

Network Working Group
Request for Comments: 1512
Updates: 1285

J. Case
The University of Tennessee and
SNMP Research, Incorporated
A. Rijsinghani
Digital Equipment Corporation
September 1993

FDDI Management Information Base

Status of this Memo

This RFC 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" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing devices which implement the FDDI based on the ANSI FDDI SMT 7.3 draft standard [8], which has been forwarded for publication by the X3T9.5 committee.

Table of Contents

1. The Network Management Framework	2
1.1 Object Definitions	2
1.2 Format of Definitions	2
2. Overview	2
2.1 Textual Conventions	3
3. Changes from RFC 1285	3
4. Object Definitions	4
4.1 The SMT Group	6
4.2 The MAC Group	17
4.3 The Enhanced MAC Counters Group	29
4.4 The PATH Group	32
4.5 The PORT Group	38
5. Acknowledgements	48
6. References	50
7. Security Considerations	51
8. Authors' Addresses	51

1. The Network Management Framework

The Internet-standard Network Management Framework consists of three components. They are:

- o STD 16, RFC 1155 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management. STD 16, RFC 1212 defines a more concise description mechanism, which is wholly consistent with the SMI.
- o STD 17, RFC 1213 defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- o STD 15, RFC 1157 which defines the SNMP, the protocol used for network access to managed objects.

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

1.1. 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.

1.2. Format of Definitions

Section 4 contains contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [7].

2. Overview

This document defines the managed objects for FDDI devices which are to be accessible via the Simple Network Management Protocol (SNMP). At present, this applies to these values of the ifType variable in the Internet-standard MIB:

fddi(15)

For these interfaces, the value of the ifSpecific variable in the

MIB-II [4] has the OBJECT IDENTIFIER value:

```
fdimib    OBJECT IDENTIFIER ::= { fddi 73 }
```

The definitions of the objects presented here draws heavily from related work in the ANSI X3T9.5 committee and the SMT subcommittee of that committee [8]. In fact, the definitions of the managed objects in this document are, to the maximum extent possible, identical to those identified by the ANSI committee. The semantics of each managed object should be the same with syntactic changes made as necessary to recast the objects in terms of the Internet-standard SMI and MIB so as to be compatible with the SNMP. Examples of these syntactic changes include remapping booleans to enumerated integers, remapping bit strings to octet strings, and the like. In addition, the naming of the objects was changed to achieve compatibility.

These minimal syntactic changes with no semantic changes should allow implementations of SNMP manageable FDDI systems to share instrumentation with other network management schemes and thereby minimize implementation cost. In addition, the translation of information conveyed by managed objects from one network management scheme to another is eased by these shared definitions.

Only the essential variables, as indicated by their mandatory status in the ANSI specification, were retained in this document. The importance of variables which have an optional status in the ANSI specification were perceived as being less widely accepted.

2.1. Textual Conventions

Several new datatypes are introduced as a textual convention in this MIB document. These textual conventions enhance the readability of the document and ease comparisons with its ANSI counterpart. It should be noted that the introduction of these textual conventions has no effect on either the syntax or the semantics of any managed objects. The use of these is merely an artifact of the explanatory method used. Objects defined in terms of one of these methods are always encoded by means of the rules that define the primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate these textual conventions which are adopted merely for the convenience of readers and writers in pursuit of the elusive goal of clear, concise, and unambiguous MIB documents.

3. Changes from RFC 1285

The changes from RFC 1285 [2] to this document, based on changes from ANSI SMT 6.2 to SMT 7.3, were so numerous that the objects in this MIB module are located on a different branch of the MIB tree. No

assumptions should be made about compatibility with RFC 1285.

4. Object Definitions

```
FDDI-SMT73-MIB DEFINITIONS ::= BEGIN

IMPORTS
    Counter
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212;

-- This MIB module uses the extended OBJECT-TYPE macro as
-- defined in [7].

-- this is the FDDI MIB module

fddi      OBJECT IDENTIFIER ::= { transmission 15 }
fddimib   OBJECT IDENTIFIER ::= { fddi 73 }

-- textual conventions

FddiTimeNano ::= INTEGER (0..2147483647)
-- This data type specifies 1 nanosecond units as
-- an integer value.
--
-- NOTE: The encoding is normal integer representation, not
-- two's complement. Since this type is used for variables
-- which are encoded as TimerTwosComplement in the ANSI
-- specification, two operations need to be performed on such
-- variables to convert from ANSI form to SNMP form:
--
-- 1) Convert from two's complement to normal integer
--    representation
-- 2) Multiply by 80 to convert from 80 nsec to 1 nsec units
--
-- No resolution is lost. Moreover, the objects for which
-- this data type is used effectively do not lose any range
-- due to the lower maximum value since they do not require
-- the full range.
--
-- Example: If fddimibMACTReq had a value of 8 ms, it would
-- be stored in ANSI TimerTwosComplement format as 0xFFFFE7960
-- [8 ms is 100000 in 80 nsec units, which is then converted
-- to two's complement] but be reported as 8000000 in SNMP
-- since it is encoded here as FddiTimeNano.
```

```
FddiTimeMilli ::= INTEGER (0..2147483647)
-- This data type is used for some FDDI timers. It specifies
-- time in 1 millisecond units, in normal integer
-- representation.

FddiResourceId ::= INTEGER (0..65535)
-- This data type is used to refer to an instance of a MAC,
-- PORT, or PATH Resource ID. Indexing begins
-- at 1. Zero is used to indicate the absence of a resource.

FddiSMTStationIdType ::= OCTET STRING (SIZE (8))
-- The unique identifier for the FDDI station. This is a
-- string of 8 octets, represented as X' yy yy xx xx xx xx
-- xx xx' with the low order 6 octet (xx) from a unique IEEE
-- assigned address. The high order two bits of the IEEE
-- address, the group address bit and the administration bit
-- (Universal/Local) bit should both be zero. The first two
-- octets, the yy octets, are implementor-defined.
--
-- The representation of the address portion of the station id
-- is in the IEEE (ANSI/IEEE P802.1A) canonical notation for
-- 48 bit addresses. The canonical form is a 6-octet string
-- where the first octet contains the first 8 bits of the
-- address, with the I/G(Individual/Group) address bit as the
-- least significant bit and the U/L (Universal/Local) bit
-- as the next more significant bit, and so on. Note that
-- addresses in the ANSI FDDI standard SMT frames are
-- represented in FDDI MAC order.

FddiMACLongAddressType ::= OCTET STRING (SIZE (6))
-- The representation of long MAC addresses as management
-- values is in the IEEE (ANSI/IEEE P802.1A) canonical
-- notation for 48 bit addresses. The canonical form is a
-- 6-octet string where the first octet contains the first 8
-- bits of the address, with the I/G (Individual/Group)
-- address bit as the least significant bit and the U/L
-- (Universal/Local) bit as the next more significant bit,
-- and so on. Note that the addresses in the SMT frames are
-- represented in FDDI MAC order.

-- groups in the FDDI MIB module

fddimibSMT          OBJECT IDENTIFIER ::= { fddimib 1 }

fddimibMAC          OBJECT IDENTIFIER ::= { fddimib 2 }

fddimibMACCounters  OBJECT IDENTIFIER ::= { fddimib 3 }
```

```
fddimibPATH          OBJECT IDENTIFIER ::= { fddimib 4 }

fddimibPORT          OBJECT IDENTIFIER ::= { fddimib 5 }

-- the SMT group
-- Implementation of the SMT group is mandatory for all
-- systems which implement manageable FDDI subsystems.

fddimibSMTNumber OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of SMT implementations (regardless of
        their current state) on this network management
        application entity.  The value for this variable
        must remain constant at least from one re-
        initialization of the entity's network management
        system to the next re-initialization."
    ::= { fddimibSMT 1 }

-- the SMT table

fddimibSMTTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FddimibSMTEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of SMT entries.  The number of entries
        shall not exceed the value of fddimibSMTNumber."
    ::= { fddimibSMT 2 }

fddimibSMTEntry OBJECT-TYPE
    SYNTAX  FddimibSMTEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "An SMT entry containing information common to a
        given SMT."
    INDEX   { fddimibSMTIndex }
    ::= { fddimibSMTTable 1 }

FddimibSMTEntry ::=
    SEQUENCE {
        fddimibSMTIndex
        INTEGER,
```

```
fddimibSMTStationId
    FddiSMTStationIdType,
fddimibSMTOpVersionId
    INTEGER,
fddimibSMTHiVersionId
    INTEGER,
fddimibSMTLoVersionId
    INTEGER,
fddimibSMTUserData
    OCTET STRING,
fddimibSMTMIBVersionId
    INTEGER,
fddimibSMTMACCts
    INTEGER,
fddimibSMTNonMasterCts
    INTEGER,
fddimibSMTMasterCts
    INTEGER,
fddimibSMTAvailablePaths
    INTEGER,
fddimibSMTConfigCapabilities
    INTEGER,
fddimibSMTConfigPolicy
    INTEGER,
fddimibSMTConnectionPolicy
    INTEGER,
fddimibSMTTNotify
    INTEGER,
fddimibSMTStatRptPolicy
    INTEGER,
fddimibSMTTraceMaxExpiration
    FddiTimeMilli,
fddimibSMTBypassPresent
    INTEGER,
fddimibSMTECMState
    INTEGER,
fddimibSMTCFState
    INTEGER,
fddimibSMTRemoteDisconnectFlag
    INTEGER,
fddimibSMTStationStatus
    INTEGER,
fddimibSMTPeerWrapFlag
    INTEGER,
fddimibSMTTimeStamp
    FddiTimeMilli,
fddimibSMTTransitionTimeStamp
    FddiTimeMilli,
```

```
        fddimibSMTStationAction
            INTEGER
    }

fddimibSMTIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "A unique value for each SMT.  The value for each
        SMT must remain constant at least from one re-
        initialization of the entity's network management
        system to the next re-initialization."
    ::= { fddimibSMTEntry 1 }

fddimibSMTStationId OBJECT-TYPE
    SYNTAX  FddiSMTStationIdType -- OCTET STRING (SIZE (8))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "Used to uniquely identify an FDDI station."
    REFERENCE
        "ANSI { fddiSMT 11 }"
    ::= { fddimibSMTEntry 2 }

fddimibSMTOpVersionId OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The version that this station is using for its
        operation (refer to ANSI 7.1.2.2). The value of
        this variable is 2 for this SMT revision."
    REFERENCE
        "ANSI { fddiSMT 13 }"
    ::= { fddimibSMTEntry 3 }

fddimibSMTHiVersionId OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The highest version of SMT that this station
        supports (refer to ANSI 7.1.2.2)."
    REFERENCE
        "ANSI { fddiSMT 14 }"
    ::= { fddimibSMTEntry 4 }
```



```
fddimibSMTLoVersionId OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The lowest version of SMT that this station
        supports (refer to ANSI 7.1.2.2)."
```

```
REFERENCE
    "ANSI { fddiSMT 15 }"
 ::= { fddimibSMTEntry 5 }
```

```
fddimibSMTUserData OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (32))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "This variable contains 32 octets of user defined
        information. The information shall be an ASCII
        string."
```

```
REFERENCE
    "ANSI { fddiSMT 17 }"
 ::= { fddimibSMTEntry 6 }
```

```
fddimibSMTMIBVersionId OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The version of the FDDI MIB of this station.  The
        value of this variable is 1 for this SMT
        revision."
```

```
REFERENCE
    "ANSI { fddiSMT 18 }"
 ::= { fddimibSMTEntry 7 }
```

```
fddimibSMTMACcts OBJECT-TYPE
    SYNTAX  INTEGER (0..255)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of MACs in this station or
        concentrator."
```

```
REFERENCE
    "ANSI { fddiSMT 21 }"
 ::= { fddimibSMTEntry 8 }
```

```
fddimibSMTNonMasterCts OBJECT-TYPE
    SYNTAX  INTEGER (0..2)
```

ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The value of this variable is the number of A, B,
 and S ports in this station or concentrator."
 REFERENCE
 "ANSI { fddiSMT 22 }"
 ::= { fddimibSMTEntry 9 }

fddimibSMTMasterCts OBJECT-TYPE

SYNTAX INTEGER (0..255)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of M Ports in a node. If the node is
 not a concentrator, the value of the variable is
 zero."
 REFERENCE
 "ANSI { fddiSMT 23 }"
 ::= { fddimibSMTEntry 10 }

fddimibSMTAvailablePaths OBJECT-TYPE

SYNTAX INTEGER (0..7)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "A value that indicates the PATH types available
 in the station.

 The value is a sum. This value initially takes
 the value zero, then for each type of PATH that
 this node has available, 2 raised to a power is
 added to the sum. The powers are according to the
 following table:

Path	Power
Primary	0
Secondary	1
Local	2

 For example, a station having Primary and Local
 PATHs available would have a value of 5 ($2^{**0} +$
 2^{**2}).

REFERENCE
 "ANSI { fddiSMT 24 }"
 ::= { fddimibSMTEntry 11 }

fddimibSMTConfigCapabilities OBJECT-TYPE

SYNTAX INTEGER (0..3)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A value that indicates the configuration capabilities of a node. The 'Hold Available' bit indicates the support of the optional Hold Function, which is controlled by fddiSMTConfigPolicy. The 'CF-Wrap-AB' bit indicates that the station has the capability of performing a wrap_ab (refer to ANSI SMT 9.7.2.2).

The value is a sum. This value initially takes the value zero, then for each of the configuration policies currently enforced on the node, 2 raised to a power is added to the sum. The powers are according to the following table:

Policy	Power
holdAvailable	0
CF-Wrap-AB	1 "

REFERENCE

"ANSI { fddiSMT 25 }"
 ::= { fddimibSMTEntry 12 }

fddimibSMTConfigPolicy OBJECT-TYPE

SYNTAX INTEGER (0..1)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"A value that indicates the configuration policies currently desired in a node. 'Hold' is one of the terms used for the Hold Flag, an optional ECM flag used to enable the optional Hold policy.

The value is a sum. This value initially takes the value zero, then for each of the configuration policies currently enforced on the node, 2 raised to a power is added to the sum. The powers are according to the following table:

Policy	Power
configurationhold	0 "

REFERENCE

"ANSI { fddiSMT 26 }"
 ::= { fddimibSMTEntry 13 }

fddimibSMTConnectionPolicy OBJECT-TYPE

SYNTAX INTEGER (32768..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"A value representing the connection policies in effect in a node. A station sets the corresponding bit for each of the connection types that it rejects. The letter designations, X and Y, in the 'rejectX-Y' names have the following significance: X represents the PC-Type of the local PORT and Y represents the PC_Type of the adjacent PORT (PC_Neighbor). The evaluation of Connection-Policy (PC-Type, PC-Neighbor) is done to determine the setting of T- Val(3) in the PC-Signalling sequence (refer to ANSI 9.6.3). Note that Bit 15, (rejectM-M), is always set and cannot be cleared.

The value is a sum. This value initially takes the value zero, then for each of the connection policies currently enforced on the node, 2 raised to a power is added to the sum. The powers are according to the following table:

Policy	Power
rejectA-A	0
rejectA-B	1
rejectA-S	2
rejectA-M	3
rejectB-A	4
rejectB-B	5
rejectB-S	6
rejectB-M	7
rejectS-A	8
rejectS-B	9
rejectS-S	10
rejectS-M	11
rejectM-A	12
rejectM-B	13
rejectM-S	14
rejectM-M	15 "

REFERENCE

"ANSI { fddiSMT 27 }"
 ::= { fddimibSMTEntry 14 }

fddimibSMTTNotify OBJECT-TYPE

SYNTAX INTEGER (2..30)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The timer, expressed in seconds, used in the Neighbor Notification protocol. It has a range of 2 seconds to 30 seconds, and its default value is 30 seconds (refer to ANSI SMT 8.2)."

REFERENCE

"ANSI { fddiSMT 29 }"
 ::= { fddimibSMTEntry 15 }

fddimibSMTStatRptPolicy OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"If true, indicates that the node will generate Status Reporting Frames for its implemented events and conditions. It has an initial value of true. This variable determines the value of the SR_Enable Flag (refer to ANSI SMT 8.3.2.1)."

REFERENCE

"ANSI { fddiSMT 30 }"
 ::= { fddimibSMTEntry 16 }

fddimibSMTTraceMaxExpiration OBJECT-TYPE

SYNTAX FddiTimeMilli

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Reference Trace_Max (refer to ANSI SMT 9.4.4.2.2)."

REFERENCE

"ANSI { fddiSMT 31 }"
 ::= { fddimibSMTEntry 17 }

fddimibSMTBypassPresent OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A flag indicating if the station has a bypass on its AB port pair."

REFERENCE

"ANSI { fddiSMT 34 }"
 ::= { fddimibSMTEntry 18 }

fddimibSMTECMState OBJECT-TYPE

SYNTAX INTEGER {
 ec0(1), -- Out

```

        ec1(2), -- In
        ec2(3), -- Trace
        ec3(4), -- Leave
        ec4(5), -- Path_Test
        ec5(6), -- Insert
        ec6(7), -- Check
        ec7(8)  -- Deinsert
    }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "Indicates the current state of the ECM state
    machine (refer to ANSI SMT 9.5.2)."
```

REFERENCE

```

    "ANSI { fddiSMT 41 }"
 ::= { fddimibSMTEntry 19 }
```

fddimibSMTCFState OBJECT-TYPE

```

SYNTAX  INTEGER {
    cf0(1),  -- isolated
    cf1(2),  -- local_a
    cf2(3),  -- local_b
    cf3(4),  -- local_ab
    cf4(5),  -- local_s
    cf5(6),  -- wrap_a
    cf6(7),  -- wrap_b
    cf7(8),  -- wrap_ab
    cf8(9),  -- wrap_s
    cf9(10), -- c_wrap_a
    cf10(11), -- c_wrap_b
    cf11(12), -- c_wrap_s
    cf12(13) -- thru
}
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "The attachment configuration for the station or
    concentrator (refer to ANSI SMT 9.7.2.2)."
```

REFERENCE

```

    "ANSI { fddiSMT 42 }"
 ::= { fddimibSMTEntry 20 }
```

fddimibSMTRemoteDisconnectFlag OBJECT-TYPE

```

SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "A flag indicating that the station was remotely
```

disconnected from the network as a result of receiving an fddiSMTAction, disconnect (refer to ANSI SMT 6.4.5.3) in a Parameter Management Frame. A station requires a Connect Action to rejoin and clear the flag (refer to ANSI SMT 6.4.5.2)."

REFERENCE

"ANSI { fddiSMT 44 }"
 ::= { fddimibSMTEntry 21 }

fddimibSMTStationStatus OBJECT-TYPE

SYNTAX INTEGER { concatenated(1), separated(2), thru(3) }
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The current status of the primary and secondary paths within this station."

REFERENCE

"ANSI { fddiSMT 45 }"
 ::= { fddimibSMTEntry 22 }

fddimibSMTPeerWrapFlag OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This variable assumes the value of the PeerWrapFlag in CFM (refer to ANSI SMT 9.7.2.4.4)."

REFERENCE

"ANSI { fddiSMT 46 }"
 ::= { fddimibSMTEntry 23 }

fddimibSMTTimeStamp OBJECT-TYPE

SYNTAX FddiTimeMilli
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This variable assumes the value of TimeStamp (refer to ANSI SMT 8.3.2.1)."

REFERENCE

"ANSI { fddiSMT 51 }"
 ::= { fddimibSMTEntry 24 }

fddimibSMTTransitionTimeStamp OBJECT-TYPE

SYNTAX FddiTimeMilli
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This variable assumes the value of
TransitionTimeStamp (refer to ANSI SMT 8.3.2.1)."

REFERENCE

"ANSI { fddiSMT 52 }"
::= { fddimibSMTEntry 25 }

fddimibSMTStationAction OBJECT-TYPE

```
SYNTAX  INTEGER {
    other(1),          -- none of the following
    connect(2),
    disconnect(3),
    path-Test(4),
    self-Test(5),
    disable-a(6),
    disable-b(7),
    disable-m(8)
}
```

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object, when read, always returns a value of
other(1). The behavior of setting this variable
to each of the acceptable values is as follows:

other(1): Results in an appropriate error.
connect(2): Generates a Connect signal to ECM
to begin a connection sequence. See ANSI
Ref 9.4.2.
disconnect(3): Generates a Disconnect signal
to ECM. see ANSI Ref 9.4.2.
path-Test(4): Initiates a station Path_Test.
The Path_Test variable (see ANSI Ref
9.4.1) is set to 'Testing'. The results
of this action are not specified in this
standard.
self-Test(5): Initiates a station Self_Test.
The results of this action are not
specified in this standard.
disable-a(6): Causes a PC_Disable on the A
port if the A port mode is peer.
disable-b(7): Causes a PC_Disable on the B
port if the B port mode is peer.
disable-m(8): Causes a PC_Disable on all M
ports.

Attempts to set this object to all other values
results in an appropriate error. The result of
setting this variable to path-Test(4) or self-


```

        Test(5) is implementation-specific."
REFERENCE
    "ANSI { fddiSMT 60 }"
 ::= { fddimibSMTEntry 26 }

-- the MAC group
-- Implementation of the MAC Group is mandatory for all
-- systems which implement manageable FDDI subsystems.

fddimibMACNumber OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of MAC implementations (across
        all SMTs) on this network management application
        entity. The value for this variable must remain
        constant at least from one re-initialization of
        the entity's network management system to the next
        re-initialization."
    ::= { fddimibMAC 1 }

-- the MAC table

fddimibMACTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FddimibMACEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of MAC entries. The number of entries
        shall not exceed the value of fddimibMACNumber."
    ::= { fddimibMAC 2 }

fddimibMACEntry OBJECT-TYPE
    SYNTAX  FddimibMACEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A MAC entry containing information common to a
        given MAC."
    INDEX   { fddimibMACSMTIndex, fddimibMACIndex }
    ::= { fddimibMACTable 1 }

FddimibMACEntry ::=
    SEQUENCE {
        fddimibMACSMTIndex

```

```
        INTEGER,
fddimibMACIndex
        INTEGER,
fddimibMACIfIndex
        INTEGER,
fddimibMACFrameStatusFunctions
        INTEGER,
fddimibMACTMaxCapability
        FddiTimeNano,
fddimibMACTVXCapability
        FddiTimeNano,
fddimibMACAvailablePaths
        INTEGER,
fddimibMACCurrentPath
        INTEGER,
fddimibMACUpstreamNbr
        FddiMACLongAddressType,
fddimibMACDownstreamNbr
        FddiMACLongAddressType,
fddimibMACOldUpstreamNbr
        FddiMACLongAddressType,
fddimibMACOldDownstreamNbr
        FddiMACLongAddressType,
fddimibMACDupAddressTest
        INTEGER,
fddimibMACRequestedPaths
        INTEGER,
fddimibMACDownstreamPORTType
        INTEGER,
fddimibMACSMTAddress
        FddiMACLongAddressType,
fddimibMACTReq
        FddiTimeNano,
fddimibMACTNeg
        FddiTimeNano,
fddimibMACTMax
        FddiTimeNano,
fddimibMACTvxValue
        FddiTimeNano,
fddimibMACFrameCts
        Counter,
fddimibMACCopiedCts
        Counter,
fddimibMACTransmitCts
        Counter,
fddimibMACErrorCts
        Counter,
fddimibMACLostCts
```

```

        Counter,
fddimibMACFrameErrorThreshold
        INTEGER,
fddimibMACFrameErrorRatio
        INTEGER,
fddimibMACRMTState
        INTEGER,
fddimibMACDaFlag
        INTEGER,
fddimibMACUnaDaFlag
        INTEGER,
fddimibMACFrameErrorFlag
        INTEGER,
fddimibMACMAUnitdataAvailable
        INTEGER,
fddimibMACHardwarePresent
        INTEGER,
fddimibMACMAUnitdataEnable
        INTEGER
    }

```

fddimibMACSMTIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The value of the SMT index associated with this
MAC."

::= { fddimibMACEntry 1 }

fddimibMACIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index variable for uniquely identifying the MAC
object instances, which is the same as the
corresponding resource index in SMT."

REFERENCE

"ANSI { fddiMAC 34 }"

::= { fddimibMACEntry 2 }

fddimibMACIfIndex OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The value of the MIB-II ifIndex corresponding to this MAC. If none is applicable, 0 is returned."

REFERENCE

"MIB-II"

::= { fddimibMACEntry 3 }

fddimibMACFrameStatusFunctions OBJECT-TYPE

SYNTAX INTEGER (0..7)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the MAC's optional Frame Status processing functions.

The value is a sum. This value initially takes the value zero, then for each function present, 2 raised to a power is added to the sum. The powers are according to the following table:

function	Power
fs-repeating	0
fs-setting	1
fs-clearing	2 "

REFERENCE

"ANSI { fddiMAC 11 }"

::= { fddimibMACEntry 4 }

fddimibMACTMaxCapability OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the maximum time value of fddiMACTMax that this MAC can support."

REFERENCE

"ANSI { fddiMAC 13 }"

::= { fddimibMACEntry 5 }

fddimibMACTVXCapability OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the maximum time value of fddiMACTvxValue that this MAC can support."

REFERENCE

"ANSI { fddiMAC 14 }"

::= { fddimibMACEntry 6 }

fddimibMACAvailablePaths OBJECT-TYPE

SYNTAX INTEGER (0..7)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the paths available for this MAC (refer to ANSI SMT 9.7.7).

The value is a sum. This value initially takes the value zero, then for each type of PATH that this MAC has available, 2 raised to a power is added to the sum. The powers are according to the following table:

Path	Power
Primary	0
Secondary	1
Local	2 "

REFERENCE

"ANSI { fddiMAC 22 }"
 ::= { fddimibMACEntry 7 }

fddimibMACCurrentPath OBJECT-TYPE

SYNTAX INTEGER {
 isolated(1),
 local(2),
 secondary(3),
 primary(4),
 concatenated(5),
 thru(6)
 }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the Path into which this MAC is currently inserted (refer to ANSI 9.7.7)."

REFERENCE

"ANSI { fddiMAC 23 }"
 ::= { fddimibMACEntry 8 }

fddimibMACUpstreamNbr OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The MAC's upstream neighbor's long individual MAC address. It has an initial value of the SMT-Unknown-MAC Address and is only modified as

specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2)."

REFERENCE

"ANSI { fddiMAC 24 }"

::= { fddimibMACEntry 9 }

fddimibMACDownstreamNbr OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The MAC's downstream neighbor's long individual MAC address. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2)."

REFERENCE

"ANSI { fddiMAC 25 }"

::= { fddimibMACEntry 10 }

fddimibMACOldUpstreamNbr OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The previous value of the MAC's upstream neighbor's long individual MAC address. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2)."

REFERENCE

"ANSI { fddiMAC 26 }"

::= { fddimibMACEntry 11 }

fddimibMACOldDownstreamNbr OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The previous value of the MAC's downstream neighbor's long individual MAC address. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2)."

REFERENCE

"ANSI { fddiMAC 27 }"

```
::= { fddimibMACEntry 12 }
```

fddimibMACDupAddressTest OBJECT-TYPE

```
SYNTAX  INTEGER { none(1), pass(2), fail(3) }
```

```
ACCESS  read-only
```

```
STATUS  mandatory
```

DESCRIPTION

"The Duplicate Address Test flag, Dup_Addr_Test (refer to ANSI 8.2)."

REFERENCE

"ANSI { fddiMAC 29 }"

```
::= { fddimibMACEntry 13 }
```

fddimibMACRequestedPaths OBJECT-TYPE

```
SYNTAX  INTEGER (0..255)
```

```
ACCESS  read-write
```

```
STATUS  mandatory
```

DESCRIPTION

"List of permitted Paths which specifies the Path(s) into which the MAC may be inserted (refer to ansi SMT 9.7)."

The value is a sum which represents the individual paths that are desired. This value initially takes the value zero, then for each type of PATH that this node is, 2 raised to a power is added to the sum. The powers are according to the following table:

Path	Power
local	0
secondary-alternate	1
primary-alternate	2
concatenated-alternate	3
secondary-preferred	4
primary-preferred	5
concatenated-preferred	6
thru	7 "

REFERENCE

"ANSI { fddiMAC 32 }"

```
::= { fddimibMACEntry 14 }
```

fddimibMACDownstreamPORTType OBJECT-TYPE

```
SYNTAX  INTEGER { a(1), b(2), s(3), m(4), none(5) }
```

```
ACCESS  read-only
```

```
STATUS  mandatory
```

DESCRIPTION

"Indicates the PC-Type of the first port that is

downstream of this MAC (the exit port)."

REFERENCE

"ANSI { fddiMAC 33 }"

::= { fddimibMACEntry 15 }

fddimibMACSMTAddress OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The 48-bit individual address of the MAC used for SMT frames."

REFERENCE

"ANSI { fddiMAC 41 }"

::= { fddimibMACEntry 16 }

fddimibMACTReq OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable is the T_Req_value passed to the MAC. Without having detected a duplicate, the time value of this variable shall assume the maximum supported time value which is less than or equal to the time value of fddiPATHMaxT-Req. When a MAC has an address detected as a duplicate, it may use a time value for this variable greater than the time value of fddiPATHTMaxLowerBound. A station shall cause claim when the new T_Req may cause the value of T_Neg to change in the claim process, (i.e., time value new T_Req < T_Neg, or old T_Req = T_Neg)."

REFERENCE

"ANSI { fddiMAC 51 }"

::= { fddimibMACEntry 17 }

fddimibMACTNeg OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS mandatory

DESCRIPTION

"It is reported as a FddiTimeNano number."

REFERENCE

"ANSI { fddiMAC 52 }"

::= { fddimibMACEntry 18 }

fddimibMACTMax OBJECT-TYPE

SYNTAX FddiTimeNano
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "This variable is the T_Max_value passed to the
 MAC. The time value of this variable shall assume
 the minimum supported time value which is greater
 than or equal to the time value of fddiPATHT-
 MaxLowerBound"

REFERENCE
 "ANSI { fddiMAC 53 }"
 ::= { fddimibMACEntry 19 }

fddimibMACTvxValue OBJECT-TYPE

SYNTAX FddiTimeNano
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "This variable is the TVX_value passed to the MAC.
 The time value of this variable shall assume the
 minimum supported time value which is greater than
 or equal to the time value of
 fddiPATHTVXLowerBound."

REFERENCE
 "ANSI { fddiMAC 54 }"
 ::= { fddimibMACEntry 20 }

fddimibMACFrameCts OBJECT-TYPE

SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "A count of the number of frames received by this
 MAC (refer to ANSI MAC 7.5.1)."

REFERENCE
 "ANSI { fddiMAC 71 }"
 ::= { fddimibMACEntry 21 }

fddimibMACCopiedCts OBJECT-TYPE

SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
 "A count that should as closely as possible match
 the number of frames addressed to (A bit set) and
 successfully copied into the station's receive
 buffers (C bit set) by this MAC (refer to ANSI MAC
 7.5). Note that this count does not include MAC

```
frames."
REFERENCE
    "ANSI { fddiMAC 72 }"
 ::= { fddimibMACEntry 22 }

fddimibMACTransmitCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count that should as closely as possible match
         the number of frames transmitted by this MAC
         (refer to ANSI MAC 7.5). Note that this count
         does not include MAC frames."
    REFERENCE
        "ANSI { fddiMAC 73 }"
 ::= { fddimibMACEntry 23 }

fddimibMACErrorCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count of the number of frames that were
         detected in error by this MAC that had not been
         detected in error by another MAC (refer to ANSI
         MAC 7.5.2)."
```

```
REFERENCE
    "ANSI { fddiMAC 81 }"
 ::= { fddimibMACEntry 24 }

fddimibMACLostCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count of the number of instances that this MAC
         detected a format error during frame reception
         such that the frame was stripped (refer to ANSI
         MAC 7.5.3)."
```

```
REFERENCE
    "ANSI { fddiMAC 82 }"
 ::= { fddimibMACEntry 25 }

fddimibMACFrameErrorThreshold OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-write
    STATUS mandatory
```

DESCRIPTION

"A threshold for determining when a MAC Condition report (see ANSI 8.3.1.1) shall be generated. Stations not supporting variable thresholds shall have a value of 0 and a range of (0..0)."

REFERENCE

"ANSI { fddiMAC 95 }"
 ::= { fddimibMACEntry 26 }

fddimibMACFrameErrorRatio OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable is the value of the ratio,

$$\frac{((\text{delta fddiMACLostCts} + \text{delta fddiMACErrorCts}) / (\text{delta fddiMACFrameCts} + \text{delta fddiMACLostCts}))}{* 2^{16}}$$

REFERENCE

"ANSI { fddiMAC 96 }"
 ::= { fddimibMACEntry 27 }

fddimibMACRMTState OBJECT-TYPE

SYNTAX INTEGER {
 rm0(1), -- Isolated
 rm1(2), -- Non_Op
 rm2(3), -- Ring_Op
 rm3(4), -- Detect
 rm4(5), -- Non_Op_Dup
 rm5(6), -- Ring_Op_Dup
 rm6(7), -- Directed
 rm7(8) -- Trace
 }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Indicates the current state of the RMT State Machine (refer to ANSI 10.3.2)."

REFERENCE

"ANSI { fddiMAC 111 }"
 ::= { fddimibMACEntry 28 }

fddimibMACDaFlag OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

```
        "The RMT flag Duplicate Address Flag, DA_Flag
        (refer to ANSI 10.2.1.2)."
```

REFERENCE

```
        "ANSI { fddiMAC 112 }"
 ::= { fddimibMACEntry 29 }
```

fddimibMACUnaDaFlag OBJECT-TYPE

```
SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "A flag, UNDA_Flag (refer to ANSI 8.2.2.1), set
        when the upstream neighbor reports a duplicate
        address condition. Cleared when the condition
        clears."
```

REFERENCE

```
        "ANSI { fddiMAC 113 }"
 ::= { fddimibMACEntry 30 }
```

fddimibMACFrameErrorFlag OBJECT-TYPE

```
SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "Indicates the MAC Frame Error Condition is
        present when set. Cleared when the condition
        clears and on station initialization."
```

REFERENCE

```
        "ANSI { fddiMAC 114 }"
 ::= { fddimibMACEntry 31 }
```

fddimibMACMAUnitdataAvailable OBJECT-TYPE

```
SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "This variable shall take on the value of the
        MAC_Avail flag defined in RMT."
```

REFERENCE

```
        "ANSI { fddiMAC 116 }"
 ::= { fddimibMACEntry 32 }
```

fddimibMACHardwarePresent OBJECT-TYPE

```
SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "This variable indicates the presence of
```

underlying hardware support for this MAC object.
 If the value of this object is false(2), the
 reporting of the objects in this entry may be
 handled in an implementation-specific manner."

REFERENCE

"ANSI { fddiMAC 117 }"
 ::= { fddimibMACEntry 33 }

fddimibMACMAUnitdataEnable OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This variable determines the value of the
 MA_UNITDATA_Enable flag in RMT. The default and
 initial value of this flag is true(1)."

REFERENCE

"ANSI { fddiMAC 118 }"
 ::= { fddimibMACEntry 34 }

-- the Enhanced MAC Counters group
 -- Implementation of this Group is optional, but systems
 -- claiming support must implement all variables in this
 -- group

-- the MAC Counters table

fddimibMACCountersTable OBJECT-TYPE

SYNTAX SEQUENCE OF FddimibMACCountersEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A list of MAC Counters entries. The number of
 entries shall not exceed the value of
 fddimibMACNumber."

::= { fddimibMACCounters 1 }

fddimibMACCountersEntry OBJECT-TYPE

SYNTAX FddimibMACCountersEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A MAC Counters entry containing information
 common to a given MAC."

INDEX { fddimibMACSMTIndex, fddimibMACIndex }

::= { fddimibMACCountersTable 1 }

```

FddimibMACCountersEntry ::=
    SEQUENCE {
        fddimibMACTokenCts
            Counter,
        fddimibMACTvxExpiredCts
            Counter,
        fddimibMACNotCopiedCts
            Counter,
        fddimibMACLateCts
            Counter,
        fddimibMACRingOpCts
            Counter,
        fddimibMACNotCopiedRatio
            INTEGER,
        fddimibMACNotCopiedFlag
            INTEGER,
        fddimibMACNotCopiedThreshold
            INTEGER
    }

fddimibMACTokenCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count that should as closely as possible match
        the number of times the station has received a
        token (total of non-restricted and restricted) on
        this MAC (see ANSI MAC 7.4). This count is
        valuable for determination of network load."
    REFERENCE
        "ANSI { fddiMAC 74 }"
    ::= { fddimibMACCountersEntry 1 }

fddimibMACTvxExpiredCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A count that should as closely as possible match
        the number of times that TVX has expired."
    REFERENCE
        "ANSI { fddiMAC 83 }"
    ::= { fddimibMACCountersEntry 2 }

fddimibMACNotCopiedCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only

```

STATUS mandatory

DESCRIPTION

"A count that should as closely as possible match the number of frames that were addressed to this MAC but were not copied into its receive buffers (see ANSI MAC 7.5). For example, this might occur due to local buffer congestion. Because of implementation considerations, this count may not match the actual number of frames not copied. It is not a requirement that this count be exact. Note that this count does not include MAC frames."

REFERENCE

"ANSI { fddiMAC 84 }"
 ::= { fddimibMACCountersEntry 3 }

fddimibMACLateCts OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A count that should as closely as possible match the number of TRT expirations since this MAC was reset or a token was received (refer to ANSI MAC 7.4.5)."

REFERENCE

"ANSI { fddiMAC 85 }"
 ::= { fddimibMACCountersEntry 4 }

fddimibMACRingOpCts OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The count of the number of times the ring has entered the 'Ring_Operational' state from the 'Ring Not Operational' state. This count is updated when a SM_MA_STATUS.Indication of a change in the Ring_Operational status occurs (refer to ANSI 6.1.4). Because of implementation considerations, this count may be less than the actual RingOp_Ct. It is not a requirement that this count be exact."

REFERENCE

"ANSI { fddiMAC 86 }"
 ::= { fddimibMACCountersEntry 5 }

fddimibMACNotCopiedRatio OBJECT-TYPE

SYNTAX INTEGER (0..65535)

```

ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "This variable is the value of the ratio:

        (delta fddiMACNotCopiedCts /
        (delta fddiMACCopiedCts +
        delta fddiMACNotCopiedCts )) * 2**16 "
REFERENCE
    "ANSI { fddiMAC 105 }"
 ::= { fddimibMACCountersEntry 6 }

```

```

fddimibMACNotCopiedFlag OBJECT-TYPE
SYNTAX  INTEGER { true(1), false(2) }
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
    "Indicates that the Not Copied condition is
    present when read as true(1). Set to false(2)
    when the condition clears and on station
    initialization."
REFERENCE
    "ANSI { fddiMAC 115 }"
 ::= { fddimibMACCountersEntry 7 }

```

```

fddimibMACNotCopiedThreshold OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
    "A threshold for determining when a MAC condition
    report shall be generated. Stations not
    supporting variable thresholds shall have a value
    of 0 and a range of (0..0)."
REFERENCE
    "ANSI { fddiMAC 103 }"
 ::= { fddimibMACCountersEntry 8 }

```

```

-- the PATH group
-- Implementation of the PATH group is mandatory for all
-- systems which implement manageable FDDI subsystems.

```

```

fddimibPATHNumber OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION

```



```

        "The total number of PATHs possible (across all
        SMTs) on this network management application
        entity. The value for this variable must remain
        constant at least from one re-initialization of
        the entity's network management system to the next
        re-initialization."
 ::= { fddimibPATH 1 }

-- the PATH table

fddimibPATHTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FddimibPATHEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of PATH entries. The number of entries
        shall not exceed the value of fddimibPATHNumber."
    ::= { fddimibPATH 2 }

fddimibPATHEntry OBJECT-TYPE
    SYNTAX  FddimibPATHEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A PATH entry containing information common to a
        given PATH."
    INDEX   { fddimibPATHSMTIndex, fddimibPATHIndex }
    ::= { fddimibPATHTable 1 }

FddimibPATHEntry ::=
    SEQUENCE {
        fddimibPATHSMTIndex
            INTEGER,
        fddimibPATHIndex
            INTEGER,
        fddimibPATHTVXLowerBound
            FddiTimeNano,
        fddimibPATHTMaxLowerBound
            FddiTimeNano,
        fddimibPATHMaxTReq
            FddiTimeNano
    }

fddimibPATHSMTIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory

```

DESCRIPTION

"The value of the SMT index associated with this PATH."

::= { fddimibPATHEntry 1 }

fddimibPATHIndex OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Index variable for uniquely identifying the primary, secondary and local PATH object instances. Local PATH object instances are represented with integer values 3 to 255."

REFERENCE

"ANSI { fddiPATH 11 }"

::= { fddimibPATHEntry 2 }

fddimibPATHTVXLowerBound OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Specifies the minimum time value of fddiMACTvxValue that shall be used by any MAC that is configured in this path. The operational value of fddiMACTvxValue is managed by setting this variable. This variable has the time value range of:

0 < fddimibPATHTVXLowerBound < fddimibPATHMaxTReq
Changes to this variable shall either satisfy the time value relationship:

fddimibPATHTVXLowerBound <=
fddimibMACTVXCapability

of each of the MACs currently on the path, or be considered out of range. The initial value of fddimibPATHTVXLowerBound shall be 2500 nsec (2.5 ms)."

REFERENCE

"ANSI { fddiPATH 21 }"

::= { fddimibPATHEntry 3 }

fddimibPATHTMaxLowerBound OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Specifies the minimum time value of fddiMACTMax that shall be used by any MAC that is configured in this path. The operational value of fddiMACTMax is managed by setting this variable. This variable has the time value range of:

fddimibPATHMaxTReq <= fddimibPATHTMaxLowerBound

and an absolute time value range of:

10000nsec (10 msec) <= fddimibPATHTMaxLowerBound

Changes to this variable shall either satisfy the time value relationship:

fddimibPATHTMaxLowerBound <
fddimibMACTMaxCapability

of each of the MACs currently on the path, or be considered out of range. The initial value of fddimibPATHTMaxLowerBound shall be 165000 nsec (165 msec)."

REFERENCE

"ANSI { fddiPATH 22 }"

::= { fddimibPATHEntry 4 }

fddimibPATHMaxTReq OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Specifies the maximum time value of fddiMACT-Req that shall be used by any MAC that is configured in this path. The operational value of fddiMACT-Req is managed by setting this variable. This variable has the time value range of:

fddimibPATHTVXLowerBound < fddimibPATHMaxTReq <=
fddimibPATHTMaxLowerBound.

The default value of fddimibPATHMaxTReq is 165000 nsec (165 msec)."

REFERENCE

"ANSI { fddiPATH 23 }"

::= { fddimibPATHEntry 5 }

```
-- the PATH Configuration table

fddimibPATHConfigTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FddimibPATHConfigEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table of Path configuration entries.  This
        table lists all the resources that may be in this
        Path."
    REFERENCE
        "ANSI { fddiPATH 18 }"
    ::= { fddimibPATH 3 }

fddimibPATHConfigEntry OBJECT-TYPE
    SYNTAX FddimibPATHConfigEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A collection of objects containing information
        for a given PATH Configuration entry."
    INDEX { fddimibPATHConfigSMTIndex,
            fddimibPATHConfigPATHIndex,
            fddimibPATHConfigTokenOrder }
    ::= { fddimibPATHConfigTable 1 }

FddimibPATHConfigEntry ::=
    SEQUENCE {
        fddimibPATHConfigSMTIndex
            INTEGER,
        fddimibPATHConfigPATHIndex
            INTEGER,
        fddimibPATHConfigTokenOrder
            INTEGER,
        fddimibPATHConfigResourceType
            INTEGER,
        fddimibPATHConfigResourceIndex
            INTEGER,
        fddimibPATHConfigCurrentPath
            INTEGER
    }

fddimibPATHConfigSMTIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of the SMT index associated with this
```

```
        configuration entry."
 ::= { fddimibPATHConfigEntry 1 }

fddimibPATHConfigPATHIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of the PATH resource index associated
         with this configuration entry."
 ::= { fddimibPATHConfigEntry 2 }

fddimibPATHConfigTokenOrder OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "An object associated with Token order for this
         entry. Thus if the token passes resources a, b, c
         and d, in that order, then the value of this
         object for these resources would be 1, 2, 3 and 4
         respectively."
 ::= { fddimibPATHConfigEntry 3 }

fddimibPATHConfigResourceType OBJECT-TYPE
    SYNTAX INTEGER { mac(2), port(4) }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The type of resource associated with this
         configuration entry."
 ::= { fddimibPATHConfigEntry 4 }

fddimibPATHConfigResourceIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of the SMT resource index used to refer
         to the instance of this MAC or Port resource."
 ::= { fddimibPATHConfigEntry 5 }

fddimibPATHConfigCurrentPath OBJECT-TYPE
    SYNTAX INTEGER {
        isolated(1), local(2), secondary(3), primary(4),
        concatenated(5), thru(6)
    }
    ACCESS read-only
```

```

STATUS mandatory
DESCRIPTION
    "The current insertion status for this resource on
    this Path."
 ::= { fddimibPATHConfigEntry 6 }

-- the PORT group
-- Implementation of the PORT group is mandatory for all
-- systems which implement manageable FDDI subsystems.

fddimibPORTNumber OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of PORT implementations (across
        all SMTs) on this network management application
        entity. The value for this variable must remain
        constant at least from one re-initialization of
        the entity's network management system to the next
        re-initialization."
    ::= { fddimibPORT 1 }

-- the PORT table

fddimibPORTTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FddimibPORTEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of PORT entries. The number of entries
        shall not exceed the value of fddimibPORTNumber."
    ::= { fddimibPORT 2 }

fddimibPORTEntry OBJECT-TYPE
    SYNTAX  FddimibPORTEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A PORT entry containing information common to a
        given PORT."
    INDEX   { fddimibPORTSMTIndex, fddimibPORTIndex }
    ::= { fddimibPORTTable 1 }

FddimibPORTEntry ::=
    SEQUENCE {

```

```
fddimibPORTSMTIndex
    INTEGER,
fddimibPORTIndex
    INTEGER,
fddimibPORTMyType
    INTEGER,
fddimibPORTNeighborType
    INTEGER,
fddimibPORTConnectionPolicies
    INTEGER,
fddimibPORTMACIndicated
    INTEGER,
fddimibPORTCurrentPath
    INTEGER,
fddimibPORTRequestedPaths
    OCTET STRING,
fddimibPORTMACPlacement
    FddiResourceId,
fddimibPORTAvailablePaths
    INTEGER,
fddimibPORTPMDCClass
    INTEGER,
fddimibPORTConnectionCapabilities
    INTEGER,
fddimibPORTBSFlag
    INTEGER,
fddimibPORTLCTFailCts
    Counter,
fddimibPORTLerEstimate
    INTEGER,
fddimibPORTLemRejectCts
    Counter,
fddimibPORTLemCts
    Counter,
fddimibPORTLerCutoff
    INTEGER,
fddimibPORTLerAlarm
    INTEGER,
fddimibPORTConnectState
    INTEGER,
fddimibPORTPCMState
    INTEGER,
fddimibPORTPCWithhold
    INTEGER,
fddimibPORTLerFlag
    INTEGER,
fddimibPORTHardwarePresent
    INTEGER,
```

```
        fddimibPORTAction
            INTEGER
    }

fddimibPORTSMTIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of the SMT index associated with this
        PORT."
    ::= { fddimibPORTEntry 1 }

fddimibPORTIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "A unique value for each PORT within a given SMT,
        which is the same as the corresponding resource
        index in SMT. The value for each PORT must remain
        constant at least from one re-initialization of
        the entity's network management system to the next
        re-initialization."
    REFERENCE
        "ANSI { fddiPORT 29 }"
    ::= { fddimibPORTEntry 2 }

fddimibPORTMyType OBJECT-TYPE
    SYNTAX  INTEGER { a(1), b(2), s(3), m(4), none(5) }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of the PORT's PC_Type (refer to ANSI
        9.4.1, and 9.6.3.2)."
    REFERENCE
        "ANSI { fddiPORT 12 }"
    ::= { fddimibPORTEntry 3 }

fddimibPORTNeighborType OBJECT-TYPE
    SYNTAX  INTEGER { a(1), b(2), s(3), m(4), none(5) }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The type of the remote PORT as determined in PCM.
        This variable has an initial value of none, and is
        only modified in PC_RCode(3)_Actions (refer to
        ANSI SMT 9.6.3.2)."
```


REFERENCE

"ANSI { fddiPORT 13 }"
 ::= { fddimibPORTEntry 4 }

fddimibPORTConnectionPolicies OBJECT-TYPE

SYNTAX INTEGER (0..3)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"A value representing the PORT's connection policies desired in the node. The value of pc-mac-lct is a term used in the PC_MAC_LCT Flag (see 9.4.3.2). The value of pc-mac-loop is a term used in the PC_MAC_Loop Flag.

The value is a sum. This value initially takes the value zero, then for each PORT policy, 2 raised to a power is added to the sum. The powers are according to the following table:

Policy	Power
pc-mac-lct	0
pc-mac-loop	1 "

REFERENCE

"ANSI { fddiPORT 14 }"
 ::= { fddimibPORTEntry 5 }

fddimibPORTMACIndicated OBJECT-TYPE

SYNTAX INTEGER {
 tVal9FalseRVal9False(1),
 tVal9FalseRVal9True(2),
 tVal9TrueRVal9False(3),
 tVal9TrueRVal9True(4)
 }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The indications (T_Val(9), R_Val(9)) in PC-Signalling, of the intent to place a MAC in the output token path to a PORT (refer to ANSI SMT 9.6.3.2.)."

REFERENCE

"ANSI { fddiPORT 15 }"
 ::= { fddimibPORTEntry 6 }

fddimibPORTCurrentPath OBJECT-TYPE

SYNTAX INTEGER {
 ce0(1), -- isolated

```

        ce1(2), -- local
        ce2(3), -- secondary
        ce3(4), -- primary
        ce4(5), -- concatenated
        ce5(6)  -- thru
    }
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Indicates the Path(s) into which this PORT is
    currently inserted."
REFERENCE
    "ANSI { fddiPORT 16 }"
 ::= { fddimibPORTEntry 7 }

fddimibPORTRequestedPaths OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (3))
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "This variable is a list of permitted Paths where
    each list element defines the Port's permitted
    Paths. The first octet corresponds to 'none', the
    second octet to 'tree', and the third octet to
    'peer'."
REFERENCE
    "ANSI { fddiPORT 17 }"
 ::= { fddimibPORTEntry 8 }

fddimibPORTMACPlacement OBJECT-TYPE
SYNTAX FddiResourceId -- INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Indicates the MAC, if any, whose transmit path
    exits the station via this PORT. The value shall
    be zero if there is no MAC associated with the
    PORT. Otherwise, the MACIndex of the MAC will be
    the value of the variable."
REFERENCE
    "ANSI { fddiPORT 18 }"
 ::= { fddimibPORTEntry 9 }

fddimibPORTAvailablePaths OBJECT-TYPE
SYNTAX INTEGER (0..7)
ACCESS read-only
STATUS mandatory
DESCRIPTION

```

"Indicates the Paths which are available to this Port. In the absence of faults, the A and B Ports will always have both the Primary and Secondary Paths available.

The value is a sum. This value initially takes the value zero, then for each type of PATH that this port has available, 2 raised to a power is added to the sum. The powers are according to the following table:

Path	Power
Primary	0
Secondary	1
Local	2 "

REFERENCE

"ANSI { fddiPORT 19 }"
 ::= { fddimibPORTEntry 10 }

fddimibPORTPMDClass OBJECT-TYPE

SYNTAX INTEGER {
 multimode(1),
 single-model(2),
 single-mode2(3),
 sonet(4),
 low-cost-fiber(5),
 twisted-pair(6),
 unknown(7),
 unspecified(8)
 }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable indicates the type of PMD entity associated with this port."

REFERENCE

"ANSI { fddiPORT 22 }"
 ::= { fddimibPORTEntry 11 }

fddimibPORTConnectionCapabilities OBJECT-TYPE

SYNTAX INTEGER (0..3)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A value that indicates the connection capabilities of the port. The pc-mac-lct bit indicates that the station has the capability of setting the PC_MAC_LCT Flag. The pc-mac-loop bit

indicates that the station has the capability of setting the PC_MAC_Loop Flag (refer to ANSI 9.4.3.2).

The value is a sum. This value initially takes the value zero, then for each capability that this port has, 2 raised to a power is added to the sum. The powers are according to the following table:

capability	Power
pc-mac-lct	0
pc-mac-loop	1 "

REFERENCE

"ANSI { fddiPORT 23 }"
 ::= { fddimibPORTEntEntry 12 }

fddimibPORTBSFlag OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This variable assumes the value of the BS_Flag (refer to ANSI SMT 9.4.3.3)."

REFERENCE

"ANSI { fddiPORT 33 }"
 ::= { fddimibPORTEntEntry 13 }

fddimibPORTLCTFailCts OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The count of the consecutive times the link confidence test (LCT) has failed during connection management (refer to ANSI 9.4.1)."

REFERENCE

"ANSI { fddiPORT 42 }"
 ::= { fddimibPORTEntEntry 14 }

fddimibPORTLerEstimate OBJECT-TYPE

SYNTAX INTEGER (4..15)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A long term average link error rate. It ranges from 10**-4 to 10**-15 and is reported as the absolute value of the base 10 logarithm (refer to ANSI SMT 9.4.7.5.)."

REFERENCE

"ANSI { fddiPORT 51 }"
 ::= { fddimibPORTEntEntry 15 }

fddimibPORTLemRejectCts OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A link error monitoring count of the times that a link has been rejected."

REFERENCE

"ANSI { fddiPORT 52 }"
 ::= { fddimibPORTEntEntry 16 }

fddimibPORTLemCts OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The aggregate link error monitor error count, set to zero only on station initialization."

REFERENCE

"ANSI { fddiPORT 53 }"
 ::= { fddimibPORTEntEntry 17 }

fddimibPORTLerCutoff OBJECT-TYPE

SYNTAX INTEGER (4..15)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The link error rate estimate at which a link connection will be broken. It ranges from 10**-4 to 10**-15 and is reported as the absolute value of the base 10 logarithm (default of 7)."

REFERENCE

"ANSI { fddiPORT 58 }"
 ::= { fddimibPORTEntEntry 18 }

fddimibPORTLerAlarm OBJECT-TYPE

SYNTAX INTEGER (4..15)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The link error rate estimate at which a link connection will generate an alarm. It ranges from 10**-4 to 10**-15 and is reported as the absolute value of the base 10 logarithm of the estimate"

```
        (default of 8)."  
REFERENCE  
    "ANSI { fddiPORT 59 }"  
 ::= { fddimibPORTEntry 19 }  
  
fddimibPORTConnectState OBJECT-TYPE  
    SYNTAX  INTEGER {  
        disabled(1),  
        connecting(2),  
        standby(3),  
        active(4)  
    }  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "An indication of the connect state of this PORT  
        and is equal to the value of Connect_State (refer  
        to ANSI 9.4.1)"  
REFERENCE  
    "ANSI { fddiPORT 61 }"  
 ::= { fddimibPORTEntry 20 }  
  
fddimibPORTPCMState OBJECT-TYPE  
    SYNTAX  INTEGER {  
        pc0(1), -- Off  
        pc1(2), -- Break  
        pc2(3), -- Trace  
        pc3(4), -- Connect  
        pc4(5), -- Next  
        pc5(6), -- Signal  
        pc6(7), -- Join  
        pc7(8), -- Verify  
        pc8(9), -- Active  
        pc9(10) -- Maint  
    }  
    ACCESS  read-only  
    STATUS  mandatory  
    DESCRIPTION  
        "The state of this Port's PCM state machine refer  
        to ANSI SMT 9.6.2)."  
REFERENCE  
    "ANSI { fddiPORT 62 }"  
 ::= { fddimibPORTEntry 21 }  
  
fddimibPORTPCWithhold OBJECT-TYPE  
    SYNTAX  INTEGER {  
        none(1),  
        m-m(2),
```

```

        otherincompatible(3),
        pathnotavailable(4)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The value of PC_Withhold (refer to ANSI SMT
    9.4.1)."
```

REFERENCE

```

    "ANSI { fddiPORT 63 }"
 ::= { fddimibPORTEntry 22 }
```

fddimibPORTLerFlag OBJECT-TYPE

```

SYNTAX   INTEGER { true(1), false(2) }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The condition becomes active when the value of
    fddiPORTLerEstimate is less than or equal to
    fddiPORTLerAlarm. This will be reported with the
    Status Report Frames (SRF) (refer to ANSI SMT
    7.2.7 and 8.3)."
```

REFERENCE

```

    "ANSI { fddiPORT 64 }"
 ::= { fddimibPORTEntry 23 }
```

fddimibPORTHardwarePresent OBJECT-TYPE

```

SYNTAX   INTEGER { true(1), false(2) }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "This variable indicates the presence of
    underlying hardware support for this Port object.
    If the value of this object is false(2), the
    reporting of the objects in this entry may be
    handled in an implementation-specific manner."
```

REFERENCE

```

    "ANSI { fddiPORT 65 }"
 ::= { fddimibPORTEntry 24 }
```

fddimibPORTAction OBJECT-TYPE

```

SYNTAX   INTEGER {
    other(1),          -- none of the following
    maintPORT(2),
    enablePORT(3),
    disablePORT(4),
    startPORT(5),
    stopPORT(6)
}
```

```
    }  
ACCESS    read-write  
STATUS    mandatory  
DESCRIPTION  
    "Causes a Control signal to be generated with a  
    control_action of 'Signal' and the 'variable'  
    parameter set with the appropriate value (i.e.,  
    PC_Maint, PC_Enable, PC_Disable, PC_Start, or  
    PC_Stop) (refer to ANSI 9.4.2)."  
REFERENCE  
    "ANSI { fddiPORT 70 }"  
 ::= { fddimibPORTEntry 25 }
```

END

5. Acknowledgements

This document was produced by the IETF FDDI MIB working group:

Hossein Alaei, 3Com Corporation
Haggar Alsaleh, Bell Northern Research
William Anderson, Mitre Corporation
Alan Apt, Addison-Wesley
Mary Artibee, Silicon Graphics
Karen Auerbach, Epilogue Technologies
Doug Bagnall, Apollo/Hewlett Packard
Chet Birger, Coral Network Corporation
Pablo Brenner, Fibronics
Howard Brown, Cabletron
Jack Brown, US Army Computer Engineering Center
Eric Brunner
Jeff Case, The University of Tennessee
Tammy Chan, Fibercom
Asheem Chandna, AT&T
Cho Y. Chang, Apollo/Hewlett Packard
Chris Chiotasso, Fibronics
Paul Ciarfella, Digital Equipment Corporation
John Cook, Chipcom
Don Coolidge, Silicon Graphics
Burt Cyr, Unisys
James R. Davin, Massachusetts Institute of Technology
Nabil Damouny
Nadya El-Afandi, Network Systems Corporation
Hunaid Engineer, Cray Research
Jeff Fitzgerald, Fibercom
Richard Fox, Synoptics
Stan Froyd, ACC

Debbie Fatcher, U.S. Naval Surface Warfare Center
Joseph Golio, Cray Research
Jeremy Greene, Coral
Peter Hayden, Digital Equipment Corporation
Scott Hiles, U.S. Naval Surface Warfare Center
Greg Jones, Data General
Satish Joshi, SynOptics Communications
Jayant Kadambi, AT&T Bell Labs
Joanna Karwowska, Data General
Frank Kastenholz, Interlan
Jim Kinder, Fibercom
Christopher Kolb, PSI
Cheryl Krupczak, NCR
Peter Lin, Vitalink
Then Liu
John R. LoVerso, Concurrent Computer Corporation
Ron Mackey, Distributed Systems International, Inc.
Gary Malkin, Proteon
Bruce McClure, Synernetics
Keith McCloghrie, Hughes Lan Systems
Donna McMaster, SynOptics
John O'Hara, Massachusetts Institute of Technology
Luc Pariseau, Digital Equipment Corporation
Dave Perkins, SynOptics Communications
James E. Reeves, SynOptics Communications
Jim Reinstedler, Ungermann-Bass
Radhi Renous, Fibronics
Sal Ricci, AT&T/NCR
Anil Rijsinghani, Digital Equipment Corporation
Bob Rolla, Synernetics
Nelson Ronkin, Synernetics
Marshall T. Rose, Performance Systems International, Inc.
Milt Roselinsky, CMC
Jon Saperia, Digital Equipment Corporation
Greg Satz, cisco Systems
Steven Senum, Network Systems Corporation
Jim Sheridan, IBM Corporation
Jeffrey Schiller, MIT
Dror Shindelman, Fibronics
Mark Sleeper, Sparta
Lou Steinberg, IBM Corporation
Larry Stefani, Digital Equipment Corporation
Mary Jane Strohl, Apollo/Hewlett Packard
Sally Tarquinio, Mitre Corporation
Kaj Tesink, Bellcore
Ian Thomas, Chipcom
Dean Throop, Data General
Bill Townsend, Xylogics

Ahmet H. Tuncay, SynOptics Communications
Mike Turico, Motorola
Chris VandenBerg, ACC
Sudhanshu Verma, Hewlett Packard
Joe Vermeulen, UNISYS
David Waiteman, BBN
Bert Williams, Synernetics
Mark Wood, Distributed Systems International, Inc.
Y. C. Yang
Denis Yaro, Sun Microsystems
Jeff Young, Cray Research

The author gratefully acknowledges the labors of Judi Talley and David Reid of SNMP Research, Inc. for their editorial assistance in the preparation of this document.

6. References

- [1] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [2] Case, J., "FDDI Management Information Base", RFC 1285, SNMP Research, Incorporated, January 1992.
- [3] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.
- [4] McCloghrie K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets", STD 17, RFC 1213, Performance Systems International, March 1991.
- [5] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).
- [6] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization. International Standard 8825, (December, 1987).
- [7] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.

- [8] American National Standards Institute, FDDI Station Management (SMT), Draft Proposed American National Standard, American National Standards Institute, X3T9.5/84-49 REV 7.3.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Jeffrey D. Case
The University of Tennessee
Department of Computer Science
107 Ayres Hall
Knoxville, Tennessee 37996

and

SNMP Research, Incorporated
3001 Kimberlin Heights Road
Knoxville, Tennessee 37920

Phone: (615) 974-5067 or (615) 573-1434
EMail: case@CS.UTK.EDU

Anil Rijasinghani
Digital Equipment Corporation
295 Foster Street
Littleton, MA 01460-1123

Phone: (508) 952-3520
EMail: anil@levers.enet.dec.com

