Fibre Channel Management Framework Integration MIB

Status of this Memo:

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract:

The goal of this document is to fill in missing pieces necessary to enable an enterprise class storage network. One of the more important features of an enterprise class storage network is management; this document gives a framework MIB that will provide an integrated management environment for the enterprise customer.

An enterprise class storage network is comprised of elements (i.e., hubs, switches, converters, gateways, and HBAs) that are developed by many different vendors. The large number of vendors that can exist in a storage network makes mangement a very hard and complicated problem. The main goal of this document’s MIB is to enable interoperability among the various vendors involved in the Fibre Channel marketplace.

Acknowledgements:

The following companies have contributed to the creation of this specification:
- Ancor Communications
- ATTO Technology, Inc.
- Crossroads Systems, Inc.
- EMC Corporation
- Emulex Corporation
The MIB:

FCMGMT-MIB
DEFINITIONS ::= BEGIN
IMPORTS  
    IpAddress, TimeTicks, experimental
    FROM RFC1155-SMI

OBJECT-TYPE 
    FROM RFC-1212
    DisplayString
    FROM RFC1213-MIB

TRAP-TYPE 
    FROM RFC-1215;

FcNameId ::= OCTET STRING (SIZE(8))

fcmgmt OBJECT IDENTIFIER ::= { experimental 94 }

-- groups in fcmgmt

connSet OBJECT IDENTIFIER ::= { fcmgmt 1 }
trapReg OBJECT IDENTIFIER ::= { fcmgmt 2 }

revisionNumber OBJECT-TYPE
SYNTAX DisplayString (SIZE (4))
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "This is the revision number for this MIB. The
    format of the revision value is as follows
    (0) = high order major revision number
    (1) = low order major revision number
    (2) = high order minor revision number
    (3) = low order minor revision number"
The value will be stored as an ASCII value. The following is the current values.

\( \begin{align*}
(0) &= '0' \\
(1) &= '1' \\
(2) &= '5' \\
(3) &= '0'
\end{align*} \)

This defines a revision of 1.5

```
::= { fcmgmt 3 }
```

-- the connectivity unit group

-- Implementation of the group is mandatory for all systems.

\textbf{uNumber OBJECT-TYPE}
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The number of connectivity units present on this system. This is used to identify the number of boards in a chassis or the number of full boxes in a rack."
DEFVAL { 1 }
::= { connSet 1 }

\textbf{systemURL OBJECT-TYPE}
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The top-level URL of the system. If does not exist the value is empty string."
DEFVAL { "" }
::= { connSet 2 }

\textbf{statusChangeTime OBJECT-TYPE}
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The sysuptime timestamp in centiseconds at which the last status change occurred for any members of the set. This represents a union of change information for connUnitStatusChangeTime."
::= { connSet 3 }

\textbf{configurationChangeTime OBJECT-TYPE}
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The sysuptime timestamp in centiseconds at which
the last configuration change occurred for any members of the set. This represents a union of change information for connUnitConfigurationChangeTime.

::= { connSet 4 }

connUnitTableChangeTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The sysuptime timestamp in centiseconds at which the connUnitTable was updated (an entry was either added or deleted)."

::= { connSet 5 }

-- The Connectivity table contains general information on the system's units.
connUnitTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A list of units under a single SNMP agent. The number of entries is given by the value of uNumber. It is 1 for stand-alone system."

::= { connSet 6 }

connUnitEntry OBJECT-TYPE
SYNTAX ConnUnitEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A connectivity unit entry containing objects for a particular unit."
INDEX { connUnitId }

::= { connUnitTable 1 }

ConnUnitEntry ::= SEQUENCE {
  connUnitId
    OCTET STRING,
  connUnitGlobalId
    OCTET STRING,
  connUnitType
    INTEGER,
  connUnitNumports
    INTEGER,
  connUnitState
    INTEGER,
  connUnitStatus
    INTEGER,
  connUnitProduct
    DisplayString

connUnitSn DisplayString,
connUnitUpTime TimeTicks,
connUnitUrl DisplayString,
connUnitDomainId OCTET STRING,
connUnitProxyMaster INTEGER,
connUnitPrincipal INTEGER,
connUnitNumSensors INTEGER,
connUnitStatusChangeTime TimeTicks,
connUnitConfigurationChangeTime TimeTicks,
connUnitNumRevs INTEGER,
connUnitNumZones INTEGER,
connUnitModuleId OCTET STRING,
connUnitName DisplayString,
connUnitInfo DisplayString,
connUnitControl INTEGER,
connUnitContact OCTET STRING,
connUnitLocation OCTET STRING
}

connUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The goal is to create a locally unique Id space for all connUnitIds within this proxy domain. It is left up to the vendor to generate the unique IDs. It is a goal that the value is kept constant between boots. It is also a goal that the relationship be retained between the connUnitId and the connUnitGlobalId if the connUnitGlobalId is non-zero."
::= { connUnitEntry 1 }
connUnitGlobalId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The goal is to create a globally unique Id space for all connUnitGlobalIds. This is a set of hex bytes, not ASCII.
connUnitGlobalId is a WWN, including the leading NAA, in any FC-PH authorized format. This requires that you have to have an IEEE OUI (vendor id) to assign connUnitGlobalId. Once you have supplied the NAA and OUI, you as a vendor are completely free as to how you assign the rest of the bits, as long as you assure they are unique. There is some merit, though not a requirement, that the WWN used as connUnitGlobalId be the same as that of (one of) the permanently attached FC/LAN interface(s). Whether it is or not, it must be unique. As an example, if your agent runs in a host and the host has an HBA, it is quite possible that agent, host, and HBA will all be distinct connUnits, so the host and agent can not use the WWN of the HBA. Another example:
If your hub has a built-in Ethernet port, it might be reasonable for the hub to use its LAN address (prefixed with the with appropriate NAA) as its connUnitGlobalId. But if the Ethernet was a replaceable PCCard, the hub should have an independent ID. If the global ID is not known then this object is set to all zeros. When this value is found by a management station, it will indicate that information about it should not be trusted to be correct. The following are some references for more information.

http://standards.ieee.org/regauth/oui/tutorials/fibrecomp_id.html"
::= { connUnitEntry 2 }

connUnitType OBJECT-TYPE
SYNTAX INTEGER {
  unknown(1),
  other(2), -- none of the following
  hub(3),
  switch(4),
  gateway(5),
  converter(6),
  hba(7),
  proxy-agent(8),
storage-device(9), -- disk, cd, tape, etc
host(10),
storage-subsystem(11), -- raid, library, etc
module(12),
swdriver(13)

}  
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Type of the FC connectivity unit."
 ::= { connUnitEntry 3 }

connUnitNumports OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Number of physical FC ports in the unit  
(internal/embedded, external)."
 ::= { connUnitEntry 4 }

connUnitState OBJECT-TYPE
SYNTAX INTEGER {
  unknown(1),
  online(2),
  offline(3)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Overall state of the connectivity unit."
 ::= { connUnitEntry 5 }

connUnitStatus OBJECT-TYPE
SYNTAX INTEGER {
  unknown(1),
  unused(2),
  ok(3),
  warning(4), -- needs attention
  failed(5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Overall status of the connectivity unit."
 ::= { connUnitEntry 6 }
connUnitProduct OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..79))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The connectivity unit vendor's product model name."
::= { connUnitEntry 7 }

connUnitSn OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..79))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The serial number for this unit. A permanent unique identifier defined by the vendor."
::= { connUnitEntry 8 }

connUnitUpTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The number of centiseconds since the last unit initialization."
::= { connUnitEntry 9 }

connUnitUrl OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"URL to launch management application, if applicable. Otherwise empty string. In a standalone unit, this would be the same as the top-level URL."
::= { connUnitEntry 10 }

connUnitDomainId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(3))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"24 bit ID, right justified, unique within the Fabric address domain. If this value is not applicable, return all bits set to one."
::= { connUnitEntry 11 }
connUnitProxyMaster OBJECT-TYPE
SYNTAX INTEGER {
    no(1),
    yes(2)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is a yes if this is the proxy master unit for a set of managed units. For example, this could be the only unit with a management card in it for a set of units. If this is a standalone unit, then this is set to 'yes'."
 ::= { connUnitEntry 12 }

connUnitPrincipal OBJECT-TYPE
SYNTAX INTEGER {
    unknown(1),
    no(2),
    yes(3)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Whether the unit is the principal unit within the fabric domain. If this value is not applicable, return unknown."
 ::= { connUnitEntry 13 }

connUnitNumSensors OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Number of sensors in the unit."
 ::= { connUnitEntry 14 }

connUnitStatusChangeTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The sysuptime timestamp in centiseconds at which the last status change occurred."
 ::= { connUnitEntry 15 }

connUnitConfigurationChangeTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The sysuptime timestamp in centiseconds at which last configuration change occurred."
 ::= { connUnitEntry 16 }
connUnitNumRevs OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "The number of revisions in the revision table."
DEFVAL { 1 }
::= { connUnitEntry 17 }

connUnitNumZones OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Number of zones defined in the zone table."
::= { connUnitEntry 18 }

connUnitModuleId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(16))
ACCESS read-only
STATUS mandatory
DESCRIPTION "This is a unique id that does not need to be persistent between boots. It is used to group a set of modules together into a single connUnit. The intended use would be to create a connUnit with a type of module and then set its moduleid to its connUnitId. All other connUnits that are part of the module will have the same module id as the container. This should be 16 zeros if it is not used."
::= { connUnitEntry 19 }

connUnitName OBJECT-TYPE
SYNTAX DisplayString (SIZE(40))
ACCESS read-write
STATUS mandatory
DESCRIPTION "A display string containing a name for this unit. This item should be persistent across coldstarts."
::= { connUnitEntry 20 }

connUnitInfo OBJECT-TYPE
SYNTAX DisplayString (SIZE(256))
ACCESS read-write
STATUS mandatory
DESCRIPTION "A display string containing comments such as comments on this unit. This item should be persistent across coldstarts."
::= { connUnitEntry 21 }
connUnitControl OBJECT-TYPE
   SYNTAX INTEGER {
      resetConnUnitColdStart(1),
      resetConnUnitWarmStart(2),
      offlineConnUnit(3),
      onlineConnUnit(4)
   }
   ACCESS read-write -- (or maybe write-only)
   STATUS mandatory
   DESCRIPTION
      "This object is used to control the addressed connUnit. Valid commands are:

      resetConnUnitColdStart: If the addressed connUnit allows this operation to be performed to this MIB object, the addressed unit performs a 'Cold Start' reset.

      If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

      resetConnUnitWarmStart: If the addressed connUnit allows this operation to be performed to this MIB object, the addressed unit performs a 'Warm Start' reset.

      If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

      offlineConnUnit: If the addressed connUnit allows this operation to be performed to this MIB object, the addressed unit puts itself into a vendor-specific 'offline' state. In general, if a unit is in an offline state, it cannot be used to perform meaningful Fibre Channel work.

      If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

      onlineConnUnit: If the addressed connUnit allows this operation to be performed to this MIB object, the addressed unit puts itself into a vendor-specific 'online' state. In general, if a unit is in an online state, it is capable of performing meaningful Fibre Channel work."
If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

For any other value set to this object, the addressed unit sends back a 'bad value' SNMP response to the SNMP set request."

::= { connUnitEntry 22 }

cnnUnitContact OBJECT-TYPE
SYNTAX OCTET STRING
-- note: the size of this
-- string is intentionally
-- left undefined
ACCESS read-write
STATUS mandatory
DESCRIPTION
"If the addressed connUnit allows sets to be done to this object, this MIB object behaves in a similar manner to the MIB2 'sysContact' object. If the size of the octet string is larger than the addressed connUnit can support, the addressed unit sends back a 'tooBig' SNMP response to the SNMP set request.

If the addressed connUnit does not allow writes to this object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request. The addressed unit may also return a null string in response to an SNMP get request for this object."

::= { connUnitEntry 23 }

cnnUnitLocation OBJECT-TYPE
SYNTAX OCTET STRING
-- note: the size of this
-- string is intentionally
-- left undefined
ACCESS read-write
STATUS mandatory
DESCRIPTION
"If the addressed connUnit allows sets to be done to this object, this MIB object behaves in a similar manner to the MIB2 'sysLocation' object. If the size of the octet string is larger than the addressed connUnit can support, the addressed unit sends back a 'tooBig' SNMP response to the SNMP set request.

If the addressed connUnit does not allow writes to this object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request. The addressed unit may also return a null string in response to an SNMP get request for this object."

::= { connUnitEntry 23 }
set request. The addressed unit may also
return a null string in response to an SNMP
get request for this object.

::= { connUnitEntry 24 }

-- The Table of revisions for hardware and software elements.

connUnitRevsTable OBJECT-TYPE
SYNTAX  SEQUENCE OF ConnUnitRevsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Table of the revisions supported by the system."
::= { connSet 7 }

connUnitRevsEntry OBJECT-TYPE
SYNTAX ConnUnitRevsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"
INDEX { connUnitRevsUnitId,
    connUnitRevsIndex }
::= { connUnitRevsTable 1 }

ConnUnitRevsEntry ::= SEQUENCE {
    connUnitRevsUnitId
        OCTET STRING,
    connUnitRevsIndex
        INTEGER,
    connUnitRevsFirmRev
        DisplayString,
    connUnitRevsDescription
        DisplayString
}

connUnitRevsUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object is identical to connUnitId."
::= { connUnitRevsEntry 1 }

connUnitRevsIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A unique value in the range between 1 and
    connUnitNumRevs."
::= { connUnitRevsEntry 2 }
connUnitRevsFirmRev OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The firmware of the unit."
 ::= { connUnitRevsEntry 3 }

connUnitRevsDescription OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Name of a component to which the firmware corresponds."
 ::= { connUnitRevsEntry 4 }

-- The Sensor table

connUnitSensorTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitSensorEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Description of each Sensor for a specific connUnit."
 ::= { connSet 8 }

connUnitSensorEntry OBJECT-TYPE
SYNTAX ConnUnitSensorEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry contains the information for a specific Sensor."
INDEX { connUnitSensorUnitId, connUnitSensorIndex }
 ::= { connUnitSensorTable 1 }

ConnUnitSensorEntry ::= SEQUENCE {
    connUnitSensorUnitId OCTET STRING,
    connUnitSensorIndex INTEGER (1..2147483647),
    connUnitSensorName DisplayString,
    connUnitSensorStatus INTEGER,
    connUnitSensorInfo DisplayString,
    connUnitSensorMessage DisplayString
}
INTERNET-DRAFT           FC-Mgmt-Int MIB          June 1, 1999

connUnitSensorUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object is identical to connUnitId."
 ::= { connUnitSensorEntry 1 }

connUnitSensorIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A unique value for each sensor. Its value
ranges between 1 and the value of
connUnitNumSensors."
 ::= { connUnitSensorEntry 2 }

connUnitSensorName OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This describes a sensor in the unit."
 ::= { connUnitSensorEntry 3 }

connUnitSensorStatus OBJECT-TYPE
SYNTAX INTEGER {
   unknown(1),
   ok(2),
   warning(3),
   failed(4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is the status for the sensor."
 ::= { connUnitSensorEntry 4 }

connUnitSensorInfo OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This describes misc info about the sensor
such as its serial number."
 ::= { connUnitSensorEntry 5 }
connUnitSensorMessage OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This describes the status of the sensor as a message."
::= { connUnitSensorEntry 6 }

-- The Zone table (still under discussion)

connUnitZoneTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitZoneEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Description of each Zone for a specific connUnit."
::= { connSet 9 }

connUnitZoneEntry OBJECT-TYPE
SYNTAX ConnUnitZoneEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry contains the information for a specific Zone."
INDEX { connUnitZoneUnitId, connUnitZoneIndex, connUnitZoneMemberIndex }
::= { connUnitZoneTable 1 }

ConnUnitZoneEntry ::= SEQUENCE {
    connUnitZoneUnitId OCTET STRING,
    connUnitZoneIndex INTEGER (1..2147483647),
    connUnitZoneMemberIndex INTEGER (1..2147483647),
    connUnitZoneNumMember INTEGER,
    connUnitZoneName DisplayString,
    connUnitZoneStatus INTEGER
}
connUnitZoneUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "This object is identical to connUnitId."
 ::= { connUnitZoneEntry 1 }

connUnitZoneIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "A unique value for each zone. Its value
   ranges between 1 and the value of
   connUnitNumZones."
 ::= { connUnitZoneEntry 2 }

connUnitZoneMemberIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "A unique value for each member of a zone.
   Its value ranges between 1 and the value of
   connUnitZoneNumMember."
 ::= { connUnitZoneEntry 3 }

connUnitZoneNumMember OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "The number of members in this zone."
 ::= { connUnitZoneEntry 4 }

connUnitZoneName OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
   "This describes a zone in the unit."
 ::= { connUnitZoneEntry 5 }
connUnitZoneStatus OBJECT-TYPE
SYNTAX INTEGER {
    unknown(1),
    ok(2),
    warning(3),
    failed(4)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is the status for the zone."
::= { connUnitZoneEntry 6 }

-- The port table

connUnitPortTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitPortEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Generic information on ports for a specific connUnit."
::= { connSet 10 }

ConnUnitPortEntry OBJECT-TYPE
SYNTAX ConnUnitPortEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry contains the information for a specific port. The entry corresponds to the connectivity unit index by connUnitId."
INDEX { connUnitPortUnitId, connUnitPortIndex }
::= { connUnitPortTable 1 }

ConnUnitPortEntry ::= SEQUENCE {
    connUnitPortUnitId OCTET STRING,
    connUnitPortIndex INTEGER (1..2147483647),
    connUnitPortType INTEGER,
    connUnitPortClassCap OCTET STRING,
    connUnitPortClassOp OCTET STRING,
    connUnitPortState INTEGER,
    connUnitPortStatus INTEGER,
connUnitPortConnector
INTEGER,
connUnitPortModuleType
INTEGER,
connUnitPortWwn
FcNameId,
connUnitPortConnId
OCTET STRING,
connUnitPortTxFrames
OCTET STRING,
connUnitPortRxFrames
OCTET STRING,
connUnitPortSn
DisplayString,
connUnitPortRevision
DisplayString,
connUnitPortVendor
DisplayString,
connUnitPortSpeed
INTEGER,
connUnitPortControl
INTEGER,
connUnitPortName
OCTET STRING
}

connUnitPortUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object is identical to connUnitId."
::= { connUnitPortEntry 1 }

connUnitPortIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A unique value for each port. Its value
ranges between 1 and the value of
connUnitNumports."
::= { connUnitPortEntry 2 }
connUnitPortType OBJECT-TYPE
SYNTAX INTEGER {
    unknown(1),
    other(2),
    hub-port(3),
    n-port(4), -- end port for fabric
    l-port(5), -- end port for loop
    fl-port(6), -- public loop
    f-port(7), -- fabric port
    e-port(8), -- fabric expansion port
    domain-ctl(9), -- domain controller
    hub-controller(10),
    scsi(11), -- parallel SCSI port
    escon(12),
    lan(13),
    wan(14)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Identification of the port type."
 ::= { connUnitPortEntry 3 }

connUnitPortClassCap OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (2))
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Bit mask that specifies the classes of service capabilities. If this is not applicable, return all bits set to zero. The bits are set right justified and have the following definition:
    unknown(0)
    class-f(1)
    class-one(2)
    class-two(4)
    class-three(8)
    class-four(16)
    class-five(32)
    class-six(64)"
 ::= { connUnitPortEntry 4 }

connUnitPortClassOp OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (2))
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Bit mask that specifies the classes of service that are currently operational. If this is not applicable, return all bits
set to zero. The bits are set right-justified and have the following definition:
unknown(0)
class-f(1)
class-one(2)
class-two(4)
class-three(8)
class-four(16)
class-five(32)
class-six(64)"
::= { connUnitPortEntry 5 }

connUnitPortState OBJECT-TYPE
SYNTAX INTEGER {
   unknown(1),
online(2),
offline(3),
bypassed(4),
linkdown(5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This represents the state of the port protocol."
::= { connUnitPortEntry 6 }

connUnitPortStatus OBJECT-TYPE
SYNTAX INTEGER {
   unknown(1),
unused(2),
ok(3),
warning(4), -- needs attention
failure(5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"An overall hardware status for the port."
::= { connUnitPortEntry 7 }

connUnitPortConnector OBJECT-TYPE
SYNTAX INTEGER {
   unknown(1),
other(2),
unused(3),
shortwave(4),
longwave(5),
copper(6),
scsi(7)
}
connUnitPortModuleType OBJECT-TYPE
  SYNTAX  INTEGER {
    unknown(1),
    other(2),
    gbic(3),
    embedded(4), -- fixed, ie, oneXnine
    glm(5)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The module type of the port connector."
  ::= { connUnitPortEntry 9 }

connUnitPortWwn OBJECT-TYPE
  SYNTAX  FcNameId
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The World Wide Name of the port
    if applicable, otherwise empty string."
 REFERENCE
    "See FC elements MIB for FcNameId definition."
  ::= { connUnitPortEntry 10 }

connUnitPortConnId OBJECT-TYPE
  SYNTAX OCTET STRING (SIZE(3))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This is the Fibre Channel ID of this port.
    It only has a meaning if the connectivity
    unit is a switch. This is expected to be
    a Big Endian value of 24 bits. If not
    supported, return all bits set to 1."
  ::= { connUnitPortEntry 11 }

connUnitPortTxFrames OBJECT-TYPE
  SYNTAX OCTET STRING (SIZE (8))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "A count of the frames that have been
    sent from this port. This is expected to
    be a Big Endian value. If not applicable,
    return all bits set to zero."
  ::= { connUnitPortEntry 12 }
connUnitPortRxFrames OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (8))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A count of the frames that have been received by this port. This is expected to be a Big Endian value. If not applicable, return all bits set to zero."
 ::= { connUnitPortEntry 13 }

connUnitPortSn OBJECT-TYPE
SYNTAX DisplayString (SIZE(79))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The serial number of the unit (e.g., for a GBIC). If this is not applicable, return empty string."
 ::= { connUnitPortEntry 14 }

connUnitPortRevision OBJECT-TYPE
SYNTAX DisplayString (SIZE(79))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The port revision (e.g., for a GBIC)."
 ::= { connUnitPortEntry 15 }

connUnitPortVendor OBJECT-TYPE
SYNTAX DisplayString (SIZE(79))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The port vendor (e.g., for a GBIC)."
 ::= { connUnitPortEntry 16 }

connUnitPortSpeed OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The speed of the port in megabytes per second."
 ::= { connUnitPortEntry 17 }

connUnitPortControl OBJECT-TYPE
SYNTAX INTEGER { 
   resetConnUnitPort(1),
   bypassConnUnitPort(2),
   unbypassConnUnitPort(3),
   offlineConnUnitPort(4),
   onlineConnUnitPort(5) 
}
ACCESS read-write -- (or maybe write-only)
STATUS mandatory
DESCRIPTION
"This object is used to control the addressed connUnit’s port. Valid commands are:

resetConnUnitPort: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific 'reset' operation. Examples of these operations are: the Link Reset protocol, the Loop Initialization protocol, or a resynchronization occurring between the transceiver in the addressed port to the transceiver that the port is connected to.

If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

bypassConnUnitPort: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific 'bypass' operation. Examples of these operations are: the Online to Offline protocol, a request(NON-PARTICIPATING) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.

If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request.

unbypassConnUnitPort: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific 'unbypass' operation. Examples of these operations are: the Link Failure protocol, a request(PARTICIPATING) command to the Loop Port state machine, or addition of the port to an arbitrated loop by a hub.

If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a 'generic error' SNMP response to the SNMP set request."
offlineConnUnitPort: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific 'offline' operation. Examples of these operations are: disabling a port’s transceiver, the Link Failure protocol, request(NON-PARTICIPATING) command to the Loop Port state machine, or removal of the port from an arbitrated loop by a hub.

If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a ’generic error’ SNMP response to the SNMP set request.

onlineConnUnitPort: If the addressed connUnit allows this operation to be performed to this port, the addressed port performs a vendor-specific 'online' operation. Examples of these operations are: enabling a port’s transceiver, the Link Failure protocol, request(PARTICIPATING) command to the Loop Port state machine, or addition of the port from an arbitrated loop by a hub.

If the addressed connUnit does not allow this operation to be performed to this MIB object, the addressed unit sends back a ’generic error’ SNMP response to the SNMP set request.

::= { connUnitPortEntry 18 }

connUnitPortName OBJECT-TYPE
SYNTAX OCTET STRING

-- note: the size of this
-- string is intentionally
-- left undefined

ACCESS read-write
STATUS mandatory
DESCRIPTION
"If the addressed connUnit allows sets to be done to this object, this MIB object can be used to store a string describing the addressed port. If the size of the octet string is larger than the addressed connUnit can support, the addressed unit sends back a ’tooBig’ SNMP response to the SNMP set request.

If the addressed connUnit does not allow writes to this object, the addressed unit sends back a ’generic error’ SNMP response to the SNMP set request. The addressed unit may also return a null string in response to an SNMP get request for this object."

::= { connUnitPortEntry 19 }
connUnitEventTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitEventEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"The table of connectivity unit events. Only errors, warnings, and information (not debugging) should be reported in this table."
 ::= { connSet 11 }

ConnUnitEventEntry OBJECT-TYPE
SYNTAX ConnUnitEventEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"Each entry contains information on a specific event for the given connectivity unit."
INDEX { connUnitEventUnitId, connUnitEventIndex }
 ::= { connUnitEventTable 1 }

ConnUnitEventEntry ::= SEQUENCE {
  connUnitEventUnitId
    OCTET STRING,
  connUnitEventIndex
    INTEGER (1..2147483647),
  connUnitEventId
    INTEGER,
  connUnitREventTime
    DisplayString,
  connUnitSEventTime
    TimeTicks,
  connUnitEventSeverity
    INTEGER,
  connUnitEventType
    INTEGER,
  connUnitEventObjectIndex
    INTEGER,
  connUnitEventDescr
    DisplayString,
  connUnitEventFilter
    INTEGER
}

connUnitEventUnitId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (16))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This object is identical to connUnitId."
 ::= { connUnitEventEntry 1 }
connUnitEventIndex OBJECT-TYPE
SYNTAX INTEGER (1..2147483647)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The unique event index. It ranges between 1 and the size of the event table for the connectivity unit. It wraps on itself when the end of allocated space is reached."
 ::= { connUnitEventEntry 2 }

connUnitEventId OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The unique event Id. Incremented for each event. Not used as table index to simplify the agent implementation. If there is a gap in the values, then it can be assumed that events have been deleted before read."
 ::= { connUnitEventEntry 3 }

connUnitREventTime OBJECT-TYPE
SYNTAX DisplayString (SIZE (15))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is the real time when the event occurred. It has the following format.
       DDMMYYYYY HHMMSS
DD=day number
MM=month number
YYYY=year number
HH=hour number
MM=minute number
SS=seconds number
If not applicable, return a NULL string."
 ::= { connUnitEventEntry 4 }

connUnitSEventTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is the sysuptime timestamp when the event occurred."
 ::= { connUnitEventEntry 5 }
connUnitEventSeverity OBJECT-TYPE
SYNTAX INTEGER {
unknown(1),
other(2),
informational(3),
warning(4),
critical(5),
failure(6),
init(7)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The event severity."
 ::= { connUnitEventEntry 6 }

connUnitEventType OBJECT-TYPE
SYNTAX INTEGER {
unknown(1),
other(2),
status(3),
configuration(4),
topology(5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The type of event that has occurred."
 ::= { connUnitEventEntry 7 }

connUnitEventObjectIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This is used with the connUnitEventType to identify which object."
 ::= { connUnitEventEntry 8 }

connUnitEventDescr OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The description of the event."
 ::= { connUnitEventEntry 9 }
connUnitEventFilter OBJECT-TYPE
SYNTAX INTEGER {
    unknown(1),
    other(2),
    informational(3),
    warning(4),
    critical(5),
    failure(6),
    init(7)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"This value defines the event severity
that will be generated. It records all
events that are greater than or equal to
the value that is set."
 ::= { connUnitEventEntry 10 }
-- The link table (still under discussion)

connUnitLinkTable OBJECT-TYPE
SYNTAX SEQUENCE OF ConnUnitLinkTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"A list of links from this proxy domain to other
(vendors’) proxy domains."
 ::= { connSet 12 }

connUnitLinkNumber OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The number of links to other (vendors’) proxy domains
found. A 0 value indicates no links are present."
 ::= { connSet 13 }

connUnitLinkTableEntry OBJECT-TYPE
SYNTAX ConnUnitLinkTableEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"An entry describing a particular link to another
(vendor’s) proxy domain."
INDEX { connUnitLinkTableId }
 ::= { connUnitLinkTable 1 }
ConnUnitLinkTableEntry ::=  
  SEQUENCE {
    connUnitLinkTableId
      OCTET STRING,
    connUnitlocalUnitNodeId
      OCTET STRING,
    connUnitlocalUnitPortNumber
      INTEGER,
    connUnitremoteUnitNodeId
      OCTET STRING,
    connUnitremoteUnitPortId
      OCTET STRING
  }

  connUnitLinkTableId OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (16))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "This object is identical to connUnitId."
    ::= { connUnitLinkTableEntry 1 }

  connUnitlocalUnitNodeId OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(64))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "The node identifier (eg, WWN) of the unit
      within our local proxy domain that is
      connected to this particular link. If this
      link does not have a WWN then use serial
      number or some other unique id for the
      unit."
    ::= { connUnitLinkTableEntry 2 }

  connUnitlocalUnitPortNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
      "The port number on the unit specified by
      connUnitlocalUnitNodeId that is connected
      to this particular link. A value of 0 in
      this object indicates that it is not known
      which port on the device specified by
      connUnitlocalUnitNodeId is connected to this
      particular link. A non-zero value specifies
      a valid (local) port number that is connected
      to this particular link."
    ::= { connUnitLinkTableEntry 3 }
connUnitRemoteUnitNodeId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(64))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The node identifier (eg, WWN) of the device within the remote proxy domain that is connected to this particular link. If this link does not have a WWN then use serial number or some other unique id for the unit."
::= { connUnitLinkTableEntry 4 }

connUnitRemoteUnitPortId OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(64))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The port identifier (eg, WWN) of the port within the remote proxy domain that is connected to this particular link. Note that this may be the same value as connUnitRemoteUnitNodeId or it may be 0. Both of these values indicate there is no unique port WWN associated with the remote port. If this link does not have a WWN then use serial number or some other unique id for the unit."
::= { connUnitLinkTableEntry 5 }

-- SNMP trap registration group

trapMaxClients OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The maximum number of SNMP trap recipients supported by the system."
::= { trapReg 1 }

trapClientCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The current number of rows in the trap table."
::= { trapReg 2 }
trapRegTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF TrapRegEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
       "A table containing a row for each IP address/port number that traps will be sent to."
   ::= { trapReg 3 }

trapRegEntry OBJECT-TYPE
   SYNTAX  TrapRegEntry
   ACCESS not-accessible
   STATUS mandatory
   DESCRIPTION
       "Ip/Port pair for a specific client."
   INDEX { trapRegIpAddress, trapRegPort }
   ::= { trapRegTable 1 }

TrapRegEntry ::= SEQUENCE {
   trapRegIpAddress  IpAddress,
   trapRegPort      INTEGER (1..2147483647),
   trapRegFilter    INTEGER,
   trapRegRowState  INTEGER
}

trapRegIpAddress OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "The Ip address of a client registered for traps. If the addressed connUnit does not allow writes to this object, addressed unit sends back a 'generic error' SNMP response to the set request."
   ::= { trapRegEntry 1 }

trapRegPort OBJECT-TYPE
   SYNTAX  INTEGER (1..2147483647)
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
       "This contains the value of the UDP port that is in use. If the addressed connUnit port does not allow writes to this object, addressed unit sends back a 'generic error' SNMP response to the set request."
   ::= { trapRegEntry 2 }
trapRegFilter OBJECT-TYPE
   SYNTAX INTEGER {
      unknown(1),
      other(2),
      informational(3),
      warning(4),
      critical(5),
      failure(6),
      init(7)
   }
   ACCESS read-write
   STATUS mandatory
   DESCRIPTION "This value defines the trap severity that will be generated. It generates all traps that are greater than or equal to the value that is set. If the addressed connUnit does not allow writes to this object, addressed unit sends back a 'generic error' SNMP response to the set request. The default value of this object is warning."
   ::= { trapRegEntry 3}

trapRegRowState OBJECT-TYPE
   SYNTAX INTEGER {
      rowDestroy(1), -- Remove row from table.
      rowInactive(2), -- Row exists, but TRAPs disabled
      rowActive(3) -- Row exists and is enabled for sending traps
   }
   ACCESS read-write
   STATUS mandatory
   DESCRIPTION "Specifies the state of the row.
      rowDestroy
      READ: Can never happen.
      WRITE: Remove this row from the table.
      rowInactive
      READ: Indicates that this row does exists, but that TRAPs are not enabled to be sent to the target.
      WRITE: If the row does not exist, and the agent allows writes to the trap table, then a new row is created. The values of the optional columns will be set to default values. TRAPS are not enabled to be sent to the target. If the row already existed, then TRAPs are disabled from being sent to the target.
      rowActive
      READ: Indicates that this row exists, and that TRAPs are enabled to be sent to the target."
WRITE: If the row does not exist, and the agent allows writes to the trap table, then a new row is created. The values of the optional columns will be set to default values. TRAPs are enabled to be sent to the target. If the row already existed, then TRAPs are enabled to be sent to the target."

::= { trapRegEntry 4}

-- Related traps

connUnitStatusChange TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitStatusChangeTime }
DESCRIPTION
"The overall status of the connectivity unit has changed."
::= 1

connUnitAddedTrap TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitId }
DESCRIPTION
"An event is generated when an entry is added to the connUnitTable."
::= 2

connUnitDeletedTrap TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitId }
DESCRIPTION
"An event is generated when an entry is deleted from the connUnitTable."
::= 3

connUnitEventTrap TRAP-TYPE
ENTERPRISE fcmgmt
VARIABLES { connUnitEventId, connUnitEventType, connUnitEventObjectIndex, connUnitEventDescr }
DESCRIPTION
"An event has been generated by the connectivity unit."
::= 4

END
Author Contact Information:

Steven Blumenau  
EMC Corporation  
171 South Street  
Hopkinton, MA 01748-9103  
USA  
Phone: 1-508-435-1000 x4657  
E-mail: blumenau_steven@isus.emc.com

(This draft expires December 1, 1999.)