBJECT   dsx0BundleRowStatus
            SYNTAX   INTEGER {
                        active(1),
                        createAndGo(4),
                        destroy(6)
                        }
            MIN-ACCESS read-only
            DESCRIPTION
                "The agent is not required to support a SET
                operation to this object, and only three of the
                six enumerated values for the RowStatus textual
                convention need be supported. Only supporting
                createAndGo for a creation process prevents the
                manager from creating an inactive row in the
                ds0BundleTable. Inactive rows in the
                ds0BundleTable do not make sense."

            OBJECT   dsx0BundleCircuitIdentifier
            MIN-ACCESS read-only
            DESCRIPTION
                "The agent is not required to support a SET

```
             operation to this object."

                 ::= { ds0BundleCompliances 1 }

             -- units of conformance

             ds0BondingGroup  OBJECT-GROUP
                 OBJECTS { dsx0BondMode,
                           dsx0BondStatus,
                           dsx0BondRowStatus }
                 STATUS   current
                 DESCRIPTION
                         "A collection of objects providing
                          configuration information applicable
                          to all DS0 interfaces."
        ::= { ds0BundleGroups 1 }
   ds0BundleConfigGroup  OBJECT-GROUP
       OBJECTS { dsx0BundleNextIndex,
                 dsx0BundleIfIndex,
                 dsx0BundleCircuitIdentifier,
                 dsx0BundleRowStatus }
       STATUS   current
       DESCRIPTION
               "A collection of objects providing the ability to
                create a new ds0Bundle in the ifTable as well as
                configuration information about the ds0Bundle."
       ::= { ds0BundleGroups 2 }
   END
```

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, E1, DS2,
         and E2 Interface Types", RFC 2495, January 1999.

    [18] Fowler, D., "Definitions of Managed Objects for the DS3/E3
         Interface Types", RFC 2496, January 1999.

    [19] Brown, T., and K. Tesink, "Definitions of Managed Objects for
         the SONET/SDH Interface Type", Work in Progress.

    [20] Sharp, H. (Editor), "Interoperability Requirements for Nx56/64
         kbit/s Calls", BONDING Spec Version 1.0, BONDING Consortium,
         Sept 1992.

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 the following objects to an inappropriate value can cause
loss of traffic. In the case of dsx0RobbedBitSignalling, for
example, the nature of the traffic flowing on the DS0 can be
affected.
     dsx0RobbedBitSignalling
     dsx0IdleCode
     dsx0SeizedCode
     dsx0TransmitCodesEnable
     dsx0BundleRowStatus
     dsx0BondMode
     dsx0BondRowStatus

Setting the following objects is mischievous, but not harmful to
traffic.
     dsx0CircuitIdentifier
     dsx0BundleNextIndex

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