

Network Working Group
Request for Comments: 1604
Obsoletes: 1596
Category: Standards Track

T. Brown, Editor
Bell Communications Research
March 1994

Definitions of Managed Objects for Frame Relay Service

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.

Abstract

This memo defines an extension to the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing the Frame Relay Service.

Table of Contents

1. The SNMPv2 Network Management Framework	2
2. Object Definitions	2
3. Overview	2
3.1 Scope of MIB	3
3.2 Frame Relay Service MIB Terminology	5
3.3 Apply MIB II to a Frame Relay Service	7
4. Object Definitions	12
4.1 The Frame Relay Service Logical Port Group	12
4.2 The Frame Relay Management VC Signaling Group	15
4.3 The PVC End-Point Group	22
4.4 Frame Relay PVC Connection Group	30
4.5 Frame Relay Accounting Groups	37
5. Frame Relay Network Service TRAPS	40
6. Conformance Information	43
7. Acknowledgments	45
8. References	45
9. Security Considerations	46
10. Author's Address	46

1. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- o RFC 1442 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- o STD 17, RFC 1213 defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- o RFC 1445 which defines the administrative and other architectural aspects of the framework.
- o RFC 1448 which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

3. Overview

These objects are used when the particular media being used to manage is Frame Relay Service. At present, this applies to these values of the ifType variable in the Internet-standard MIB:

```
frameRelayService (44)
```

This section provides an overview and background of how to use this MIB and other potential MIBs when managing a Frame Relay Service.

Figure 1 shows the MIB stack that could be followed for managing a Frame Relay Service. This is only an example and not meant to be inclusive.

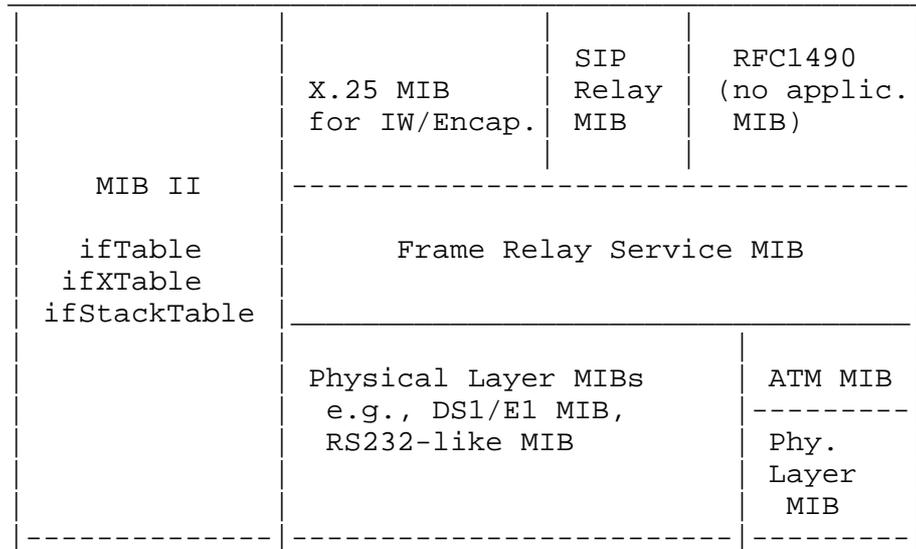


Figure 1. Frame Relay MIB Architecture

3.1. Scope of MIB

The Frame Relay Service MIB will only manage the Frame Relay portion of the network. This MIB is based upon the Customer Network Management concepts presented in the document "Service Management Architecture for Virtual Connection Services" [6].

This MIB will NOT be implemented on User Equipment (e.g., DTE), and the Frame Relay DTE MIB (RFC 1315) should be used to manage those devices [8].

Frame Relay Service MIB is intended to be used for Customer Network Management (CNM) of a Frame Relay Network Service. It provides information that allows end-customers to obtain performance monitoring, fault detection, and configuration information about their Frame Relay Service. It is an implementation decision as to whether this MIB is used to create/delete/modify PVCs and to turn PVCs on or off.

By using this and other related MIBs, a customer's NMS can monitor their PVCs and UNI/NNI logical ports. Internal aspects of the network (e.g., switching elements, line cards, and network routing tables) are outside the scope of this MIB. The Customer's NMS will typically access the SNMP proxy-agent within the Frame Relay network using SNMP over UDP over IP with IP encapsulated in Frame Relay according to RFC1490/ANSI T1.617 Annex F [7,9]. The customer, thus,

has a PVC to the SNMP proxy-agent. Alternate access mechanisms and SNMP agent implementations are possible. The service capabilities include retrieving information and receiving TRAPS. It is beyond the scope of this MIB to define managed objects to monitor the physical layer. Existing physical layer MIBs (e.g., DS1 MIB) and MIB II will be used as possible. The Frame Relay Service SNMP MIB for CNM will not contain any managed objects to monitor the physical layer. This MIB primarily addresses Frame Relay PVCs. This MIB may be extended at a later time to handle Frame Relay SVCs.

This MIB is only used to manage a single Frame Relay Service offering from one network. This MIB will typically be implemented on a service provider's SNMP proxy-agent. The SNMP proxy-agent proxies for all Frame Relay equipment within one service provider's Frame Relay network. (Other SNMP agent implementations are not precluded.) Therefore, this MIB models a PVC segment through one Frame Relay Network. See Figure 2. If the customer's PVCs traverse multiple networks, then the customer needs to poll multiple network proxy-agents within each Frame Relay Network to retrieve their end-to-end view of their service. See Figure 2 and the Service Management Architecture [6].

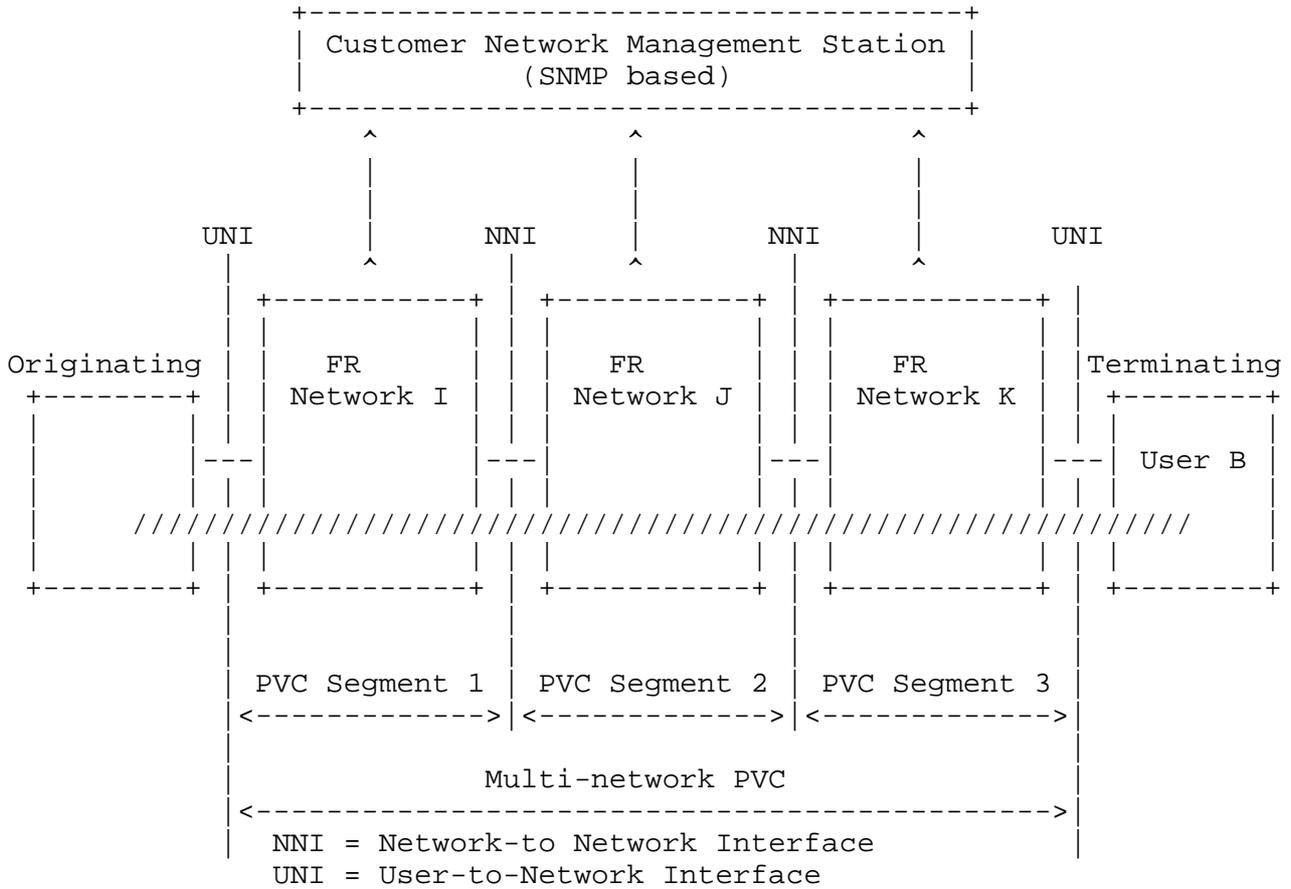


Figure 2. Multi-network PVC

Also, since the Frame Relay network is a shared network amongst many Frame Relay subscribers, each subscriber will only have access to their information (e.g., information with respect to their interfaces and PVCs). Therefore, in order to provide this capability, the Frame Relay PVC CNM proxy agent should be able to support instance level granularity for MIB views. See the Service Management Architecture.

3.2. Frame Relay Service MIB Terminology

Access Channel - An access channel generically refers to the DS1/E1 or DS3/E3-based UNI access channel or NNI access channel across which frame relay data transits. An access channel is the access pathway for a single stream of user data.

Within a given T1 line, an access channel can denote any one of the following:

- o Unchannelized T1 - the entire T1 line is considered an access channel. Each access channel is comprised of 24 T1 time slots.
- o Channelized T1 - an access channel is any one of 24 channels. Each access channel is comprised of a single T1 time slot.
- o Fractional T1 - an access channel is a grouping of N T1 time slots (NX56/64 Kbps, where N = 1-23 T1 Time slots per FT1 Access Channel) that may be assigned in consecutive or non-consecutive order.

Within a given E1 line, a channel can denote any one of the following:

- o Unchannelized E1 - the entire E1 line is considered a single access channel. Each access channel is comprised of 31 E1 time slots.
- o Channelized E1 - an access channel is any one of 31 channels. Each access channel is comprised of a single E1 time slot.
- o Fractional E1 - an access channel is a grouping of N E1 time slots (NX64 Kbps, where N = 1-30 E1 time slots per FE1 access channel) that may be assigned in consecutive or non-consecutive order.

in 3 Within a given unformatted line, the entire unformatted line is considered an access channel. Examples include RS-232, V.35, V.36 and X.21 (non- switched).

Access Rate - The data rate of the access channel, expressed in bits/second. The speed of the user access channel determines how rapidly the end user can inject data into the network.

Bc - The Committed Burst Size (Bc) is the maximum amount of subscriber data (expressed in bits) that the network agrees to transfer, under normal conditions, during a time interval Tc.

Be - The Excess Burst Size (Be) is the maximum amount of subscriber data (expressed in bits) in excess of Bc that the network will attempt to deliver during the time interval Tc. This data (Be) is delivered in general with a lower probability than Bc.

CIR - The Committed Information Rate (CIR) is the subscriber data rate (expressed in bits/second) that the network commits to deliver under normal network conditions. CIR is averaged over the

time interval T_c ($CIR = B_c/T_c$).

DLCI - Data Link Connection Identifier

Logical Port - This term is used to model the Frame Relay "interface" on a device.

NNI - Network to Network Interface

Permanent Virtual Connection (PVC) - A virtual connection that has its end-points and bearer capabilities defined at subscription time.

Time slot (E1) - An octet within the 256-bit information field in each E1 frame is defined as a time slot. Time slots are position sensitive within the 256-bit information field. Fractional E1 service is provided in contiguous or non- contiguous time slot increments.

Time slot (T1) - An octet within the 192-bit information field in each T1 frame is defined as a time slot. Time slots are position sensitive within the 192-bit information field. Fractional T1 service is provided in contiguous or non- contiguous time slot increments.

UNI - User to Network Interface

N391 - Full status (status of all PVCs) polling counter

N392 - Error threshold

N393 - Monitored events count

T391 - Link integrity verification polling timer

T392 - Polling verification timer

nT3 - Status enquiry timer

nN3 - Maximum status enquiry counter

3.3. Apply MIB II to a Frame Relay Service

Use the System Group to apply to the SNMP proxy-agent, since the proxy-agent may be monitoring many Frame Relay devices in one network. System Group applies to only one system. This group is not instantiated.

sysDescr: ASCII string describing the SNMP proxy-agent. Can be up to 255 characters long. This field is generally used to indicate the network providers identification and type of service offered.

sysObjectID: Unique OBJECT IDENTIFIER (OID) for the SNMP proxy-agent.

sysUpTime: Clock in the SNMP proxy-agent; TimeTicks in 1/100s of a second. Elapsed type since the proxy-agent came on line.

sysContact: Contact for the SNMP proxy-agent. ASCII string of up to 255 characters.

sysName: Domain name of the SNMP proxy-agent, for example, acme.com

sysLocation: Location of the SNMP proxy-agent. ASCII string of up to 255 characters.

sysServices: Services of the managed device. The value "2", which implies that the Frame Relay network is providing a subnetwork level service, is recommended.

This specifies how the Interfaces Group defined in MIB II shall be used for the management of Frame Relay based interfaces, and in conjunction with the Frame Relay Service MIB module. This memo assumes the interpretation of the evolution of the Interfaces group to be in accordance with: "The interfaces table (ifTable) contains information on the managed resource's interfaces. Each sub-layer below the internetwork layer of a network interface is considered an interface." Thus, the ifTable allows the following Frame Relay-based interfaces to be represented as table entries:

- Frame Relay interfaces in the Frame Relay equipment (e.g., switches, routers or networks) with Frame Relay interfaces. This level is concerned with generic frame counts and not with individual virtual connections.

In accordance with the guidelines of ifTable, frame counts per virtual connection are not covered by ifTable, and are considered interface specific and covered in the Frame Relay Service MIB module. In order to interrelate the ifEntries properly, the Interfaces Stack Group shall be supported.

Some specific interpretations of ifTable for Frame Relay follow.

Object	Use for the generic Frame Relay layer
=====	=====
ifIndex	Each Frame Relay port is represented by an ifEntry.
ifDescr	Description of the Frame Relay interface. ASCII string describing the UNI/NNI logical port. Can be up to 255 characters long.
ifType	The value allocated for Frame Relay Service is equal to 44.
ifMtu	Set to maximum frame size in octets for this frame relay logical port.
ifSpeed	Peak bandwidth in bits per second available for use. This could be the speed of the logical port and not the access rate. Actual user information transfer rate (i.e., access rate) of the UNI or NNI logical port in bits per second (this is not the clocking speed). For example, it is 1,536,000 bits per second for a DS1-based UNI/NNI logical port and 1,984,000 bits per second for an E1-based UNI/NNI logical port.
ifPhysAddress	The primary address for this logical port assigned by the Frame Relay interface provider. An octet string of zero length if no address is used for this logical port.
ifAdminStatus	The desired administrative status of the frame relay logical port.
ifOperStatus	The current operational status of the Frame Relay UNI or NNI logical port.
ifLastChange	The elapsed time since the last re-initialization of the logical port. The value of sysUpTime at the time the logical port entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then

this object contains a zero value.

ifInOctets The number of received octets.
This counter only counts octets from the beginning of the frame relay header field to the end of user data.

ifInUcastPkts The number of received unerrored, unicast frames.

ifInDiscards The number of received frames discarded. Possible reasons are as follows: policing, congestion.

ifInErrors The number of received frames that are discarded, because of an error. Possible errors can be the following: the frame relay frames were too long or were too short, the frames had an invalid or unrecognized DLCI values, or incorrect header values.

ifInUnknownProtos The number of packets discarded because of an unknown or unsupported protocol. For Frame Relay Service interfaces, this counter will always be zero.

ifOutOctets The number of transmitted octets.
This counter only counts octets from the beginning of the frame relay header field to the end of user data.

ifOutUcastpkts The number of frames sent.

ifOutDiscards The number of frames discarded in the egress direction. Possible reasons are as follows: policing, congestion.

ifOutErrors The number of frames discarded in the egress direction, because of errors. Possible reason is transmit underruns.

ifName This variable is not applicable for Frame Relay Service interfaces, therefore, this variable contains a zero-length string.

`ifInMulticastPkts` The number of received unerrored, multicast frames.

`ifInBroadcastPkts` This variable is not applicable for Frame Relay Service interfaces, therefore, this counter is always zero.

`ifOutMulticastPkts` The number of sent unerrored, multicast frames.

`ifOutBroadcastPkts` This variable is not applicable for Frame Relay Service interfaces, therefore, this counter is always zero.

`ifHCInOctets` Only used for DS3-based (and greater) Frame Relay logical ports. The number of received octets. This counter only counts octets from the beginning of the frame relay header field to the end of user data.

`ifHCOctets` Only used for DS3-based (and greater) Frame Relay logical ports. The number of transmitted octets. This counter only counts octets from the beginning of the frame relay header field to the end of user data.

`ifLinkUpDownTrapEnable` The value of this object is implementation specific for Frame Relay logical ports.

`ifHighSpeed` Set to the user data rate of the frame relay logical port in millions of bits per second. If the user data rate is less than 1 Mbps, then this value is zero.

`ifPromiscuousMode` Set to false(2).

`ifConnectorPresent` Set to false(2).

Frame Relay Network Service interfaces support the Interface Stack Group. Frame Relay Network Service interfaces do not support any other groups or objects in the Interfaces group of MIB II. Also, supporting the SNMP Group of MIB II is an implementation choice.

4. Object Definitions

```

FRNETSERV-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Counter32, Integer32,
    TimeTicks                               FROM SNMPv2-SMI
    DisplayString, PhysAddress,
    TimeStamp, RowStatus                    FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP        FROM SNMPv2-CONF
    ifIndex, transmission                   FROM RFC-1213;

frnetservMIB MODULE-IDENTITY
    LAST-UPDATED "9311161200Z"
    ORGANIZATION "IETF Frame Relay Network MIB Working Group"
    CONTACT-INFO
        " Tracy A. Brown
          Bellcore
          331 Newman Springs Rd.
          Red Bank, NJ 07701 USA

          Tel: 1-908-758-2107
          Fax: 1-908-758-4177
          E-mail: tacox@mail.bellcore.com."
    DESCRIPTION
        "The MIB module to describe generic objects for
         Frame Relay Network Service."
    ::= { transmission 44 }

IfIndex ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "The value of this object identifies the
         interface for which this entry contains
         management information. The value of this
         object for a particular interface has the same
         value as the ifIndex object, defined in RFC
         1213, for the same interface."
    SYNTAX Integer32

    frnetservObjects OBJECT IDENTIFIER ::= { frnetservMIB 1 }
    frnetservTraps   OBJECT IDENTIFIER ::= { frnetservMIB 2 }

-- The Frame Relay Service Logical Port Group
-- the Frame Relay Logical Port Group

```

```
-- This table is an interface specific addendum
-- to the generic ifTable from MIB-II.
```

```
frLportTable OBJECT-TYPE
SYNTAX SEQUENCE OF FrLportEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "The Frame Relay Logical Port Information table."
 ::= { frnetservObjects 1 }
```

```
frLportEntry OBJECT-TYPE
SYNTAX FrLportEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the Frame Relay Logical Port
    Information table."
INDEX { ifIndex }
 ::= { frLportTable 1 }
```

```
FrLportEntry ::=
SEQUENCE {
    frLportNumPlan
        INTEGER,
    frLportContact
        DisplayString,
    frLportLocation
        DisplayString,
    frLportType
        INTEGER,
    frLportAddrDLCILen
        INTEGER,
    frLportVCSigProtocol
        INTEGER,
    frLportVCSigPointer
        OBJECT IDENTIFIER
}
```

```
frLportNumPlan OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    e164(2),
    x121(3),
    none(4)
}
MAX-ACCESS read-only
```

```
STATUS current
DESCRIPTION
    "The value of this object identifies the network
    address numbering plan for this UNI/NNI logical
    port. The network address is the object
    ifPhysAddress. The value none implies that there
    is no ifPhysAddress. The SNMP proxy-agent will
    return an octet string of zero length for
    ifPhysAddress. The value other means that an
    address has been assigned to this interface, but
    the numbering plan is not enumerated here."
 ::= { frLportEntry 1 }

frLportContact OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the network
    contact for this UNI/NNI logical port."
 ::= { frLportEntry 2 }

frLportLocation OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the Frame
    Relay network location for this UNI/NNI logical
    port."
 ::= { frLportEntry 3 }

frLportType OBJECT-TYPE
SYNTAX INTEGER {
    uni(1),
    nni(2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the type of
    network interface for this logical port."
 ::= { frLportEntry 4 }

frLportAddrDLCILen OBJECT-TYPE
SYNTAX INTEGER {
    twoOctets10Bits(1),
    threeOctets10Bits(2),
```

```

    threeOctets16Bits(3),
    fourOctets17Bits(4),
    fourOctets23Bits(5)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the Q.922
    Address field length and DLCI length for this
    UNI/NNI logical port."
 ::= { frLportEntry 5 }

frLportVCSigProtocol OBJECT-TYPE
SYNTAX INTEGER {
    none(1),
    lmi(2),
    ansiT1617D(3),
    ansiT1617B(4),
    ccittQ933A(5)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the Local
    In-Channel Signaling Protocol that is used for
    this frame relay UNI/NNI logical port."
 ::= { frLportEntry 6 }

frLportVCSigPointer OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object is used as a pointer to
    the table that contains the Local In-Channel
    Signaling Protocol parameters and errors for this
    UNI/NNI logical port.  See the Frame Relay
    Management VC Signaling Parameters and Errors
    Group."
 ::= { frLportEntry 7 }

-- the Frame Relay Management VC Signaling Group

-- This Group contains managed objects for the
-- Local In-Channel Signaling Parameters and
-- for signaling errors.
```

```

frMgtVCSigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FrMgtVCSigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The Frame Relay Management VC Signaling
        Parameters and Errors table."
    ::= { frnetservObjects 2 }

frMgtVCSigEntry OBJECT-TYPE
    SYNTAX FrMgtVCSigEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the Frame Relay Management VC
        Signaling Parameters Errors table."
    INDEX { ifIndex }
    ::= { frMgtVCSigTable 1 }

FrMgtVCSigEntry ::=
    SEQUENCE {
        frMgtVCSigProced
            INTEGER,
        frMgtVCSigUserN391
            INTEGER,
        frMgtVCSigUserN392
            INTEGER,
        frMgtVCSigUserN393
            INTEGER,
        frMgtVCSigUserT391
            INTEGER,
        frMgtVCSigNetN392
            INTEGER,
        frMgtVCSigNetN393
            INTEGER,
        frMgtVCSigNetT392
            INTEGER,
        frMgtVCSigNetnN4
            INTEGER,
        frMgtVCSigNetnT3
            INTEGER,
        frMgtVCSigUserLinkRelErrors
            Counter32,
        frMgtVCSigUserProtErrors
            Counter32,
        frMgtVCSigUserChanInactive
            Counter32,
        frMgtVCSigNetLinkRelErrors
    }

```

```

        Counter32,
    frMgtVCSigNetProtErrors
        Counter32,
    frMgtVCSigNetChanInactive
        Counter32
}

```

frMgtVCSigProced OBJECT-TYPE

```

SYNTAX  INTEGER {
    u2nnet(1),
    bidirect(2)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the Local In-Channel Signaling Procedure that is used for this UNI/NNI logical port. The UNI/NNI logical port can be performing only user-to-network network-side procedures or bidirectional procedures."

```
 ::= { frMgtVCSigEntry 1 }
```

frMgtVCSigUserN391 OBJECT-TYPE

```
SYNTAX  INTEGER (1..255)
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the User-side N391 full status polling cycle value for this UNI/NNI logical port. If the logical port is not performing user-side procedures, then this value is equal to noSuchName. This object applies to Q.933 Annex A and T1.617 Annex D."

```
DEFVAL { 6 }
```

```
 ::= { frMgtVCSigEntry 2 }
```

frMgtVCSigUserN392 OBJECT-TYPE

```
SYNTAX  INTEGER (1..10)
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the User-side N392 error threshold value for this UNI/NNI logical port. If the logical port is not performing user-side procedures, then this value is equal to noSuchName. This object applies to

```
        Q.933 Annex A and T1.617 Annex D."
DEFVAL { 3 }
 ::= { frMgtVCSigEntry 3 }

frMgtVCSigUserN393 OBJECT-TYPE
SYNTAX  INTEGER (1..10)
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION
    "The value of this object identifies the User-side
    N393 monitored events count value for this UNI/NNI
    logical port.  If the logical port is not
    performing user-side procedures, then this value
    is equal to noSuchName. This object applies to
    Q.933 Annex A and T1.617 Annex D."
DEFVAL { 4 }
 ::= { frMgtVCSigEntry 4 }

frMgtVCSigUserT391 OBJECT-TYPE
SYNTAX  INTEGER (5..30)
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION
    "The value of this object identifies the User-side
    T391 link integrity verification polling timer
    value for this UNI/NNI logical port.  If the
    logical port is not performing user-side
    procedures, then this value is equal to
    noSuchName. This object applies to Q.933 Annex A
    and T1.617 Annex D."
DEFVAL { 10 }
 ::= { frMgtVCSigEntry 5 }

frMgtVCSigNetN392 OBJECT-TYPE
SYNTAX  INTEGER (1..10)
MAX-ACCESS  read-only
STATUS  current
DESCRIPTION
    "The value of this object identifies the Network-
    side N392 error threshold value (nN2 for LMI) for
    this UNI/NNI logical port.  If the logical port is
    not performing network-side procedures, then this
    value is equal to noSuchName. This object applies
    to Q.933 Annex A, T1.617 Annex D and LMI."
DEFVAL { 3 }
 ::= { frMgtVCSigEntry 6 }

frMgtVCSigNetN393 OBJECT-TYPE
```

SYNTAX INTEGER (1..10)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the Network-side N393 monitored events count value (nN3 for LMI) for this UNI/NNI logical port. If the logical port is not performing network-side procedures, then this value is equal to noSuchName. This object applies to Q.933 Annex A, T1.617 Annex D and LMI."

DEFVAL { 4 }

::= { frMgtVCSigEntry 7 }

frMgtVCSigNetT392 OBJECT-TYPE

SYNTAX INTEGER (5..30)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the Network-side T392 polling verification timer value (nT2 for LMI) for this UNI/NNI logical port. If the logical port is not performing network-side procedures, then this value is equal to noSuchName. This object applies to Q.933 Annex A, T1.617 Annex D and LMI."

DEFVAL { 15 }

::= { frMgtVCSigEntry 8 }

frMgtVCSigNetnN4 OBJECT-TYPE

SYNTAX INTEGER (5..5)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the Network-side nN4 maximum status enquires received value for this UNI/NNI logical port. If the logical port is not performing network-side procedures or is not performing LMI procedures, then this value is equal to noSuchName. This object applies only to LMI and always has a value of 5."

::= { frMgtVCSigEntry 9 }

frMgtVCSigNetnT3 OBJECT-TYPE

SYNTAX INTEGER (5 | 10 | 15 | 20 | 25 | 30)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the Network-side nT3 timer (for nN4 status enquires received) value for this UNI/NNI logical port. If the logical port is not performing network-side procedures or is not performing LMI procedures, then this value is equal to noSuchName. This object applies only to LMI."

DEFVAL { 20 }
 ::= { frMgtVCSigEntry 10 }

frMgtVCSigUserLinkRelErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of user-side local in-channel signaling link reliability errors (i.e., non-receipt of Status/Status Enquiry messages or invalid sequence numbers in a Link Integrity Verification Information Element) for this UNI/NNI logical port. If the logical port is not performing user-side procedures, then this value is equal to noSuchName."

::= { frMgtVCSigEntry 11 }

frMgtVCSigUserProtErrors OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of user-side local in-channel signaling protocol errors (i.e., protocol discriminator, message type, call reference, and mandatory information element errors) for this UNI/NNI logical port. If the logical port is not performing user-side procedures, then this value is equal to noSuchName."

::= { frMgtVCSigEntry 12 }

frMgtVCSigUserChanInactive OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The number of times the user-side channel was declared inactive (i.e., N392 errors in N393 events) for this UNI/NNI logical port. If the logical port is not performing user-side

```
        procedures, then this value is equal to
        noSuchName."
 ::= { frMgtVCSigEntry 13 }

frMgtVCSigNetLinkRelErrors OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of network-side local in-channel
    signaling link reliability errors (i.e., non-
    receipt of Status/Status Enquiry messages or
    invalid sequence numbers in a Link Integrity
    Verification Information Element) for this UNI/NNI
    logical port.  If the logical port is not
    performing network-side procedures, then this
    value is equal to noSuchName."
 ::= { frMgtVCSigEntry 14 }

frMgtVCSigNetProtErrors OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of network-side local in-channel
    signaling protocol errors (i.e., protocol
    discriminator, message type, call reference, and
    mandatory information element errors) for this
    UNI/NNI logical port.  If the logical port is not
    performing network-side procedures, then this
    value is equal to noSuchName."
 ::= { frMgtVCSigEntry 15 }

frMgtVCSigNetChanInactive OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of times the network-side channel was
    declared inactive (i.e., N392 errors in N393
    events) for this UNI/NNI logical port.  If the
    logical port is not performing network-side
    procedures, then this value is equal to
    noSuchName."
 ::= { frMgtVCSigEntry 16 }
```

```
-- The PVC End-Point Group

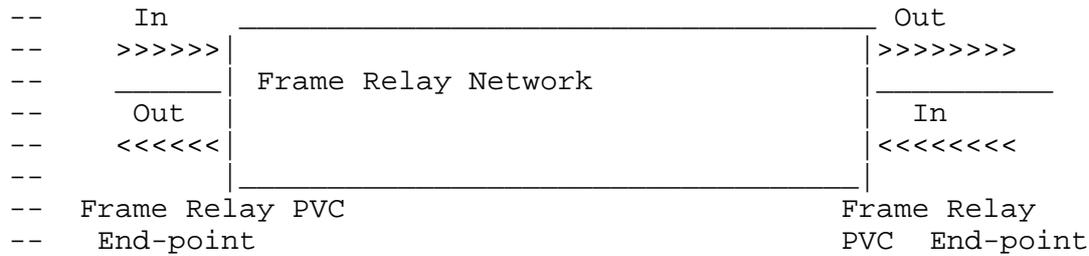
-- This table is used to identify the traffic parameters
-- for a bi-directional PVC segment end-point, and it also
-- provides statistics for a PVC segment
-- end-point.

-- A PVC segment end-point is identified by a UNI/NNI
-- logical port index value and DLCI index value.

-- If the Frame Relay service provider allows
-- the Frame Relay CNM subscriber to create, modify
-- or delete PVCs using SNMP, then this table is used to identify
-- and reserve
-- the requested traffic parameters of each
-- PVC segment end-point. The Connection table
-- is used to "connect" the end-points together.
-- Not all implementations will support the
-- capability of creating/modifying/deleting
-- PVCs using SNMP as a feature of Frame Relay
-- CNM service.

-- Uni-directional PVCs are modeled with zero
-- valued traffic parameters in one of the
-- directions (In or Out direction) in this table.

-- To create a PVC, the following procedures
-- shall be followed:
-- 1). Create the entries for the PVC segment endpoints in the
-- frPVCEndptTable by specifying the traffic parameters
-- for the bi-directional PVC segment endpoints.
-- As shown in the figure, a point-to-point PVC has
-- two endpoints, thus two entries in this table.
-- Uni-directional PVCs are modeled
-- with zero valued traffic parameters in one
-- direction; all the 'In' direction parameters
-- for one Frame Relay PVC End-point or
-- all the 'Out' direction
-- parameters for the other Frame Relay PVC
-- End-point.
```



```
-- 2). Go to the Frame Relay Connection Group.
```

```
--
```

```
-- The Frame Relay PVC End-point Table
```

```
frPVCEndptTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF FrPVCEndptEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

```
"The Frame Relay PVC End-Point table. This table
is used to model a PVC end-point. This table
contains the traffic parameters and statistics for
a PVC end-point."
```

```
::= { frnetservObjects 3 }
```

```
frPVCEndptEntry OBJECT-TYPE
```

```
SYNTAX FrPVCEndptEntry
```

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

```
STATUS current
```

```
DESCRIPTION
```

```
"An entry in the Frame Relay PVC Endpoint table."
```

```
INDEX { ifIndex, frPVCEndptDLCIIndex }
```

```
::= { frPVCEndptTable 1 }
```

```
FrPVCEndptEntry ::=
```

```
SEQUENCE {
```

```
frPVCEndptDLCIIndex
```

```
Integer32,
```

```
frPVCEndptInMaxFrameSize
```

```
Integer32,
```

```
frPVCEndptInBc
```

```
Integer32,
```

```
frPVCEndptInBe
```

```
Integer32,
```

```
frPVCEndptInCIR
```

```
Integer32,
```

```
frPVCEndptOutMaxFrameSize
```

```
Integer32,
```

```

    frPVCEndptOutBc
        Integer32,
    frPVCEndptOutBe
        Integer32,
    frPVCEndptOutCIR
        Integer32,
    frPVCEndptConnectIdentifier
        Integer32,
    frPVCEndptRowStatus
        RowStatus,
    frPVCEndptRcvdSigStatus
        INTEGER,
    frPVCEndptInFrames
        Counter32,
    frPVCEndptOutFrames
        Counter32,
    frPVCEndptInDEFrames
        Counter32,
    frPVCEndptInExcessFrames
        Counter32,
    frPVCEndptOutExcessFrames
        Counter32,
    frPVCEndptInDiscards
        Counter32,
    frPVCEndptInOctets
        Counter32,
    frPVCEndptOutOctets
        Counter32
}

```

frPVCEndptDLCIIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the DLCI value for this PVC end-point."

::= { frPVCEndptEntry 1 }

frPVCEndptInMaxFrameSize OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value of this object is the size in octets of the largest frame relay information field for this PVC end-point in the ingress direction (into the

```
frame relay network).  The value of
frPVCEndptInMaxFrameSize must be less than or
equal to the corresponding ifMtu for this Frame
Relay UNI/NNI logical port."
 ::= { frPVCEndptEntry 2 }

frPVCEndptInBc OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the
    committed burst size (Bc) parameter (measured in
    bits) for this PVC end-point in the ingress
    direction (into the frame relay network)."
```

```
 ::= { frPVCEndptEntry 3 }

frPVCEndptInBe OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the excess
    burst size (Be) parameter (measured in bits) for
    this PVC end-point in the ingress direction (into
    the frame relay network)."
```

```
 ::= { frPVCEndptEntry 4 }

frPVCEndptInCIR OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the
    committed information rate (CIR) parameter
    (measured in bits per second) for this PVC end-
    point in the ingress direction (into the frame
    relay network)."
```

```
 ::= { frPVCEndptEntry 5 }

frPVCEndptOutMaxFrameSize OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is the size in octets of
    the largest frame relay information field for this
    PVC end-point in the egress direction (out of the
```

```
frame relay network).  The value of
frPVCEndptOutMaxFrameSize must be less than or
equal to the corresponding ifMtu for this Frame
Relay UNI/NNI logical port."
 ::= { frPVCEndptEntry 6 }

frPVCEndptOutBc OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the
    committed burst size (Bc) parameter (measured in
    bits) for this PVC end-point in the egress
    direction (out of the frame relay network)."
```

```
 ::= { frPVCEndptEntry 7 }

frPVCEndptOutBe OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the excess
    burst size (Be) parameter (measured in bits) for
    this PVC end-point in the egress direction (out of
    the frame relay network)."
```

```
 ::= { frPVCEndptEntry 8 }

frPVCEndptOutCIR OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of this object is equal to the
    committed information rate (CIR) parameter
    (measured in bits per second) for this PVC end-
    point in the egress direction (out of the frame
    relay network)."
```

```
 ::= { frPVCEndptEntry 9 }

frPVCEndptConnectIdentifier OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "This object is used to associate PVC end-points
    as being part of one PVC segment connection.  This
    value of this object is equal to the value of
```

frPVCCConnectIndex, which is used as one of the indices into the frPVCCConnectTable. The value of this object is provided by the agent, after the associated entries in the frPVCCConnectTable have been created."

```
::= { frPVCEndptEntry 10 }
```

frPVCEndptRowStatus 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. To create a new PVC, the entries for the PVC segment end-points in the frPVCEndptTable must first be created. Next, the frPVCCConnectTable is used to associate the Frame Relay PVC segment end-points. In order for the manager to have the necessary error diagnostics, the frPVCEndptRowStatus object must initially be set to 'createAndWait'. While the frPVCEndptRowStatus object is in the 'createAndWait' state, the manager can set each columnar object and get the necessary error diagnostics. The frPVCEndptRowStatus object may not be set to 'active' unless the following columnar objects exist in this row:
frPVCEndptInMaxFrameSize, frPVCEndptInBc,
frPVCEndptInBe, frPVCEndptInCIR,
frPVCEndptOutMaxFrameSize, frPVCEndptOutBc,
frPVCEndptOutBe, and frPVCEndptOutCIR."

```
::= { frPVCEndptEntry 11 }
```

frPVCEndptRcvdSigStatus OBJECT-TYPE

SYNTAX INTEGER {
 deleted(1),
 active(2),
 inactive(3),
 none(4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object identifies the PVC status received via the local in-channel signaling procedures for this PVC end-point. This object is only pertinent for interfaces that perform the

bidirectional procedures. For user-to-network network side procedures, the value of this object should be none."

```
::= { frPVCEndptEntry 12 }
```

frPVCEndptInFrames OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of frames received by the network (ingress) for this PVC end-point. This includes any frames discarded by the network due to submitting more than Bc + Be data or due to any network congestion recovery procedures."

```
::= { frPVCEndptEntry 13 }
```

frPVCEndptOutFrames OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of frames sent by the network (egress) regardless of whether they are Bc or Be frames for this PVC end-point."

```
::= { frPVCEndptEntry 14 }
```

frPVCEndptInDEFrames OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of frames received by the network (ingress) with the DE bit set to (1) for this PVC end-point."

```
::= { frPVCEndptEntry 15 }
```

frPVCEndptInExcessFrames OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of frames received by the network (ingress) for this PVC end-point which were treated as excess traffic. Frames which are sent to the network with DE set to zero are treated as excess when more than Bc bits are submitted to the network during the Committed Information Rate

Measurement Interval (Tc). Excess traffic may or may not be discarded at the ingress if more than Bc + Be bits are submitted to the network during Tc. Traffic discarded at the ingress is not recorded in frPVCEndptInExcessFrames. Frames which are sent to the network with DE set to one are also treated as excess traffic."

::= { frPVCEndptEntry 16 }

frPVCEndptOutExcessFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of frames sent by the network (egress) for this PVC end-point which were treated as excess traffic. (The DE bit may be set to one.)"

::= { frPVCEndptEntry 17 }

frPVCEndptInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of frames received by the the network (ingress) that were discarded due to traffic enforcement for this PVC end-point."

::= { frPVCEndptEntry 18 }

frPVCEndptInOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets received by the network (ingress) for this PVC end-point. This counter should only count octets from the beginning of the frame relay header field to the end of user data. If the network supporting Frame Relay can not count octets, then this count should be an approximation."

::= { frPVCEndptEntry 19 }

frPVCEndptOutOctets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```
        "The number of octets sent by the network (egress)
        for this PVC end-point. This counter should only
        count octets from the beginning of the frame relay
        header field to the end of user data. If the
        network supporting Frame Relay can not count
        octets, then this count should be an
        approximation."
 ::= { frPVCEndptEntry 20 }

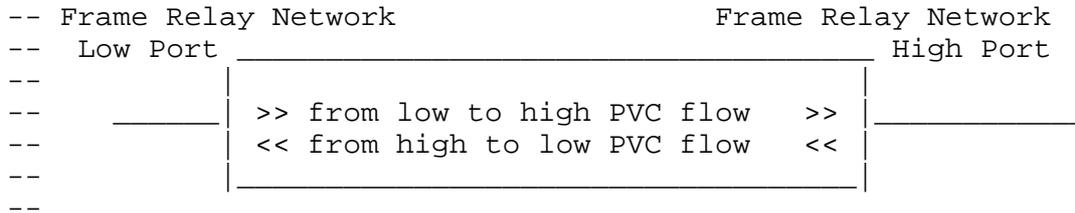
-- The Frame Relay PVC Connection Group

-- The Frame Relay PVC Connection Group
-- is used to model the bi-directional
-- PVC segment flows;
-- point-to-point PVCs, point-to-multipoint
-- PVCs, and multipoint-to-multipoint
-- PVCs.

-- This table has read-create access and
-- is used to associate PVC end-points
-- together as belonging to one connection.
-- The frPVCCConnectIndex is used to associate
-- all the bi-directional flows.
-- Not all implementations will support the
-- capability of creating/modifying/deleting
-- PVCs using SNMP as a feature of Frame Relay
-- CNM service.

-- Once the entries in the frPVCEndptTable
-- are created, the following step are used
-- to associate the PVC end-points as belonging
-- to one PVC connection:
-- 1). Obtain a unique frPVCCConnectIndex
--    using the frPVCCConnectIndexValue object.
-- 2). Connect the PVC segment endpoints together
--    with the applicable frPVCCConnectIndex value
--    obtained via
--    frPVCCConnectIndexValue.
--    The entries in this table are created by using
--    the frPVCCConnectRowStatus object.
-- 3). The agent will provide the value of the
--    corresponding instances of
--    frPVCEndptConnectIdentifier with the
--    the frPVCCConnectIndex value.
-- 4). Set frPVCCConnectAdminStatus to 'active' in all
--    rows for this PVC segment to
--    turn the PVC on.
```

```
-- For example, the Frame Relay PVC Connection Group
-- models a bi-directional, point-to-point PVC segment
-- as one entry in this table.
```



```
-- The terms low and high are chosen to represent numerical
-- ordering of a PVC segment's endpoints for representation
-- in this table. That is, the endpoint with the lower value
-- of ifIndex is termed 'low', while the opposite endpoint
-- of the segment is termed 'high'.
-- This terminology is to provide directional information;
-- for example the frPVCCConnectL2hOperStatus and
-- frPVCCConnectH2lOperStatus as illustrated above.
```

```
-- If the Frame Relay Connection table is used to model
-- a unidirectional PVC, then one direction (either from low to high
-- or from high to low) has its Operational Status equal to down.
```

```
-- A PVC segment is a portion of a PVC
-- that traverses one Frame Relay Network, and
-- a PVC segment is identified
-- by its two end-points (UNI/NNI logical port index
-- value and DLCI index value)
-- through one Frame Relay Network.
```

```
frPVCCConnectIndexValue OBJECT-TYPE
  SYNTAX  INTEGER (0..2147483647)
  MAX-ACCESS  read-only
  STATUS  current
  DESCRIPTION
    "This object contains an appropriate value to be
    used for frPVCCConnectIndex when creating entries
    in the frPVCCConnectTable. The value 0 indicates
    that no unassigned entries are available. To
    obtain the frPVCCConnectIndex value for a new
    entry, the manager issues a management protocol
    retrieval operation to obtain the current value of
    this object. After each retrieval, the agent
    should modify the value to the next unassigned
    index."
```

```

 ::= { frnetservObjects 4 }

-- The Frame Relay PVC Connection Table

frPVCConnectTable OBJECT-TYPE
SYNTAX SEQUENCE OF FrPVCConnectEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "The Frame Relay PVC Connect table. A bi-
    directional PVC segment is modeled as one entry in
    this table."
 ::= { frnetservObjects 5 }

frPVCConnectEntry OBJECT-TYPE
SYNTAX FrPVCConnectEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the Frame Relay PVC Connect table.
    This entry is used to model a PVC segment in two
    directions."
INDEX { frPVCConnectIndex,
        frPVCConnectLowIfIndex, frPVCConnectLowDLCIIndex,
        frPVCConnectHighIfIndex, frPVCConnectHighDLCIIndex }
 ::= { frPVCConnectTable 1 }

FrPVCConnectEntry ::=
SEQUENCE {
    frPVCConnectIndex
        Integer32,
    frPVCConnectLowIfIndex
        IfIndex,
    frPVCConnectLowDLCIIndex
        Integer32,
    frPVCConnectHighIfIndex
        IfIndex,
    frPVCConnectHighDLCIIndex
        Integer32,
    frPVCConnectAdminStatus
        INTEGER,
    frPVCConnectL2hOperStatus
        INTEGER,
    frPVCConnectH2lOperStatus
        INTEGER,
    frPVCConnectL2hLastChange
        TimeStamp,
    frPVCConnectH2lLastChange

```

```
        TimeStamp,  
        frPVCCConnectRowStatus  
        RowStatus  
    }
```

```
frPVCCConnectIndex OBJECT-TYPE
```

```
    SYNTAX Integer32
```

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

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "The value of this object is equal to the  
        frPVCCConnectIndexValue obtained to uniquely  
        identify this PVC segment connection."
```

```
    ::= { frPVCCConnectEntry 1 }
```

```
frPVCCConnectLowIfIndex OBJECT-TYPE
```

```
    SYNTAX IfIndex
```

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

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "The value of this object is equal to MIB II's  
        ifIndex value of the UNI/NNI logical port for this  
        PVC segment. The term low implies that this PVC  
        segment end-point has the numerically lower  
        ifIndex value than the connected/associated PVC  
        segment end-point. If the value is equal to zero,  
        then this logical port is not a Frame Relay  
        UNI/NNI logical port."
```

```
    ::= { frPVCCConnectEntry 2 }
```

```
frPVCCConnectLowDLCIIndex OBJECT-TYPE
```

```
    SYNTAX Integer32
```

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

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "The value of this object is equal to the DLCI  
        value for this end-point of the PVC segment. If  
        the value is equal to zero, then this endpoint of  
        the PVC segment is not a Frame Relay connection."
```

```
    ::= { frPVCCConnectEntry 3 }
```

```
frPVCCConnectHighIfIndex OBJECT-TYPE
```

```
    SYNTAX IfIndex
```

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

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "The value of this object is equal to MIB II's  
        ifIndex value for the UNI/NNI logical port for
```

this PVC segment. The term high implies that this PVC segment end-point has the numerically higher ifIndex value than the connected/associated PVC segment end-point."
 ::= { frPVCCConnectEntry 4 }

frPVCCConnectHighDLCIIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is equal to the egress DLCI value for this end-point of the PVC segment."

::= { frPVCCConnectEntry 5 }

frPVCCConnectAdminStatus OBJECT-TYPE

SYNTAX INTEGER {
 active(1),
 inactive(2),
 testing(3)
 }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value of this object identifies the desired administrative status of this bi-directional PVC segment. The active state means the PVC segment is currently operational; the inactive state means the PVC segment is currently not operational; the testing state means the PVC segment is currently undergoing a test. This state is set by an administrative entity. This value affects the PVC status indicated across the ingress NNI/UNI of both end-points of the bi-directional PVC segment. When a PVC segment connection is created using this table, this object is initially set to 'inactive'. After the frPVCCConnectRowStatus object is set to 'active' (and the corresponding/associated entries in the frPVCEndptTable have their frPVCEndptRowStatus object set to 'active'), the frPVCCConnectAdminStatus object may be set to 'active' to turn on the PVC segment connection."

::= { frPVCCConnectEntry 6 }

frPVCCConnectL2hOperStatus OBJECT-TYPE

SYNTAX INTEGER {
 active(1),

```

        inactive(2),
        testing(3),
        unknown(4)
    }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the current
    operational status of the PVC segment connection
    in one direction; (i.e., in the low to high
    direction). The active state means it is
    currently operational; the inactive state means it
    is currently not operational; the testing state
    means it is currently undergoing a test; the
    unknown state means the status of it currently can
    not be determined. This value affects the PVC
    status indicated across the ingress NNI/UNI (low
    side) of the PVC segment."
 ::= { frPVCCConnectEntry 7 }

```

```

frPVCCConnectH2lOperStatus OBJECT-TYPE
SYNTAX INTEGER {
    active(1),
    inactive(2),
    testing(3),
    unknown(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object identifies the current
    operational status of the PVC segment connection
    in one direction; (i.e., in the high to low
    direction). The active state means it is
    currently operational; the inactive state means it
    is currently not operational; the testing state
    means it is currently undergoing a test; the
    unknown state means the status of it currently can
    not be determined. This value affects the PVC
    status indicated across the ingress NNI/UNI (high
    side) of the PVC segment."
 ::= { frPVCCConnectEntry 8 }

```

```

frPVCCConnectL2hLastChange OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION

```

"The value of MIB II's sysUpTime object at the time this PVC segment entered its current operational state in the low to high direction. If the current state was entered prior to the last re-initialization of the proxy-agent, then this object contains a zero value."

::= { frPVCCConnectEntry 9 }

frPVCCConnectH2lLastChange OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of MIB II's sysUpTime object at the time this PVC segment entered its current operational state in the high to low direction. If the current state was entered prior to the last re-initialization of the proxy-agent, then this object contains a zero value."

::= { frPVCCConnectEntry 10 }

frPVCCConnectRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The status of this entry in the frPVCCConnectTable. This variable is used to create new connections for the PVC end-points and to change existing connections of the PVC end-points. This object must be initially set to 'createAndWait'. In this state, the agent checks the parameters in the associated entries in the frPVCEndptTable to verify that the PVC end-points can be connected (i.e., the In parameters for one PVC end-point are equal to the Out parameters for the other PVC end-point). This object can not be set to 'active' unless the following columnar object exist in this row: frPVCCConnectAdminStatus. The agent also supplies the associated value of frPVCCConnectIndex for the frPVCEndptConnectIdentifier instances. To turn on a PVC segment connection, the frPVCCConnectAdminStatus is set to 'active'."

::= { frPVCCConnectEntry 11 }

```

-- The Frame Relay Accounting Groups

-- The groups are the following:
-- Accounting on a PVC basis
-- Accounting on an Interface/Logical Port basis

-- The Accounting on a Frame Relay PVC basis Group

-- The accounting information is collected for a PVC
-- segment end-point.

frAccountPVCTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FrAccountPVCEnterY
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The Frame Relay Accounting PVC table. This table
        is used to perform accounting on a PVC segment
        end-point basis."
    ::= { frnetservObjects 6 }

frAccountPVCEnterY OBJECT-TYPE
    SYNTAX FrAccountPVCEnterY
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the Frame Relay Accounting PVC
        table."
    INDEX { ifIndex, frAccountPVCDLCIIndex }
    ::= { frAccountPVCTable 1 }

FrAccountPVCEnterY ::=
    SEQUENCE {
        frAccountPVCDLCIIndex
            Integer32,
        frAccountPVCSegmentSize
            Integer32,
        frAccountPVCInSegments
            Counter32,
        frAccountPVCOutSegments
            Counter32
    }

frAccountPVCDLCIIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current

```

DESCRIPTION

"The value of this object is equal to the DLCI value for this PVC segment end-point."

::= { frAccountPVCEntry 1 }

frAccountPVCSegmentSize OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is equal to the Segment Size for this PVC segment end-point."

::= { frAccountPVCEntry 2 }

frAccountPVCInSegments OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is equal to the number of segments received by this PVC segment end-point."

::= { frAccountPVCEntry 3 }

frAccountPVCOutSegments OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is equal to the number of segments sent by this PVC segment end-point."

::= { frAccountPVCEntry 4 }

-- The Accounting on a Frame Relay Logical Port basis Group

frAccountLportTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrAccountLportEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Frame Relay Accounting Logical Port table. This table is used to perform accounting on a UNI/NNI Logical Port basis."

::= { frnetservObjects 7 }

frAccountLportEntry OBJECT-TYPE

```
SYNTAX FrAccountLportEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "An entry in the Frame Relay Accounting Logical
    Port table."
INDEX { ifIndex }
 ::= { frAccountLportTable 1 }
```

```
FrAccountLportEntry ::=
SEQUENCE {
    frAccountLportSegmentSize
        Integer32,
    frAccountLportInSegments
        Counter32,
    frAccountLportOutSegments
        Counter32
}
```

```
frAccountLportSegmentSize OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object is equal to the Segment
    Size for this UNI/NNI logical port."
 ::= { frAccountLportEntry 1 }
```

```
frAccountLportInSegments OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object is equal to the number
    of segments received by this UNI/NNI logical
    port."
 ::= { frAccountLportEntry 2 }
```

```
frAccountLportOutSegments OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The value of this object is equal to the number
    of segments sent by this UNI/NNI logical port."
 ::= { frAccountLportEntry 3 }
```

-- Frame Relay Network Service TRAPS

```
frPVCCoconnectStatusChange NOTIFICATION-TYPE
  OBJECTS { frPVCCoconnectIndex,
            frPVCCoconnectLowIfIndex, frPVCCoconnectLowDLCIIndex,
            frPVCCoconnectHighIfIndex, frPVCCoconnectHighDLCIIndex,
            frPVCCoconnectL2hOperStatus, frPVCCoconnectH2lOperStatus,
            frPVCEndptRcvdSigStatus }
  STATUS current
  DESCRIPTION
    "This trap indicates that the indicated PVC has
    changed state. This trap is not sent if an FR-UNI
    changes state; a linkDown or linkUp trap should be
    sent instead."
  ::= { frnetservTraps 1 }
```

-- Conformance Information

```
frnetservConformance OBJECT IDENTIFIER ::= { frnetservMIB 3 }

frnetservGroups      OBJECT IDENTIFIER ::= { frnetservConformance 1 }
frnetservCompliances OBJECT IDENTIFIER ::= { frnetservConformance 2 }
```

-- Compliance Statements

```
frnetservCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for SNMPv2 entities
    which have Frame Relay Network Service Interfaces."

  MODULE -- this module
    MANDATORY-GROUPS { frnetservLportGroup,
                      frnetservMgtVCSigGroup,
                      frnetservPVCEndptGroup,
                      frnetservPVCCoconnectGroup }

  GROUP          frnetservAccountPVCGroup
  DESCRIPTION
    "This group is optional for Frame Relay interfaces.
    It is
    mandatory if and only if accounting is performed
    on a PVC
    basis this
    Frame Relay interface."

  GROUP          frnetservAccountLportGroup
```

DESCRIPTION

"This group is optional for Frame Relay interfaces.
It is
mandatory if and only if accounting is
performed on a
logical port basis this
Frame Relay interface."

OBJECT frPVCEndptInMaxFrameSize
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptInBc
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptInBe
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptInCIR
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptOutMaxFrameSize
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptOutBc
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION
"Write access is not required."

OBJECT frPVCEndptOutBe
SYNTAX Integer32
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT frPVCEndptOutCIR
 SYNTAX Integer32
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required."

OBJECT frPVCEndptConnectIdentifier
 SYNTAX Integer32
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required."

OBJECT frPVCEndptRowStatus
 SYNTAX INTEGER { active(1) } -- subset of RowStatus
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

OBJECT frPVCCConnectAdminStatus
 SYNTAX INTEGER {
 active(1),
 inactive(2),
 testing(3)
 }
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required."

OBJECT frPVCCConnectRowStatus
 SYNTAX INTEGER { active(1) } -- subset of RowStatus
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required."

OBJECT frPVCCConnectRowStatus
 SYNTAX INTEGER { active(1) } -- subset of RowStatus
 MIN-ACCESS read-only
 DESCRIPTION

"Write access is not required, and only one of the six enumerated values for the RowStatus textual convention need be supported, specifically: active(1)."

::= { frnetservCompliances 1 }

-- Units of Conformance

```
frnetservLportGroup OBJECT-GROUP
  OBJECTS { frLportNumPlan, frLportContact, frLportLocation,
            frLportType,
            frLportAddrDLCILen, frLportVCSigProtocol,
            frLportVCSigPointer }
  STATUS current
  DESCRIPTION
    "A collection of objects providing information applicable
    to a Frame Relay Logical Port."
  ::= { frnetservGroups 1 }
```

```
frnetservMgtVCSigGroup OBJECT-GROUP
  OBJECTS { frMgtVCSigProced,
            frMgtVCSigUserN391,
            frMgtVCSigUserN392,
            frMgtVCSigUserN393,
            frMgtVCSigUserT391,
            frMgtVCSigNetN392,
            frMgtVCSigNetN393,
            frMgtVCSigNetT392,
            frMgtVCSigNetnN4,
            frMgtVCSigNetnT3,
            frMgtVCSigUserLinkRelErrors,
            frMgtVCSigUserProtErrors,
            frMgtVCSigUserChanInactive,
            frMgtVCSigNetLinkRelErrors,
            frMgtVCSigNetProtErrors,
            frMgtVCSigNetChanInactive }
  STATUS current
  DESCRIPTION
    "A collection of objects providing information
    applicable to the
    Local In-Channel Signaling Procedures used for a
    UNI/NNI logical port."
  ::= { frnetservGroups 2 }
```

```
frnetservPVCEndptGroup OBJECT-GROUP
  OBJECTS { frPVCCoconnectIndexValue,
            frPVCEndptInMaxFrameSize, frPVCEndptInBc,
            frPVCEndptInBe, frPVCEndptInCIR,
            frPVCEndptOutMaxFrameSize, frPVCEndptOutBc,
            frPVCEndptOutBe, frPVCEndptOutCIR,
            frPVCEndptConnectIdentifier, frPVCEndptRowStatus,
            frPVCEndptRcvdSigStatus, frPVCEndptInFrames,
            frPVCEndptOutFrames, frPVCEndptInDEFrames,
            frPVCEndptInExcessFrames, frPVCEndptOutExcessFrames,
```

```
        frPVCEndptInDiscards,
        frPVCEndptInOctets, frPVCEndptOutOctets }
STATUS    current
DESCRIPTION
    "A collection of objects providing information application
    to a Frame Relay PVC end-point."
 ::= { frnetservGroups 3 }

frnetservPVCConnectGroup OBJECT-GROUP
OBJECTS { frPVCConnectAdminStatus, frPVCConnectL2hOperStatus,
          frPVCConnectH2lOperStatus, frPVCConnectL2hLastChange,
          frPVCConnectH2lLastChange,
          frPVCConnectRowStatus }
STATUS    current
DESCRIPTION
    "A collection of objects providing information applicable
    to a Frame Relay PVC connection."
 ::= { frnetservGroups 4 }

frnetservAccountPVCGroup OBJECT-GROUP
OBJECTS { frAccountPVCSegmentSize, frAccountPVCInSegments,
          frAccountPVCOutSegments }
STATUS    current
DESCRIPTION
    "A collection of objects providing accounting
    information application
    to a Frame Relay PVC end-point."
 ::= { frnetservGroups 5 }

frnetservAccountLportGroup OBJECT-GROUP
OBJECTS { frAccountLportSegmentSize, frAccountLportInSegments,
          frAccountLportOutSegments }
STATUS    current
DESCRIPTION
    "A collection of objects providing accounting
    information application
    to a Frame Relay logical port."
 ::= { frnetservGroups 6 }
```

END

7. Acknowledgments

This document was produced jointly by the Frame Relay Forum Technical Committee MIB Working Group and the Frame Relay Service MIB Working Group.

8. References

- [1] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [2] Galvin, J., and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Trusted Information Systems, Hughes LAN Systems, April 1993.
- [3] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [4] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets - MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.
- [5] McCloghrie, K., and F. Kastenholtz, "Evolution of Interfaces Group of MIB-II", RFC 1573, Hughes LAN Systems, FTP Software, January 1994.
- [6] Rodemann, K., "Service Management Architecture for Virtual Connection Services", Work in Progress, July 1993.
- [7] ANSI T1.617-1991, American National Standard for Telecommunications - Integrated Services Digital Network (ISDN) - Digital Subscriber Signaling System No. 1 (DSS1) - Signaling Specification for Frame Relay Bearer Service.
- [8] Brown, C., Baker, F., and C. Carvalho, "Management Information Base for Frame Relay DTEs", RFC 1315, Wellfleet Communications, Inc., Advanced Computer Communications, April 1992.
- [9] Bradley, T., Brown, C., and A. Malis, "Multi-Protocol Interconnect over Frame Relay", RFC 1490, Wellfleet Communications, Inc., Ascom Timeplex, Inc., July 1993.

9. Security Considerations

Security issues are not discussed in this memo.

10. Author's Address

Tracy A. Brown
Bell Communications Research
331 Newman Springs Road
P.O. Box 7020
Red Bank, NJ 07701-7020

Phone: (908) 758-2107
EMail: tacox@mail.bellcore.com

