Definitions of Managed Objects for
IEEE 802.3 Medium Attachment Units (MAUs)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing IEEE 802.3 Medium Attachment Units (MAUs). This document obsoletes RFC 3636. It amends that specification by moving MAU type OBJECT-IDENTITY definitions and relevant textual conventions into a separate Internet Assigned Number Authority (IANA) maintained MIB module. In addition, management information is added to enable support for Ethernet in the First Mile (EFM) and 10GBASE-CX4 MAUs.
# Table of Contents

1. Introduction .................................................. 3  
2. The Internet-Standard Management Framework .................. 3  
3. Overview ...................................................... 4  
   3.1. Relationship to RFC 3636 ................................. 4  
   3.2. Relationship to Other MIBs ......................... 5  
      3.2.1. Relationship to the Interfaces MIB .............. 5  
      3.2.2. Relationship to the 802.3 Repeater MIB Module ... 6  
   3.3. Management of Internal MAUs ........................... 6  
   3.4. Mapping of IEEE 802.3 Managed Objects ............... 6  
   3.5. Addition of New MAU Types ......................... 9  
      3.5.1. dot3MauType OBJECT-IDENTITIES ................. 9  
      3.5.2. IANAifMauTypeListBits TEXTUAL-CONVENTION .... 9  
      3.5.3. IANAifMauMediaAvailable TEXTUAL-CONVENTION .... 9  
      3.5.4. IANAifMauAutoNegCapBits TEXTUAL-CONVENTION ... 10  
      3.5.5. JackType TEXTUAL-CONVENTION .................... 10  
4. MAU MIB Definitions ........................................ 10  
5. IANA-Maintained MAU TC Definitions .......................... 46  
6. Security Considerations .................................... 62  
7. IANA Considerations ....................................... 63  
8. Acknowledgments ............................................. 63  
9. References ................................................... 64  
   9.1. Normative References .................................. 64  
   9.2. Informative References ............................... 66
1. Introduction

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing IEEE 802.3 [IEEE802.3] Medium Attachment Units (MAUs).

The previous version of this document, RFC 3636 [RFC3636], defined a single MIB module. This document splits the original MIB module into two, putting frequently updated object identities and textual conventions into a separate, IANA-maintained MIB module, in order to decrease the need of updating the basic MAU MIB module.

The first version of the IANA-maintained MIB module also extends the list of managed objects to support Ethernet in the First Mile (EFM) and 10GBASE-CX4 interfaces.

Ethernet technology, as defined by the 802.3 Working Group of the IEEE, continues to evolve, with scalable increases in speed, new types of cabling and interfaces, and new features. This evolution may require changes in the managed objects in order to reflect this new functionality. This document, as with other documents issued by this working group, reflects a certain stage in the evolution of Ethernet technology. In the future, this document might be revised, or new documents might be issued by the Ethernet Interfaces and Hub MIB Working Group, in order to reflect the evolution of Ethernet technology.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].
3. Overview

Instances of these object types represent attributes of an IEEE 802.3 MAU. Several types of MAUs are defined in the IEEE 802.3 CSMA/CD standard [IEEE802.3]. These MAUs may be connected to IEEE 802.3 repeaters or to 802.3 (Ethernet-like) interfaces. For convenience, this document refers to these devices as "repeater MAUs" and "interface MAUs."

The definitions presented here are based on Section 30.5, "Layer Management for 10 Mb/s, 100 Mb/s, 1000 Mb/s, and 10 Gb/s Medium Attachment Units (MAUs)", Section 30.6, "Management for link Auto-Negotiation", and Annex 30A, "GDMO Specifications for 802.3 managed object classes" of IEEE Std. 802.3-2005 [IEEE802.3]. This specification is intended to provide for management of all types of Ethernet/802.3 MAUs.

3.1. Relationship to RFC 3636

The management definitions provided in this document are intended to be a superset of those defined by RFC 3636 [RFC3636].

In order to decrease the need of updating the basic MAU MIB module due to the new MAU type, Media Available state, Auto Negotiation capability and/or Jack type introduction, all relevant object identities and textual conventions have been moved to a separate, IANA-maintained MIB module IANA-MAU-MIB, the first version of which is defined in this document. Thus when a new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type is defined by the IEEE 802.3 working group, only the IANA-maintained module needs to be revised, leaving the basic MAU-MIB module defined in this document unchanged.

In addition, the new definitions are added to the IANA-maintained MIB module, to support Ethernet in the First Mile (EFM) and 10GBASE-CX4 interfaces, defined in IEEE Std 802.3ah-2004 [IEEE802.3ah] and IEEE Std 802.3ak-2004 [IEEE802.3ak] respectively, now part of IEEE Std 802.3-2005 [IEEE802.3].

It should be noted that the changes made in this revision will not be entirely backward-compatible with MIB modules that currently import MAU type object identity descriptors from the MAU-MIB; such modules will need to be revised to import those DESCRIPTORS from the IANA-MAU-MIB. Similarly, any management applications that process the object identity definitions (e.g., to present the DESCRIPTION text to a user) will need to get those definitions from the IANA-MAU-MIB instead of the MAU-MIB. While it is true that changes that require such adjustments are not strictly compliant with the SMIv2 rules.
governing MIB module revisions (see [RFC2578] Section 10), in this case continued high maintenance costs that would result from not making these changes make the deviation from the rules justified. It should be noted that the working group was not able to find any examples of MIB modules or management applications that would actually be negatively affected by the changes.

3.2. Relationship to Other MIBs

It is assumed that an agent implementing MAU-MIB will also implement (at least) the ‘system’ group defined in the SNMPv2 MIB [RFC3418]. The following sections identify other MIBs that such an agent should implement.

3.2.1. Relationship to the Interfaces MIB

The sections of this document that define interface MAU-related objects specify an extension to the Interfaces MIB [RFC2863]. An agent implementing these interface-MAU related objects MUST also implement the relevant groups of the ifCompliance3 MODULE-COMPLIANCE statement of the Interface MIB. The value of the object ifMauIfIndex is the same as the value of ‘ifIndex’ used to instantiate the interface to which the given MAU is connected.

It is REQUIRED that an agent implementing the interface-MAU related objects in the MAU-MIB will also fully comply with the dot3Compliance2 MODULE-COMPLIANCE statement of the Ethernet-like Interfaces MIB, [RFC3635]. Furthermore, when the interface-MAU related objects are used to manage a 10GBASE-W PHY -- i.e., when ifMauType is equal to dot3MauType10GigBaseW or any other 10GBASE-W variant -- then the agent MUST also support the Ethernet WAN Interface Sublayer (WIS) MIB [RFC3637] and must follow the interface layering model specified therein. In that case the value of the object ifMauIfIndex is the same as the value of ‘ifIndex’ for the layer at the top of the stack, i.e., for the ifTable entry that has ‘ifType’ equal to ethernetCsmacd(6). If the interface-MAU related objects are used to manage a PHY that allows the MAU type to be changed dynamically, then the agent SHALL create ifTable, ifStackTable, and ifInvStackTable entries that pertain to the WIS when ifMauDefaultType is changed to a 10GBASEW variant (i.e., one of dot3MauType10GigBaseW, dot3MauType10GigBaseEW, dot3MauType10GigBaseLW, or dot3MauType10GigBaseSW) from any other type, and shall destroy the WIS-related entries when ifMauDefaultType is changed to a non-10GBASE-W type. The agent SHALL also change the values of ‘ifConnectorPresent’ and ‘ifHighSpeed’ in the ifTable entry indexed by ifMauIfIndex as specified in [RFC3635] and [RFC3637] when ifMauDefaultType is manipulated in this way, but SHALL NOT otherwise alter that entry.
(Note that repeater ports are not represented as interfaces in the Interface MIB.)

3.2.2. Relationship to the 802.3 Repeater MIB Module

The section of this document that defines repeater MAU-related objects specifies an extension to the 802.3 Repeater MIB defined in [RFC2108]. An agent implementing these repeater-MAU related objects MUST also comply with the snmpRptrModCompl compliance statement of the 802.3 Repeater MIB module.

The values of ‘rpMauGroupIndex’ and ‘rpMauPortIndex’ used to instantiate a repeater MAU variable SHALL be the same as the values of ‘rptrPortGroupIndex’ and ‘rptrPortIndex’ used to instantiate the port that the given MAU is connected to.

3.3. Management of Internal MAUs

In some situations, a MAU can be "internal" -- i.e., its functionality is implemented entirely within a device. For example, a managed repeater may contain an internal repeater-MAU and/or an internal interface-MAU through which management communications originating on one of the repeater’s external ports pass, in order to reach the management agent associated with the repeater. Such internal MAUs may or may not be managed. If they are managed, objects describing their attributes should appear in the appropriate MIB subtree: dot3RpMauBasicGroup for internal repeater-MAUs and dot3IfMauBasicGroup for internal interface-MAUs.

3.4. Mapping of IEEE 802.3 Managed Objects

This section contains the mapping between relevant managed objects (attributes) defined in [IEEE802.3] Clause 30, and managed objects defined in this document.
<table>
<thead>
<tr>
<th>IEEE 802.3 Managed Object</th>
<th>Corresponding SNMP Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>oMAU</td>
<td></td>
</tr>
<tr>
<td>.aMAUID</td>
<td>rpMauIndex or ifMauIndex or broadMauIndex</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aMAUType</td>
<td>rpMauType or ifMauType</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aMAUTypeList</td>
<td>ifMauTypeListBits</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aMediaAvailable</td>
<td>rpMauMediaAvailable or ifMauMediaAvailable</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aLoseMediaCounter</td>
<td>rpMauMediaAvailableStateExits or ifMauMediaAvailableStateExits</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aJabber</td>
<td>rpMauJabberState and rpMauJabberingStateEnters or ifMauJabberState and ifMauJabberingStateEnters</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aMAUAdminState</td>
<td>rpMauStatus or ifMauStatus</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aBbMAUXmitRcvSplitType</td>
<td>broadMauXmitRcvSplitType</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aBroadbandFrequencies</td>
<td>broadMauXmtCarrierFreq and broadMauTranslationFreq</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aFalseCarriers</td>
<td>rpMauFalseCarriers or ifMauFalseCarriers</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.acResetMAU</td>
<td>rpMauStatus or ifMauStatus</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.acMAUAdminControl</td>
<td>rpMauStatus or ifMauStatus</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.nJabber</td>
<td>rpMauJabberTrap or ifMauJabberTrap</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>oAutoNegotiation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aAutoNegID</td>
<td>ifMauIndex</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aAutoNegAdminState</td>
<td>ifMauAutoNegAdminStatus</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.aAutoNegRemoteSignalling</td>
<td>ifMauAutoNegRemoteSignalling</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following IEEE 802.3 managed objects have not been included in the MAU-MIB for the following reasons.

<table>
<thead>
<tr>
<th>IEEE 802.3 Managed Object</th>
<th>Reason for exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>oMAU</td>
<td></td>
</tr>
<tr>
<td>.aIdleErrorCount</td>
<td>Only useful for 100BaseT2, which is not widely implemented.</td>
</tr>
<tr>
<td>oAutoNegotiation</td>
<td></td>
</tr>
<tr>
<td>.aAutoNegLocalSelectorAbility</td>
<td>Only needed for support of isoethernet (802.9a), which is not supported by MAU-MIB.</td>
</tr>
<tr>
<td>.aAutoNegAdvertisedSelectorAbility</td>
<td></td>
</tr>
<tr>
<td>.aAutoNegReceivedSelectorAbility</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Unmapped IEEE 802.3 Managed Objects
3.5. Addition of New MAU Types

3.5.1. dot3MauType OBJECT-IDENTITIES

The dot3MauType OBJECT IDENTIFIER and its OBJECT-IDENTITY definitions has been moved from the MAU-MIB into the IANA-maintained IANA-MAU-MIB, the first version of which is defined in this document.

When a new IEEE 802.3 MAU is defined, IANA can re-issue a version of IANA-MAU-MIB with the new dot3MauType OBJECT-IDENTITY and its matching IANAifMauTypeListBits textual convention value and, possibly, with new IANAifMauMediaAvailable, IANAifMauAutoNegCapBits, and/or IANAifJackType values.

An Expert Review, as defined in RFC 2434 [RFC2434], is REQUIRED for the addition of the new MAU, Media Available states, Auto Negotiation capabilities, and/or Jack types.

In some cases, new MAU types may require additional managed objects or may have side effects on the behavior of existing managed objects. In such cases a standards-track specification (which may be a new document or a revision of this document) is also REQUIRED. Any such document is REQUIRED to note any special properties of the MAU types that it defines – for example, side effects on the ifStackTable as noted in this document for 10GBASE-W MAUs.

3.5.2. IANAifMauTypeListBits TEXTUAL-CONVENTION

The syntax of ifMauTypeListBits is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauTypeListBits, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

3.5.3. IANAifMauMediaAvailable TEXTUAL-CONVENTION

The syntax of ifMauMediaAvailable and rpMauMediaAvailable is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauMediaAvailable, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.
3.5.4.  IANAifMauAutoNegCapBits TEXTUAL-CONVENTION

The syntax of ifMauAutoNegCapabilityBits, ifMauAutoNegCapAdvertisedBits, and ifMauAutoNegCapReceivedBits objects is changed to be a textual convention, such that the enumerated integer values are now defined in the textual convention IANAifMauAutoNegCapBits, which can be re-specified (with additional values, when defined by IEEE 802.3) in the IANA-maintained MIB module without issuing a new version of this document.

3.5.5.  JackType TEXTUAL-CONVENTION

The JackType Textual Convention has been deprecated in favor of the IANAifJackType defined in the IANA-maintained MIB module, so the new Jack types can be added (when defined by IEEE 802.3) without issuing a new version of this document.

4.  MAU MIB Definitions

MAU-MIB DEFINITIONS ::= BEGIN

IMPORTS
   Counter32, Integer32, Counter64,
   OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, mib-2
   FROM SNMPv2-SMI -- RFC 2578
   TruthValue, AutonomousType, TEXTUAL-CONVENTION
   FROM SNMPv2-TC -- RFC 2579
   OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
   FROM SNMPv2-CONF -- RFC 2580
   InterfaceIndex
   FROM IF-MIB -- RFC 2863
   IANAifMauTypeListBits, IANAifMauMediaAvailable,
   IANAifMauAutoNegCapBits, IANAifJackType
   FROM IANA-MAU-MIB
   -- http://www.iana.org/assignments/ianamau-mib

mauMod MODULE-IDENTITY
LAST-UPDATED "200704210000Z" -- April 21, 2007
ORGANIZATION "IETF Ethernet Interfaces and Hub MIB Working Group"
CONTACT-INFO
"WG charter:
   http://www.ietf.org/html.charters/hubmib-charter.html

Mailing Lists:
   General Discussion: hubmib@ietf.org
   To Subscribe: hubmib-request@ietf.org
   In Body: subscribe your_email_address

Beili Standards Track [Page 10]
"Management information for 802.3 MAUs.

The following reference is used throughout this MIB module:

[IEEE802.3] refers to:
technology - Telecommunications and information exchange
between systems - Local and metropolitan area networks -
Specific requirements - Part 3: Carrier sense multiple
access with collision detection (CSMA/CD) access method and
physical layer specifications’.

Of particular interest is Clause 30, ‘Management’.

Copyright (C) The IETF Trust (2007).
This version of this MIB module is part of RFC 4836;
see the RFC itself for full legal notices."

REVISION    "200704210000Z"  -- April 21, 2007
DESCRIPTION "Updated to reference IANA maintained textual
conventions for MAU types, Media Availability state,
Auto Negotiation capabilities, and jack types,
instead of using internally defined values.

This version is published as RFC 4836."

REVISION    "200309190000Z"  -- September 19, 2003
DESCRIPTION "Updated to include support for 10 Gb/s MAUs.
This resulted in the following revisions:
- Added OBJECT-IDENTITY definitions for
10 gigabit MAU types
- Added fiberLC jack type to JackType TC
- Extended ifMauTypeListBits with bits for the 10 gigabit MAU types
- Added enumerations to ifMauMediaAvailable, and updated its DESCRIPTION to reflect behaviour at 10 Gb/s
- Added 64-bit version of ifMauFalseCarriers and added mauIfGrpHCStats object group to contain the new object
- Deprecated mauModIfCompl2 and replaced it with mauModIfCompl3, which includes the new object group

This version published as RFC 3636.

REVISION    "199908240400Z" -- August 24, 1999
DESCRIPTION "This version published as RFC 2668. Updated to include support for 1000 Mb/sec MAUs and flow control negotiation."

REVISION    "199710310000Z" -- October 31, 1997
DESCRIPTION "Version published as RFC 2239."

REVISION    "199309300000Z" -- September 30, 1993
DESCRIPTION "Initial version, published as RFC 1515."

::= { snmpDot3MauMgt 6 }

snmpDot3MauMgt OBJECT IDENTIFIER ::= { mib-2 26 }

-- Textual Conventions

JackType ::= TEXTUAL-CONVENTION

STATUS      deprecated
DESCRIPTION "********* THIS TC IS DEPRECATED **********
This TC has been deprecated in favour of IANAifJackType.

Common enumeration values for repeater and interface MAU jack types."

SYNTAX      INTEGER {
          other(1),
          rj45(2),
          rj45S(3), -- rj45 shielded
          db9(4),
          bnc(5),
          fAUI(6), -- female aui
mAUI(7),  -- male aui
fiberSC(8),
fiberMIC(9),
fiberST(10),
telco(11),
mtrj(12),  -- fiber MT-RJ
hssdc(13),  -- fiber channel style-2
fiberLC(14)

dot3RpMauBasicGroup
   OBJECT IDENTIFIER ::= { snmpDot3MauMgt 1 }
dot3IfMauBasicGroup
   OBJECT IDENTIFIER ::= { snmpDot3MauMgt 2 }
dot3BroadMauBasicGroup
   OBJECT IDENTIFIER ::= { snmpDot3MauMgt 3 }

-- OIDs under the following branch are reserved for
-- the IANA-MAU-MIB to assign as MAU type values:
--    { snmpDot3MauMgt 4 }
dot3IfMauAutoNegGroup
   OBJECT IDENTIFIER ::= { snmpDot3MauMgt 5 }

-- the following OID is the MODULE-IDENTITY value
-- for this MIB module:   { snmpDot3MauMgt 6 }

--
-- The Basic Repeater MAU Table
--
rpMauTable OBJECT-TYPE
SYNTAX      SEQUENCE OF RpMauEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "Table of descriptive and status information
about the MAU(s) attached to the ports of a
repeater."
::= { dot3RpMauBasicGroup 1 }

rpMauEntry OBJECT-TYPE
SYNTAX      RpMauEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "An entry in the table, containing information
about a single MAU."
INDEX       { rpMauGroupIndex,
               rpMauPortIndex,}


```
rfcMauTable

::= { rpMauTable 1 }

rpMauEntry ::= SEQUENCE {
    rpMauGroupIndex                     Integer32,
    rpMauPortIndex                      Integer32,
    rpMauIndex                          Integer32,
    rpMauType                           AutonomousType,
    rpMauStatus                         INTEGER,
    rpMauMediaAvailable                 IANAifMauMediaAvailable,
    rpMauMediaAvailableStateExits       Counter32,
    rpMauJabberState                    INTEGER,
    rpMauJabberingStateEnters           Counter32,
    rpMauFalseCarriers                  Counter32
}

rpMauGroupIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  read-only  -- read-only since originally an
-- SMIv1 index
STATUS      current
DESCRIPTION "This variable uniquely identifies the group
containing the port to which the MAU described
by this entry is connected.

Note: In practice, a group will generally be
a field-replaceable unit (i.e., module, card,
or board) that can fit in the physical system
enclosure, and the group number will correspond
to a number marked on the physical enclosure.

The group denoted by a particular value of this
object is the same as the group denoted by the
same value of rptrGroupIndex."
REFERENCE   "RFC 2108, rptrGroupIndex."
::= { rpMauEntry 1 }

rpMauPortIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  read-only  -- read-only since originally an
-- SMIv1 index
STATUS      current
DESCRIPTION "This variable uniquely identifies the repeater
port within group rpMauGroupIndex to which the
MAU described by this entry is connected."
REFERENCE   "RFC 2108, rptrPortIndex."
```
::= { rpMauEntry 2 }

rpMauIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  read-only -- read-only since originally an
             -- SMIv1 index
STATUS      current
DESCRIPTION "This variable uniquely identifies the MAU
             described by this entry from among other
             MAUs connected to the same port
             (rpMauPortIndex)."
REFERENCE   "[IEEE802.3], 30.5.1.1.1, aMAUID."
::= { rpMauEntry 3 }

rpMauType OBJECT-TYPE
SYNTAX      AutonomousType
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "This object identifies the MAU type. Values for
             standard IEEE 802.3 MAU types are defined in the
             IANA maintained IANA-MAU-MIB module, as
             OBJECT-IDENTITIES of dot3MauType.
             If the MAU type is unknown, the object identifier
             zeroDotZero is returned."
REFERENCE   "[IEEE802.3], 30.5.1.1.2, aMAUType."
::= { rpMauEntry 4 }

rpMauStatus OBJECT-TYPE
SYNTAX      INTEGER { other(1),
             unknown(2),
             operational(3),
             standby(4),
             shutdown(5),
             reset(6) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "The current state of the MAU. This object MAY
             be implemented as a read-only object by those
             agents and MAUs that do not implement software
             control of the MAU state. Some agents may not
             support setting the value of this object to some
             of the enumerated values.
             The value other(1) is returned if the MAU is in
             a state other than one of the states 2 through
             6."

RF
The value unknown(2) is returned when the MAU’s true state is unknown; for example, when it is being initialized.

A MAU in the operational(3) state is fully functional; it operates, and passes signals to its attached DTE or repeater port in accordance to its specification.

A MAU in standby(4) state forces DI and CI to idle, and the media transmitter to idle or fault, if supported. Standby(4) mode only applies to link type MAUs. The state of rpMauMediaAvailable is unaffected.

A MAU in shutdown(5) state assumes the same condition on DI, CI, and the media transmitter, as though it were powered down or not connected. The MAU MAY return other(1) value for the rpMauJabberState and rpMauMediaAvailable objects when it is in this state. For an AUI, this state will remove power from the AUI.

Setting this variable to the value reset(6) resets the MAU in the same manner as a power-off, power-on cycle of at least one-half second would. The agent is not required to return the value reset(6).

Setting this variable to the value operational(3), standby(4), or shutdown(5) causes the MAU to assume the respective state, except that setting a mixing-type MAU or an AUI to standby(4) will cause the MAU to enter the shutdown state."

REFERENCE  "[IEEE802.3], 30.5.1.1.7, aMAUAdminState, 30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1, acResetMAU."

::= { rpMauEntry 5 }

rpMauMediaAvailable OBJECT-TYPE
SYNTAX      IANAifMauMediaAvailable
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "This object identifies Media Available state of the MAU, complementary to the rpMauStatus. Values for the standard IEEE 802.3 Media Available states are defined in the IANA maintained IANA-MAU-MIB"
module, as IANAifMauMediaAvailable TC.

REFERENCE   "[IEEE802.3], 30.5.1.1.4, aMediaAvailable."
::= { rpMauEntry 6 }

rpMauMediaAvailableStateExits OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of the number of times that
rpMauMediaAvailable for this MAU instance leaves
the state available(3).

Discontinuities in the value of this counter can
occur at re-initialization of the management
system and at other times, as indicated by the
value of rpMauMonitorPortLastChange."
REFERENCE   "[IEEE802.3], 30.5.1.1.5, aLoseMediaCounter.
RFC 2108, rpMauMonitorPortLastChange"
::= { rpMauEntry 7 }

rpMauJabberState OBJECT-TYPE
SYNTAX      INTEGER {
         other(1),
         unknown(2),
         noJabber(3),
         jabbering(4)
     }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "The value other(1) is returned if the jabber
state is not 2, 3, or 4. The agent MUST always
return other(1) for MAU type dot3MauTypeAUI.

The value unknown(2) is returned when the MAU’s
ture state is unknown; for example, when it is
being initialized.

If the MAU is not jabbering the agent returns
noJabber(3). This is the ‘normal’ state.

If the MAU is in jabber state the agent returns
the jabbering(4) value."
REFERENCE   "[IEEE802.3], 30.5.1.1.6, aJabber.jabberFlag."
::= { rpMauEntry 8 }

rpMauJabberingStateEnters OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS            current
DESCRIPTION "A count of the number of times that
mauJabberState for this MAU instance enters the
state jabbering(4). For MAUs of type
dot3MauTypeAUI, dot3MauType100BaseT4,
dot3MauType100BaseTX, dot3MauType100BaseFX, and
all 1000Mbps types, this counter will always
indicate zero.

Discontinuities in the value of this counter can
occur at re-initialization of the management
system and at other times, as indicated by the
value of rpMauJabberPortLastChange."
REFERENCE  "[IEEE802.3], 30.5.1.1.6, aJabber.jabberCounter.
RFC 2108, rpMauJabberPortLastChange"
 ::= { rpMauEntry 9 }
rpJackTable OBJECT-TYPE
SYNTAX      SEQUENCE OF RpJackEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "Information about the external jacks attached
to MAUs attached to the ports of a repeater."
::= { dot3RpMauBasicGroup 2 }

rpJackEntry OBJECT-TYPE
SYNTAX      RpJackEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "An entry in the table, containing information
about a particular jack."
INDEX       { rpMauGroupIndex,
             rpMauPortIndex,
             rpMauIndex,
             rpJackIndex
             }
::= { rpJackTable 1 }

RpJackEntry ::= 
SEQUENCE {
  rpJackIndex                         Integer32,
  rpJackType                          IANAifJackType
}

rpJackIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "This variable uniquely identifies the jack
described by this entry from among other jacks
attached to the same MAU (rpMauIndex)."
::= { rpJackEntry 1 }

rpJackType OBJECT-TYPE
SYNTAX      IANAifJackType
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "The jack connector type, as it appears on the
outside of the system."
::= { rpJackEntry 2 }

--
-- The Basic Interface MAU Table
--
ifMauTable OBJECT-TYPE
SYNTAX  SEQUENCE OF IfMauEntry
MAX-ACCESS not-accessible
STATUS  current
DESCRIPTION "Table of descriptive and status information
about MAU(s) attached to an interface."
 ::= { dot3IfMauBasicGroup 1 }

ifMauEntry OBJECT-TYPE
SYNTAX   IfMauEntry
MAX-ACCESS not-accessible
STATUS   current
DESCRIPTION "An entry in the table, containing information
about a single MAU."
INDEX    
  { ifMauIfIndex,  
    ifMauIndex   
  }
 ::= { ifMauTable 1 }

IfMauEntry ::= 
SEQUENCE 
  { ifMauIfIndex   InterfaceIndex,  
    ifMauIndex     Integer32,  
    ifMauType      AutonomousType,  
    ifMauStatus    INTEGER,  
    ifMauMediaAvailable IANAifMauMediaAvailable,  
    ifMauMediaAvailableStateExits Counter32,  
    ifMauJabberState INTEGER,  
    ifMauJabberingStateEnters Counter32,  
    ifMauFalseCarriers Counter32,  
    ifMauTypeList   Integer32,  
    ifMauDefaultType AutonomousType,  
    ifMauAutoNegSupported TruthValue,  
    ifMauTypeListBits IANAifMauTypeListBits,  
    ifMauHCFalseCarriers Counter64  
}

ifMauIfIndex OBJECT-TYPE
SYNTAX   InterfaceIndex
MAX-ACCESS read-only  -- read-only since originally an
                       -- SMIv1 index
STATUS   current
DESCRIPTION "This variable uniquely identifies the interface
to which the MAU described by this entry is connected."
REFERENCE  "RFC 2863, ifIndex"
 ::= { ifMauEntry 1 }
ifMauIndex OBJECT-TYPE
SYNTAX Integer32 (1..2147483647)
MAX-ACCESS read-only -- read-only since originally an
-- SMIv1 index
STATUS current
DESCRIPTION "This variable uniquely identifies the MAU
described by this entry from among other MAUs
connected to the same interface (ifMauIfIndex)."
REFERENCE "[IEEE802.3], 30.5.1.1.1, aMAUID."
::= { ifMauEntry 2 }

ifMauType OBJECT-TYPE
SYNTAX AutonomousType
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object identifies the MAU type. Values for
standard IEEE 802.3 MAU types are defined in the
IANA maintained IANA-MAU-MIB module, as
OBJECT-IDENTITIES of dot3MauType.
If the MAU type is unknown, the object identifier
zeroDotZero is returned.

This object represents the operational type of
the MAU, as determined by either 1) the result
of the auto-negotiation function or 2) if
auto-negotiation is not enabled or is not
implemented for this MAU, by the value of the
object ifMauDefaultType. In case 2), a set to
the object ifMauDefaultType will force the MAU
into the new operating mode."
REFERENCE "[IEEE802.3], 30.5.1.1.2, aMAUType."
::= { ifMauEntry 3 }

ifMauStatus OBJECT-TYPE
SYNTAX INTEGER {
  other(1),
  unknown(2),
  operational(3),
  standby(4),
  shutdown(5),
  reset(6)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "The current state of the MAU. This object MAY
be implemented as a read-only object by those
agents and MAUs that do not implement software
control of the MAU state. Some agents may not
support setting the value of this object to some of the enumerated values.

The value other(1) is returned if the MAU is in a state other than one of the states 2 through 6.

The value unknown(2) is returned when the MAU’s true state is unknown; for example, when it is being initialized.

A MAU in the operational(3) state is fully functional; it operates, and passes signals to its attached DTE or repeater port in accordance to its specification.

A MAU in standby(4) state forces DI and CI to idle and the media transmitter to idle or fault, if supported. Standby(4) mode only applies to link type MAUs. The state of ifMauMediaAvailable is unaffected.

A MAU in shutdown(5) state assumes the same condition on DI, CI, and the media transmitter, as though it were powered down or not connected. The MAU MAY return other(1) value for the ifMauJabberState and ifMauMediaAvailable objects when it is in this state. For an AUI, this state will remove power from the AUI.

Setting this variable to the value reset(6) resets the MAU in the same manner as a power-off, power-on cycle of at least one-half second would. The agent is not required to return the value reset(6).

Setting this variable to the value operational(3), standby(4), or shutdown(5) causes the MAU to assume the respective state, except that setting a mixing-type MAU or an AUI to standby(4) will cause the MAU to enter the shutdown state."

REFERENCE "[IEEE802.3], 30.5.1.1.7, aMAUAdminState, 30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1, acResetMAU."

::= { ifMauEntry 4 }
SYNTAX      IANAifMauMediaAvailable
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "This object identifies Media Available state of the MAU, complementary to the ifMauStatus. Values for the standard IEEE 802.3 Media Available states are defined in the IANA maintained IANA-MAU-MIB module, as IANAIfMauMediaAvailable TC."
REFERENCE   "[IEEE802.3], 30.5.1.1.4, aMediaAvailable."
 ::= { ifMauEntry 5 }

ifMauMediaAvailableStateExits OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of the number of times that ifMauMediaAvailable for this MAU instance leaves the state available(3).

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime."
REFERENCE   "[IEEE802.3], 30.5.1.1.5, aLoseMediaCounter. RFC 2863, ifCounterDiscontinuityTime."
 ::= { ifMauEntry 6 }

ifMauJabberState OBJECT-TYPE
SYNTAX      INTEGER {
            other(1),
            unknown(2),
            noJabber(3),
            jabbering(4)
        }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "The value other(1) is returned if the jabber state is not 2, 3, or 4. The agent MUST always return other(1) for MAU type dot3MauTypeAUI.

The value unknown(2) is returned when the MAU’s true state is unknown; for example, when it is being initialized.

If the MAU is not jabbering the agent returns noJabber(3). This is the 'normal' state.

If the MAU is in jabber state the agent returns
the jabbering(4) value."
REFERENCE "[IEEE802.3], 30.5.1.1.6, aJabber.jabberFlag."
::= { ifMauEntry 7 }

ifMauJabberingStateEnters OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A count of the number of times that mauJabberState for this MAU instance enters the state jabbering(4). This counter will always indicate zero for MAUs of type dot3MauTypeAUI and those of speeds above 10Mbps.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime."
REFERENCE "[IEEE802.3], 30.5.1.1.6, aJabber.jabberCounter. RFC 2863, ifCounterDiscontinuityTime."
::= { ifMauEntry 8 }

ifMauFalseCarriers OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A count of the number of false carrier events during IDLE in 100BASE-X and 1000BASE-X links.

For all other MAU types, this counter will always indicate zero. This counter does not increment at the symbol rate.

It can increment after a valid carrier completion at a maximum rate of once per 100 ms for 100BASE-X and once per 10us for 1000BASE-X until the next CarrierEvent.

This counter can roll over very quickly. A management station is advised to poll the ifMauHCFalseCarriers instead of this counter in order to avoid loss of information.

Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime."
REFERENCE "[IEEE802.3], 30.5.1.1.10, aFalseCarriers."
This object has been deprecated in favour of ifMauTypeListBits.

A value that uniquely identifies the set of possible IEEE 802.3 types that the MAU could be. The value is a sum that initially takes the value zero. Then, for each type capability of this MAU, 2 raised to the power noted below is added to the sum. For example, a MAU that has the capability to be only 10BASE-T would have a value of 512 \((2^{9})\). In contrast, a MAU that supports both 10Base-T (full duplex) and 100BASE-TX (full duplex) would have a value of \(((2^{11}) + (2^{16}))\), or 67584.

The powers of 2 assigned to the capabilities are these:

<table>
<thead>
<tr>
<th>Power</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>other or unknown</td>
</tr>
<tr>
<td>1</td>
<td>AUI</td>
</tr>
<tr>
<td>2</td>
<td>10BASE-5</td>
</tr>
<tr>
<td>3</td>
<td>FOIRL</td>
</tr>
<tr>
<td>4</td>
<td>10BASE-2</td>
</tr>
<tr>
<td>5</td>
<td>10BASE-T duplex mode unknown</td>
</tr>
<tr>
<td>6</td>
<td>10BASE-FP</td>
</tr>
<tr>
<td>7</td>
<td>10BASE-FB</td>
</tr>
<tr>
<td>8</td>
<td>10BASE-FL duplex mode unknown</td>
</tr>
<tr>
<td>9</td>
<td>10BROAD36</td>
</tr>
<tr>
<td>10</td>
<td>10BASE-T half duplex mode</td>
</tr>
<tr>
<td>11</td>
<td>10BASE-T full duplex mode</td>
</tr>
<tr>
<td>12</td>
<td>10BASE-FL half duplex mode</td>
</tr>
<tr>
<td>13</td>
<td>10BASE-FL full duplex mode</td>
</tr>
<tr>
<td>14</td>
<td>100BASE-T4</td>
</tr>
<tr>
<td>15</td>
<td>100BASE-TX half duplex mode</td>
</tr>
<tr>
<td>16</td>
<td>100BASE-TX full duplex mode</td>
</tr>
<tr>
<td>17</td>
<td>100BASE-FX half duplex mode</td>
</tr>
<tr>
<td>18</td>
<td>100BASE-FX full duplex mode</td>
</tr>
<tr>
<td>19</td>
<td>100BASE-T2 half duplex mode</td>
</tr>
</tbody>
</table>
20  100BASE-T2 full duplex mode

If auto-negotiation is present on this MAU, this object will map to ifMauAutoNegCapability.

::= { ifMauEntry 10 }

ifMauDefaultType OBJECT-TYPE
SYNTAX     AutonomousType
MAX-ACCESS read-write
STATUS     current
DESCRIPTION "This object identifies the default administrative baseband MAU type to be used in conjunction with the operational MAU type denoted by ifMauType.

The set of possible values for this object is the same as the set defined for the ifMauType object.

This object represents the administratively-configured type of the MAU. If auto-negotiation is not enabled or is not implemented for this MAU, the value of this object determines the operational type of the MAU. In this case, a set to this object will force the MAU into the specified operating mode.

If auto-negotiation is implemented and enabled for this MAU, the operational type of the MAU is determined by auto-negotiation, and the value of this object denotes the type to which the MAU will automatically revert if/when auto-negotiation is later disabled.

NOTE TO IMPLEMENTORS: It may be necessary to provide for underlying hardware implementations which do not follow the exact behavior specified above. In particular, when ifMauAutoNegAdminStatus transitions from enabled to disabled, the agent implementation MUST ensure that the operational type of the MAU (as reported by ifMauType) correctly transitions to the value specified by this object, rather than continuing to operate at the value earlier determined by the auto-negotiation function."

REFERENCE "[IEEE802.3], 30.5.1.1.1, aMAUID, and 22.2.4.1.4."

::= { ifMauEntry 11 }
ifMauAutoNegSupported OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object indicates whether or not auto-negotiation is supported on this MAU."
::= { ifMauEntry 12 }

ifMauTypeListBits OBJECT-TYPE
SYNTAX IANAifMauTypeListBits
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A value that uniquely identifies the set of possible IEEE 802.3 types that the MAU could be. If auto-negotiation is present on this MAU, this object will map to ifMauAutoNegCapabilityBits. Note that this MAU may be capable of operating as a MAU type that is beyond the scope of this MIB. This is indicated by returning the bit value bOther in addition to any bit values for standard capabilities that are listed in the IANAifMauTypeListBits TC."
::= { ifMauEntry 13 }

ifMauHCFalseCarriers OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A count of the number of false carrier events during IDLE in 100BASE-X and 1000BASE-X links. For all other MAU types, this counter will always indicate zero. This counter does not increment at the symbol rate. This counter is a 64-bit version of ifMauFalseCarriers. Since the 32-bit version of this counter can roll over very quickly, management stations are advised to poll the 64-bit version instead, in order to avoid loss of information. Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times, as indicated by the value of ifCounterDiscontinuityTime."
The ifJackTable applies to MAUs attached to interfaces which have one or more external jacks (connectors).

**ifJackTable**

**SYNTAX**

SEQUENCE OF IfJackEntry

**MAX-ACCESS**

not-accessible

**STATUS**

current

**DESCRIPTION**

"Information about the external jacks attached to MAUs attached to an interface."

::= { dot3IfMauBasicGroup 2 }

**ifJackEntry**

**SYNTAX**

IfJackEntry

**MAX-ACCESS**

not-accessible

**STATUS**

current

**DESCRIPTION**

"An entry in the table, containing information about a particular jack."

**INDEX**

{ ifMauIfIndex, ifMauIndex, ifJackIndex }

::= { ifJackTable 1 }

**IfJackEntry ::=**

SEQUENCE {
    ifJackIndex                         Integer32,
    ifJackType                          IANAifJackType
}

**ifJackIndex**

**SYNTAX**

Integer32 (1..2147483647)

**MAX-ACCESS**

not-accessible

**STATUS**

current

**DESCRIPTION**

"This variable uniquely identifies the jack described by this entry from among other jacks attached to the same MAU."

::= { ifJackEntry 1 }

**ifJackType**

**SYNTAX**

IANAifJackType

**MAX-ACCESS**

read-only

**STATUS**

current

**DESCRIPTION**

"The jack connector type, as it appears on the outside of the system."

::= { ifJackEntry 2 }
-- The MAU Auto-Negotiation Table

ifMauAutoNegTable OBJECT-TYPE
SYNTAX      SEQUENCE OF IfMauAutoNegEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "Configuration and status objects for the
auto-negotiation function of MAUs attached to
interfaces.

The ifMauAutoNegTable applies to systems in
which auto-negotiation is supported on one or
more MAUs attached to interfaces. Note that if
auto-negotiation is present and enabled, the
ifMauType object reflects the result of the
auto-negotiation function."
::= { dot3IfMauAutoNegGroup 1 }

ifMauAutoNegEntry OBJECT-TYPE
SYNTAX      IfMauAutoNegEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "An entry in the table, containing configuration
and status information for the auto-negotiation
function of a particular MAU."
INDEX       { ifMauIfIndex,
                   ifMauIndex
 }
::= { ifMauAutoNegTable 1 }

IfMauAutoNegEntry ::= SEQUENCE {
  ifMauAutoNegAdminStatus           INTEGER,
  ifMauAutoNegRemoteSignaling       INTEGER,
  ifMauAutoNegConfig                INTEGER,
  ifMauAutoNegCapability            Integer32,
  ifMauAutoNegCapAdvertised         Integer32,
  ifMauAutoNegCapReceived           Integer32,
  ifMauAutoNegRestart               INTEGER,
  ifMauAutoNegCapabilityBits        IANAifMauAutoNegCapBits,
  ifMauAutoNegCapAdvertisedBits     IANAifMauAutoNegCapBits,
  ifMauAutoNegCapReceivedBits       IANAifMauAutoNegCapBits,
  ifMauAutoNegRemoteFaultAdvertised INTEGER,
  ifMauAutoNegRemoteFaultReceived   INTEGER
}
ifMauAutoNegAdminStatus OBJECT-TYPE
SYNTAX INTEGER {
    enabled(1),
    disabled(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Setting this object to enabled(1) will cause the interface that has the auto-negotiation signaling ability to be enabled.

If the value of this object is disabled(2) then the interface will act as it would if it had no auto-negotiation signaling. Under these conditions, an IEEE 802.3 MAU will immediately be forced to the state indicated by the value of the object ifMauDefaultType.

NOTE TO IMPLEMENTORS: When ifMauAutoNegAdminStatus transitions from enabled to disabled, the agent implementation MUST ensure that the operational type of the MAU (as reported by ifMauType) correctly transitions to the value specified by the ifMauDefaultType object, rather than continuing to operate at the value earlier determined by the auto-negotiation function."
REFERENCE "[IEEE802.3], 30.6.1.1.2, aAutoNegAdminState, and 30.6.1.2.2, acAutoNegAdminControl."
 ::= { ifMauAutoNegEntry 1 }

ifMauAutoNegRemoteSignaling OBJECT-TYPE
SYNTAX INTEGER {
    detected(1),
    notdetected(2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A value indicating whether the remote end of the link is using auto-negotiation signaling. It takes the value detected(1) if and only if, during the previous link negotiation, FLP Bursts were received."
REFERENCE "[IEEE802.3], 30.6.1.1.3, aAutoNegRemoteSignaling."
 ::= { ifMauAutoNegEntry 2 }

ifMauAutoNegConfig OBJECT-TYPE
SYNTAX      INTEGER {
    other(1),
    configuring(2),
    complete(3),
    disabled(4),
    parallelDetectFail(5)
}
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A value indicating the current status of the
auto-negotiation process. The enumeration
parallelDetectFail(5) maps to a failure in
parallel detection as defined in 28.2.3.1 of
[IEEE802.3]."
REFERENCE   "[IEEE802.3], 30.6.1.1.4, aAutoNegAutoConfig."
::= { ifMauAutoNegEntry 4 }

ifMauAutoNegCapability OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********

This object has been deprecated in favour of
ifMauAutoNegCapabilityBits.

A value that uniquely identifies the set of
capabilities of the local auto-negotiation
entity. The value is a sum that initially
takes the value zero. Then, for each capability
of this interface, 2 raised to the power noted
below is added to the sum. For example, an
interface that has the capability to support
only 100Base-TX half duplex would have a value
of 32768 (2**15). In contrast, an interface
that supports both 100Base-TX half duplex and
100Base-TX full duplex would have a value of
98304 ((2**15) + (2**16)).

The powers of 2 assigned to the capabilities are
these:

<table>
<thead>
<tr>
<th>Power</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>other or unknown</td>
</tr>
<tr>
<td>(1-9)</td>
<td>(reserved)</td>
</tr>
<tr>
<td>10</td>
<td>10BASE-T half duplex mode</td>
</tr>
<tr>
<td>11</td>
<td>10BASE-T full duplex mode</td>
</tr>
<tr>
<td>12</td>
<td>(reserved)</td>
</tr>
</tbody>
</table>
Note that interfaces that support this MIB may have capabilities that extend beyond the scope of this MIB.

REFERENCE

"[IEEE802.3], 30.6.1.1.5, aAutoNegLocalTechnologyAbility."

::= { ifMauAutoNegEntry 5 }

ifMauAutoNegCapAdvertised OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********
This object has been deprecated in favour of ifMauAutoNegCapAdvertisedBits.

A value that uniquely identifies the set of capabilities advertised by the local auto-negotiation entity. Refer to ifMauAutoNegCapability for a description of the possible values of this object.

Capabilities in this object that are not available in ifMauAutoNegCapability cannot be enabled."

REFERENCE

"[IEEE802.3], 30.6.1.1.6, aAutoNegAdvertisedTechnologyAbility."

::= { ifMauAutoNegEntry 6 }

ifMauAutoNegCapReceived OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********
This object has been deprecated in favour of ifMauAutoNegCapReceivedBits.

A value that uniquely identifies the set of
capabilities received from the remote auto-negotiation entity. Refer to
ifMauAutoNegCapability for a description of the possible values of this object.

Note that interfaces that support this MIB may be attached to remote auto-negotiation entities that have capabilities beyond the scope of this MIB.

REFERENCE "[IEEE802.3], 30.6.1.1.7,
aAutoNegReceivedTechnologyAbility."
::= { ifMauAutoNegEntry 7 }

ifMauAutoNegRestart OBJECT-TYPE
SYNTAX INTEGER {
    restart(1),
    norestart(2)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "If the value of this object is set to
restart(1) then this will force auto-negotiation
to begin link renegotiation. If auto-negotiation
signaling is disabled, a write to this object
has no effect.
Setting the value of this object to norestart(2)
has no effect."
REFERENCE "[IEEE802.3], 30.6.1.2.1,
aAutoNegRestartAutoConfig."
::= { ifMauAutoNegEntry 8 }

ifMauAutoNegCapabilityBits OBJECT-TYPE
SYNTAX IANAifMauAutoNegCapBits
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A value that uniquely identifies the set of
capabilities of the local auto-negotiation
text entity. Note that interfaces that support this
MIB may have capabilities that extend beyond the
scope of this MIB.

Note that the local auto-negotiation entity may
support some capabilities beyond the scope of
this MIB. This is indicated by returning the
bit value bOther in addition to any bit values
for standard capabilities that are listed in the
IANAifMauAutoNegCapBits TC."
ifMauAutoNegCapAdvertisedBits OBJECT-TYPE
SYNTAX      IANAifMauAutoNegCapBits
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION "A value that uniquely identifies the set of capabilities advertised by the local auto-negotiation entity. Capabilities in this object that are not available in ifMauAutoNegCapabilityBits cannot be enabled.

Note that the local auto-negotiation entity may advertise some capabilities beyond the scope of this MIB. This is indicated by returning the bit value bOther in addition to any bit values for standard capabilities that are listed in the IANAifMauAutoNegCapBits TC."
REFERENCE   "[IEEE802.3], 30.6.1.1.6, aAutoNegAdvertisedTechnologyAbility."
::= { ifMauAutoNegEntry 10 }

ifMauAutoNegCapReceivedBits OBJECT-TYPE
SYNTAX      IANAifMauAutoNegCapBits
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A value that uniquely identifies the set of capabilities received from the remote auto-negotiation entity. Note that interfaces that support this MIB may be attached to remote auto-negotiation entities that have capabilities beyond the scope of this MIB. This is indicated by returning the bit value bOther in addition to any bit values for standard capabilities that are listed in the IANAifMauAutoNegCapBits TC."
REFERENCE   "[IEEE802.3], 30.6.1.1.7, aAutoNegReceivedTechnologyAbility."
::= { ifMauAutoNegEntry 11 }

ifMauAutoNegRemoteFaultAdvertised OBJECT-TYPE
SYNTAX      INTEGER { noError(1), offline(2),
linkFailure(3),
   autoNegError(4)
}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "A value that identifies any local fault
indications that this MAU has detected and will
advertise at the next auto-negotiation
interaction for 1000Mbps MAUs."
REFERENCE "[IEEE802.3], 30.6.1.1.6,
aAutoNegAdvertisedTechnologyAbility."
::= { ifMauAutoNegEntry 12 }

ifMauAutoNegRemoteFaultReceived OBJECT-TYPE
SYNTAX INTEGER {
   noError(1),
   offline(2),
   linkFailure(3),
   autoNegError(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A value that identifies any fault indications
received from the far end of a link by the
local auto-negotiation entity for 1000Mbps
MAUs."
REFERENCE "[IEEE802.3], 30.6.1.1.7,
aAutoNegReceivedTechnologyAbility."
::= { ifMauAutoNegEntry 13 }

--
-- The Basic Broadband MAU Table
--

broadMauBasicTable OBJECT-TYPE
SYNTAX SEQUENCE OF BroadMauBasicEntry
MAX-ACCESS not-accessible
STATUS deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********

This entire table has been deprecated. There
have been no reported implementations of this
table, and it is unlikely that there ever will
be. IEEE recommends that broadband MAU types
should not be used for new installations.

Table of descriptive and status information
about the broadband MAUs connected to interfaces.
 ::= { dot3BroadMauBasicGroup 1 }

broadMauBasicEntry OBJECT-TYPE
SYNTAX      BroadMauBasicEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********
 An entry in the table, containing information about a single broadband MAU.
 INDEX       { broadMauIfIndex,
                      broadMauIndex }
 ::= { broadMauBasicTable 1 }

BroadMauBasicEntry ::= 
  SEQUENCE {
    broadMauIfIndex                     InterfaceIndex,
    broadMauIndex                       Integer32,
    broadMauXmtRcvSplitType             INTEGER,
    broadMauXmtCarrierFreq              Integer32,
    broadMauTranslationFreq             Integer32
  }

broadMauIfIndex OBJECT-TYPE
SYNTAX      InterfaceIndex
MAX-ACCESS  read-only  -- read-only since originally an
  -- SMIv1 index
STATUS      deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********
 This variable uniquely identifies the interface to which the MAU described
 by this entry is connected."
REFERENCE   "RFC 2863, ifIndex."
 ::= { broadMauBasicEntry 1 }

broadMauIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  read-only  -- read-only since originally an
  -- SMIv1 index
STATUS      deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********
 This variable uniquely identifies the MAU connected to interface broadMauIfIndex that is
broadMauXmtRcvSplitType OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    single(2),
    dual(3)
}
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********

This object indicates the type of frequency multiplexing/cabling system used to separate the transmit and receive paths for the 10BROAD36 MAU.

The value other(1) is returned if the split type is not either single or dual.

The value single(2) indicates a single cable system. The value dual(3) indicates a dual cable system, offset normally zero."
REFERENCE "[IEEE802.3], 30.5.1.1.8, aBroadbandFrequencies.xmitCarrierFrequency." ::= { broadMauBasicEntry 3 }

broadMauTranslationFreq OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION "********** THIS OBJECT IS DEPRECATED **********

This variable indicates the translation offset

REFERENCE "[IEEE802.3], 30.5.1.1.9, aBroadbandFrequencies.xmitCarrierFrequency." ::= { broadMauBasicEntry 4 }
frequency of the 10BROAD36 MAU in MHz/4; that is, in units of 250 kHz.

REFERENCE "[IEEE802.3], 30.5.1.1.9, aBroadbandFrequencies.translationFrequency."

 ::= { broadMauBasicEntry 5 }

-- Notifications for use by 802.3 MAUs

snmpDot3MauTraps OBJECT IDENTIFIER ::= { snmpDot3MauMgt 0 }

rpMauJabberTrap NOTIFICATION-TYPE
  OBJECTS   { rpMauJabberState }
  STATUS    current
  DESCRIPTION "This trap is sent whenever a managed repeater MAU enters the jabber state.

  The agent MUST throttle the generation of consecutive rpMauJabberTraps so that there is at least a five-second gap between them."

REFERENCE "[IEEE802.3], 30.5.1.3.1, nJabber notification."

 ::= { snmpDot3MauTraps 1 }

ifMauJabberTrap NOTIFICATION-TYPE
  OBJECTS   { ifMauJabberState }
  STATUS    current
  DESCRIPTION "This trap is sent whenever a managed interface MAU enters the jabber state.

  The agent MUST throttle the generation of consecutive ifMauJabberTraps so that there is at least a five-second gap between them."

REFERENCE "[IEEE802.3], 30.5.1.3.1, nJabber notification."

 ::= { snmpDot3MauTraps 2 }

-- Conformance information

mauModConf
  OBJECT IDENTIFIER ::= { mauMod 1 }

mauModCompls
  OBJECT IDENTIFIER ::= { mauModConf 1 }

mauModObjGrps
  OBJECT IDENTIFIER ::= { mauModConf 2 }

mauModNotGrps
  OBJECT IDENTIFIER ::= { mauModConf 3 }

-- Object groups
mauRpGrpBasic OBJECT-GROUP
OBJECTS     { rpMauGroupIndex,
               rpMauPortIndex,
               rpMauIndex,
               rpMauType,
               rpMauStatus,
               rpMauMediaAvailable,
               rpMauMediaAvailableStateExits,
               rpMauJabberState,
               rpMauJabberingStateEnters
               }
STATUS      current
DESCRIPTION "Basic conformance group for MAUs attached to
repeater ports. This group is also the
conformance specification for
RFC 1515 implementations."
::= { mauModObjGrps 1 }

mauRpGrp100Mbs OBJECT-GROUP
OBJECTS     { rpMauFalseCarriers }
STATUS      current
DESCRIPTION "Conformance group for MAUs attached to
repeater ports with 100 Mb/s or greater
capability."
::= { mauModObjGrps 2 }

mauRpGrpJack OBJECT-GROUP
OBJECTS     { rpJackType }
STATUS      current
DESCRIPTION "Conformance group for MAUs attached to
repeater ports with managed jacks."
::= { mauModObjGrps 3 }

mauIfGrpBasic OBJECT-GROUP
OBJECTS     { ifMauIfIndex,
               ifMauIndex,
               ifMauType,
               ifMauStatus,
               ifMauMediaAvailable,
               ifMauMediaAvailableStateExits,
               ifMauJabberState,
               ifMauJabberingStateEnters
               }
STATUS      current
DESCRIPTION "Basic conformance group for MAUs attached to
interfaces. This group also provides a
conformance specification for RFC 1515
implementations."
mauIfGrp100Mbs OBJECT-GROUP
OBJECTS { ifMauFalseCarriers,
           ifMauTypeList,
           ifMauDefaultType,
           ifMauAutoNegSupported
        }
STATUS deprecated
DESCRIPTION "********** THIS GROUP IS DEPRECATED **********

Conformance group for MAUs attached to interfaces with 100 Mb/s capability.

This object group has been deprecated in favor of mauIfGrpHighCapacity."

mauIfGrpJack OBJECT-GROUP
OBJECTS { ifJackType }
STATUS current
DESCRIPTION "Conformance group for MAUs attached to interfaces with managed jacks."

mauIfGrpAutoNeg OBJECT-GROUP
OBJECTS { ifMauAutoNegAdminStatus,
           ifMauAutoNegRemoteSignaling,
           ifMauAutoNegConfig,
           ifMauAutoNegCapability,
           ifMauAutoNegCapAdvertised,
           ifMauAutoNegCapReceived,
           ifMauAutoNegRestart
        }
STATUS deprecated
DESCRIPTION "********** THIS GROUP IS DEPRECATED **********

Conformance group for MAUs attached to interfaces with managed auto-negotiation.

This object group has been deprecated in favor of mauIfGrpAutoNeg2."

mauBroadBasic OBJECT-GROUP
OBJECTS { broadMauIfIndex,
           broadMauIndex,
MAU MIB

broadMauXmtRcvSplitType,
broadMauXmtCarrierFreq,
broadMauTranslationFreq

STATUS deprecated
DESCRIPTION "********* THIS GROUP IS DEPRECATED **********
Conformance group for broadband MAUs attached to interfaces.

This object group is deprecated. There have been no reported implementations of this group, and it was felt to be unlikely that there will be any future implementations."

::= { mauModObjGrps 8 }

mauIfGrpHighCapacity OBJECT-GROUP
OBJECTS { ifMauFalseCarriers,
    ifMauTypeListBits,
    ifMauDefaultType,
    ifMauAutoNegSupported
    }
STATUS current
DESCRIPTION "Conformance group for MAUs attached to interfaces with 100 Mb/s or greater capability."
::= { mauModObjGrps 9 }

mauIfGrpAutoNeg2 OBJECT-GROUP
OBJECTS { ifMauAutoNegAdminStatus,
    ifMauAutoNegRemoteSignaling,
    ifMauAutoNegConfig,
    ifMauAutoNegCapabilityBits,
    ifMauAutoNegCapAdvertisedBits,
    ifMauAutoNegCapReceivedBits,
    ifMauAutoNegRestart
    }
STATUS current
DESCRIPTION "Conformance group for MAUs attached to interfaces with managed auto-negotiation."
::= { mauModObjGrps 10 }

mauIfGrpAutoNeg1000Mbps OBJECT-GROUP
OBJECTS { ifMauAutoNegRemoteFaultAdvertised,
    ifMauAutoNegRemoteFaultReceived
    }
STATUS current
DESCRIPTION "Conformance group for 1000Mbps MAUs attached to interfaces with managed auto-negotiation."
::= { mauModObjGrps 11 }
mauIfGrpHCStats OBJECT-GROUP
OBJECTS { ifMauHCFalseCarriers }
STATUS current
DESCRIPTION "Conformance for high capacity statistics for
MAUs attached to interfaces."
 ::= { mauModObjGrps 12 }

-- Notification groups

rpMauNotifications NOTIFICATION-GROUP
NOTIFICATIONS { rpMauJabberTrap }
STATUS current
DESCRIPTION "Notifications for repeater MAUs."
 ::= { mauModNotGrps 1 }

ifMauNotifications NOTIFICATION-GROUP
NOTIFICATIONS { ifMauJabberTrap }
STATUS current
DESCRIPTION "Notifications for interface MAUs."
 ::= { mauModNotGrps 2 }

-- Compliances

mauModRpCompl MODULE-COMPLIANCE
STATUS deprecated
DESCRIPTION "******** THIS COMPLIANCE IS DEPRECATED ********
Compliance for MAUs attached to repeater
ports.

This compliance is deprecated and replaced by
mauModRpCompl2, which corrects an oversight by
allowing rpMauStatus to be implemented
read-only."

MODULE -- this module
MANDATORY-GROUPS { mauRpGrpBasic }

GROUP mauRpGrp100Mbs
DESCRIPTION "Implementation of this optional group is
recommended for MAUs that have 100Mb/s or
greater capability."

GROUP mauRpGrpJack
DESCRIPTION "Implementation of this optional group is
recommended for MAUs that have one or more
external jacks."

GROUP rpMauNotifications
DESCRIPTION "Implementation of this group is recommended for MAUs attached to repeater ports."
::= { mauModCompls 1 }

mauModIfCompl MODULE-COMPLIANCE
STATUS deprecated
DESCRIPTION "******** THIS COMPLIANCE IS DEPRECATED ********

Compliance for MAUs attached to interfaces. This compliance is deprecated and replaced by mauModIfCompl2."

MODULE -- this module
MANDATORY-GROUPS { mauIfGrpBasic }

GROUP mauIfGrp100Mbs
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have 100Mb/s capability."

GROUP mauIfGrpJack
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have one or more external jacks."

GROUP mauIfGrpAutoNeg
DESCRIPTION "Implementation of this group is mandatory for MAUs that support managed auto-negotiation."

GROUP mauBroadBasic
DESCRIPTION "Implementation of this group is mandatory for broadband MAUs."

GROUP ifMauNotifications
DESCRIPTION "Implementation of this group is recommended for MAUs attached to interfaces."
::= { mauModCompls 2 }

mauModIfComp12 MODULE-COMPLIANCE
STATUS deprecated
DESCRIPTION "******** THIS COMPLIANCE IS DEPRECATED ********

Compliance for MAUs attached to interfaces. This compliance is deprecated and replaced by mauModIfComp13."
MODULE -- this module
MANDATORY-GROUPS { mauIfGrpBasic }

GROUP     mauIfGrpHighCapacity
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have 100Mb/s or greater capability."

GROUP     mauIfGrpJack
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have one or more external jacks."

GROUP     mauIfGrpAutoNeg2
DESCRIPTION "Implementation of this group is mandatory for MAUs that support managed auto-negotiation."

GROUP     mauIfGrpAutoNeg1000Mbps
DESCRIPTION "Implementation of this group is mandatory for MAUs that have 1000Mb/s or greater capability and support managed auto-negotiation."

GROUP     ifMauNotifications
DESCRIPTION "Implementation of this group is recommended for MAUs attached to interfaces."

OBJECT     ifMauStatus
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."
::= { mauModCompls 3 }

mauModRpCompl12 MODULE-COMPLIANCE
STATUS     current
DESCRIPTION "Compliance for MAUs attached to repeater ports.

Note that compliance with this compliance statement requires compliance with the
snmpRptrModCompl MODULE-COMPLIANCE statement of
the SNMP-REPEATER-MIB (RFC 2108)."

MODULE -- this module
MANDATORY-GROUPS { mauRpGrpBasic }

GROUP     mauRpGrp100Mbs
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have 100Mb/s or greater capability."

GROUP mauRpGrpJack
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have one or more external jacks."

GROUP rpMauNotifications
DESCRIPTION "Implementation of this group is recommended for MAUs attached to repeater ports."

OBJECT rpMauStatus
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."
::= { mauModCompls 4 }

mauModIfComp13 MODULE-COMPLIANCE
STATUS current
DESCRIPTION "Compliance for MAUs attached to interfaces. Note that compliance with this compliance statement requires compliance with the ifCompliance3 MODULE-COMPLIANCE statement of the IF-MIB (RFC 2863) and the dot3Compliance2 MODULE-COMPLIANCE statement of the EtherLike-MIB (RFC3635)."

MODULE -- this module
MANDATORY-GROUPS { mauIfGrpBasic }

GROUP mauIfGrpHighCapacity
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have 100Mb/s or greater capability."

GROUP mauIfGrpHCStats
DESCRIPTION "Implementation of this group is mandatory for MAUs that have 1000Mb/s capacity, and is recommended for MAUs that have 100Mb/s capacity."

GROUP mauIfGrpJack
DESCRIPTION "Implementation of this optional group is recommended for MAUs that have one or more external jacks."
GROUP mauIfGrpAutoNeg2
DESCRIPTION "Implementation of this group is mandatory for MAUs that support managed auto-negotiation."

GROUP mauIfGrpAutoNeg1000Mbps
DESCRIPTION "Implementation of this group is mandatory for MAUs that have 1000Mb/s or greater capability and support managed auto-negotiation."

GROUP ifMauNotifications
DESCRIPTION "Implementation of this group is recommended for MAUs attached to interfaces."

OBJECT ifMauStatus
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."
::= { mauModCompls 5 }

END

5. IANA-Maintained MAU TC Definitions

IANA-MAU-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-IDENTITY, mib-2
   FROM SNMPv2-SMI
   TEXTUAL-CONVENTION
   FROM SNMPv2-TC

ianaMauMIB MODULE-IDENTITY
LAST-UPDATED "200704210000Z" -- April 21, 2007
ORGANIZATION "IANA"
CONTACT-INFO "Internet Assigned Numbers Authority

Postal: ICANN
   4676 Admiralty Way, Suite 330
   Marina del Rey, CA 90292

   Tel: +1-310-823-9358
   EMail: iana@iana.org"

DESCRIPTION
 "This MIB module defines dot3MauType OBJECT-IDENTITIES and IANAIfMauListBits, IANAIfMauMediaAvailable, IANAIfMauAutoNegCapBits, and IANAIfJackType
TEXTUAL-CONVENTIONS, specifying enumerated values of the ifMauTypeListBits, ifMauMediaAvailable / rpMauMediaAvailable, ifMauAutoNegCapabilityBits / ifMauAutoNegCapAdvertisedBits / ifMauAutoNegCapReceivedBits and ifJackType / rpJackType objects respectively, defined in the MAU-MIB.

It is intended that each new MAU type, Media Availability state, Auto Negotiation capability and/or Jack type defined by the IEEE 802.3 working group and approved for publication in a revision of IEEE Std 802.3 will be added to this MIB module, provided that it is suitable for being managed by the base objects in the MAU-MIB. An Expert Review, as defined in RFC 2434 [RFC2434], is REQUIRED for such additions.

The following reference is used throughout this MIB module:

[IEEE802.3] refers to:
IEEE Std 802.3, 2005 Edition: 'IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications'.

This reference should be updated as appropriate when new MAU types, Media Availability states, Auto Negotiation capabilities, and/or Jack types are added to this MIB module.

Copyright (C) The IETF Trust (2007).
The initial version of this MIB module was published in RFC 4836; for full legal notices see the RFC itself. Supplementary information may be available at: http://www.ietf.org/copyrights/ianamib.html

REVISION     "200704210000Z"  -- April 21, 2007
DESCRIPTION  "Initial version of this MIB as published in RFC 4836."
::= { mib-2 154 }

-- Textual Conventions
IANAIfMauTypeListBits ::= TEXTUAL-CONVENTION
  STATUS current
  DESCRIPTION "This data type is used as the syntax of the ifMauTypeListBits object in the (updated) definition of MAU-MIB's ifMauTable."
The most recent version of this textual convention is available in the online version of this MIB module on the IANA web site.

Requests for new values should be made to IANA via email (iana@iana.org).

Note that changes in this textual convention SHALL be synchronized with relevant changes in the dot3MauType OBJECT-IDENTITIES.

REFERENCE
"[IEEE802.3], Section 30.5.1.1.2"

SYNTAX       BITS {
bOther(0),          -- other or unknown
bAUI(1),            -- AUI
b10base5(2),        -- 10BASE-5
bFoirl(3),          -- FOIRL
b10base2(4),        -- 10BASE-2
b10baseT(5),        -- 10BASE-T duplex mode unknown
b10baseFP(6),       -- 10BASE-FP
b10baseFB(7),       -- 10BASE-FB
b10baseFL(8),       -- 10BASE-FL duplex mode unknown
b10broad36(9),      -- 10BROAD36
b10baseTHD(10),     -- 10BASE-T half duplex mode
b10baseTFD(11),     -- 10BASE-T full duplex mode
b10baseFLHD(12),    -- 10BASE-FL half duplex mode
b10baseFLFD(13),    -- 10BASE-FL full duplex mode
b100baseT4(14),     -- 100BASE-T4
b100baseTXHD(15),   -- 100BASE-TX half duplex mode
b100baseTF(16),     -- 100BASE-TX full duplex mode
b100baseFXHD(17),   -- 100BASE-FX half duplex mode
b100baseFXFD(18),   -- 100BASE-FX full duplex mode
b100baseT2HD(19),   -- 100BASE-T2 half duplex mode
b100baseT2FD(20),   -- 100BASE-T2 full duplex mode
b1000baseXHD(21),   -- 1000BASE-X half duplex mode
b1000baseXF(22),    -- 1000BASE-X full duplex mode
b1000baseLXHD(23),  -- 1000BASE-LX half duplex mode
b1000baseLXFD(24),  -- 1000BASE-LX full duplex mode
b1000baseSXHD(25),  -- 1000BASE-SX half duplex mode
b1000baseSXF(26),   -- 1000BASE-SX full duplex mode
b1000baseCXHD(27),  -- 1000BASE-CX half duplex mode
b1000baseC XF(28),  -- 1000BASE-CX full duplex mode
b1000baseTHD(29),   -- 1000BASE-T half duplex mode
b1000baseTFD(30),   -- 1000BASE-T full duplex mode
b10GbaseX(31),      -- 10GBASE-X
b10GbaseLX4(32),    -- 10GBASE-LX4
IANAifMauMediaAvailable ::= TEXTUAL-CONVENTION
STATUS       current
DESCRIPTION
"This data type is used as the syntax of the
ifMauMediaAvailable and rpMauMediaAvailable objects in the
(updated) definition of MAU-MIB’s ifMauTable and rpMauTable
respectively.

Possible values are:
other(1)             - undefined (not listed below)
unknown(2)           - MAU’s true state is unknown; e.g.,
during initialization
available(3)         - link, light, or loopback is normal
notAvailable(4)      - link loss, low light, or no loopback
remoteFault(5)       - a fault has been detected at the
remote end of the link. This value
applies to 10BASE-FB, 100BASE-T4 Far
End Fault Indication and non-specified
remote faults from a system running
auto-negotiation
invalidSignal(6)     - invalid signal has been received from
the other end of the link, 10BASE-FB
only
remoteJabber(7)      - remote fault, due to jabber
remoteLinkLoss(8) - remote fault, due to link loss
remoteTest(9) - remote fault, due to test
offline(10) - offline, Clause 37 Auto-Negotiation only
autoNegError(11) - Auto-Negotiation Error, Clause 37 Auto-Negotiation only
pmdLinkFault(12) - PMA/PMD receive link fault. In case of PAF (2BASE-TL / 10PASS-TS PHYs), all PMEs in the aggregation group have detected a link fault
wisFrameLoss(13) - WIS loss of frame, 10GBASE-W only
wisSignalLoss(14) - WIS loss of signal, 10GBASE-W only
pcsLinkFault(15) - PCS receive link fault
excessiveBER(16) - PCS Bit Error Ratio monitor reporting excessive error ratio
dxsLinkFault(17) - DTE XGXS receive link fault, XAUI only
pxsLinkFault(18) - PHY XGXS receive link fault, XAUI only
availableReduced(19) - link normal, reduced bandwidth, 2BASE-TL / 10PASS-TS only
ready(20) - at least one PME in the aggregation group is detecting handshake tones, 2BASE-TL / 10PASS-TS only

If the MAU is a 10M b/s link or fiber type (FOIRL, 10BASE-T, 10BASE-F), then this is equivalent to the link test fail state/low light function. For an AUI, 10BASE2, 10BASE5, or 10BROAD36 MAU, this indicates whether loopback is detected on the DI circuit. The value of this attribute persists between packets for MAU types AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASEFP.

At power-up or following a reset, the Media Available state will be unknown(2) for AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP MAUs. For these MAUs loopback will be tested on each transmission during which no collision is detected. If DI is receiving input when DO returns to IDL after a transmission and there has been no collision during the transmission, then loopback will be detected. The Media Available state will only change during noncollided transmissions for AUI, 10BASE2, 10BASE5, 10BROAD36, and 10BASE-FP MAUs.

For 100BASE-T2, 100BASE-T4, 100BASE-TX, 100BASE-FX, 100BASE-LX10, and 100BASE-BX10 PHYs the enumerations match the states within the link integrity state diagram. Any MAU that implements management of [IEEE802.3] Clause 28 Auto-Negotiation, will map remote fault indication to remoteFault(5).
Any MAU that implements management of Clause 37 Auto-Negotiation, will map the received RF1 and RF2 bits as follows: Offline maps to offline(10), Link_Failure maps to remoteFault(5), and Auto-Negotiation Error maps to autoNegError(11).

The value remoteFault(5) applies to 10BASE-FB remote fault indication, the 100BASE-X far-end fault indication, and nonspecified remote faults from a system running Clause 28 Auto-Negotiation.

The value remoteJabber(7), remoteLinkLoss(8), or remoteTest(9) SHOULD be used instead of remoteFault(5) where the reason for remote fault is identified in the remote signaling protocol. Where a Clause 22 MII or Clause 35 GMII is present, a logic one in the remote fault bit maps to the value remoteFault(5), a logic zero in the link status bit maps to the enumeration notAvailable(4). The value notAvailable(4) takes precedence over remoteFault(5).

For 2BASE-TL and 10PASS-TS PHYs, the value unknown(2) maps to the condition where the PHY (PCS with connected PMEs) is initializing, the value ready(20) maps to the condition where the interface is down and at least one PME in the aggregation group is ready for handshake, the value available(3) maps to the condition where all the PMEs in the aggregation group are up, the value notAvailable(4) maps to the condition where all the PMEs in the aggregation group are down and no handshake tones are detected, the value availableReduced(19) maps to the condition where the interface is up, a link fault is detected at the receive direction by one or more PMEs in the aggregation group, but at least one PME is up and the enumeration pmdLinkFault(12) maps to the condition where a link fault is detected at the receive direction by all of the PMEs in the aggregation group.

For 10 Gb/s the enumerations map to value of the link_fault variable within the Link Fault Signaling state diagram as follows: the value OK maps to the value available(3), the value Local Fault maps to the value notAvailable(4), and the value Remote Fault maps to the value remoteFault(5). The value pmdLinkFault(12), wisFrameLoss(13), wisSignalLoss(14), pcsLinkFault(15), excessiveBER(16), or dxsLinkFault(17) SHOULD be used instead of the value notAvailable(4), where the reason for the Local Fault state can be identified through the use of the Clause 45 MDIO Interface. Where multiple reasons for the Local Fault state can be identified, only the highest precedence error SHOULD be
reported. This precedence in descending order is as follows:

pxsLinkFault
pmdLinkFault
wisFrameLoss
wisSignalLoss
pcsLinkFault
excessiveBER
dxsLinkFault.

Where a Clause 45 MDIO interface is present a logic zero in the PMA/PMD Receive link status bit ([IEEE802.3] Section 45.2.1.2.2) maps to the value pmdLinkFault(12), logic one in the LOF status bit (Section 45.2.2.10.4) maps to the value wisFrameLoss(13), a logic one in the LOS status bit (Section 45.2.2.10.5) maps to the value wisSignalLoss, a logic zero in the PCS Receive link status bit (Section 45.2.3.2.2) maps to the value pcsLinkFault(15), a logic one in the 10GBASE-R PCS Latched high BER status bit (Section 45.2.3.12.2) maps to the value excessiveBER, a logic zero in the DTE XS receive link status bit (Section 45.2.5.2.2) maps to the value dxsLinkFault(17) and a logic zero in the PHY XS transmit link status bit (Section 45.2.4.2.2) maps to the value pxsLinkFault(18).

The most recent version of this textual convention is available in the online version of this MIB module on the IANA web site.

Requests for new values should be made to IANA via email (iana@iana.org)."

REFERENCE
"[IEEE802.3], Section 30.5.1.1.4"

SYNTAX       INTEGER {
    other(1),
    unknown(2),
    available(3),
    notAvailable(4),
    remoteFault(5),
    invalidSignal(6),
    remoteJabber(7),
    remoteLinkLoss(8),
    remoteTest(9),
    offline(10),
    autoNegError(11),
    pmdLinkFault(12),
    wisFrameLoss(13),
    wisSignalLoss(14),
    pcsLinkFault(15),
IANAifMauAutoNegCapBits ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
"This data type is used as the syntax of the
ifMauAutoNegCapabilityBits, ifMauAutoNegCapAdvertisedBits, and
ifMauAutoNegCapReceivedBits objects in the (updated) definition
of MAU-MIB’s ifMauAutoNegTable.

The most recent version of this textual convention is available
in the online version of this MIB module on the IANA web site.

Requests for new values should be made to IANA via email
(iana@iana.org)."
REFERENCE
"[IEEE802.3], Section 30.6.1.1.5"
SYNTAX BITS {
bOther(0), -- other or unknown
b10baseT(1), -- 10BASE-T half duplex mode
b10baseTFD(2), -- 10BASE-T full duplex mode
b100baseT4(3), -- 100BASE-T4
b100baseTX(4), -- 100BASE-TX half duplex mode
b100baseTXFD(5), -- 100BASE-TX full duplex mode
b100baseT2(6), -- 100BASE-T2 half duplex mode
b100baseT2FD(7), -- 100BASE-T2 full duplex mode
bFdxPause(8), -- PAUSE for full-duplex links
bFdxAPause(9), -- Asymmetric PAUSE for full-duplex links
bFdxSPause(10), -- Symmetric PAUSE for full-duplex links
bFdxBPause(11), -- Asymmetric and Symmetric PAUSE for full-duplex links
b1000baseX(12), -- 1000BASE-X, -LX, -SX, -CX half duplex mode
b1000baseXFD(13), -- 1000BASE-X, -LX, -SX, -CX full duplex mode
b1000baseT(14), -- 1000BASE-T half duplex mode
b1000baseTFD(15) -- 1000BASE-T full duplex mode
}
IANAifJackType ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION

"Common enumeration values for repeater and interface MAU jack types. This data type is used as the syntax of the ifJackType and rpJackType objects in the (updated) definition of MAU-MIB’s ifJackTable and rpJackTable respectively.

Possible values are:
other(1) - undefined or unknown
rj45(2) - RJ45
rj45S(3) - RJ45 shielded
db9(4) - DB9
bnc(5) - BNC
faUI(6) - AUI female
mAUI(7) - AUI male
fiberSC(8) - SC fiber
fiberMIC(9) - MIC fiber
fiberST(10) - ST fiber
telco(11) - Telco
mtrj(12) - MT-RJ fiber
hssdc(13) - fiber channel style-2
fiberLC(14) - LC fiber
cx4(15) - IB4X for 10GBASE-CX4

The most recent version of this textual convention is available in the online version of this MIB module on the IANA web site.

Requests for new values should be made to IANA via email (iana@iana.org)."

SYNTAX    INTEGER {
    other(1),
    rj45(2),
    rj45S(3),
    db9(4),
    bnc(5),
    faUI(6),
    mAUI(7),
    fiberSC(8),
    fiberMIC(9),
    fiberST(10),
    telco(11),
    mtrj(12),
    hssdc(13),
    fiberLC(14),
    -- new since RFC 3636
    cx4(15)
}

-- OBJECT IDENTITIES for MAU types
-- (see rpMauType and ifMauType of MAU-MIB for usage)
-- The following definitions has been moved from RFC 3636 and
-- no longer appear in its revision.

dot3MauType OBJECT IDENTIFIER ::= { mib-2 snmpDot3MauMgt(26) 4 }

...
dot3MauType10BaseFB OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "sync fiber MAU"
   REFERENCE "[IEEE802.3], Section 17"
   ::= { dot3MauType 7 }

dot3MauType10BaseFL OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "async fiber MAU.
   Note that it is strongly recommended that
   agents return either dot3MauType10BaseFLHD or
   dot3MauType10BaseFLFD if the duplex mode is
   known. However, management applications should
   be prepared to receive this MAU type value from
   older agent implementations."
   REFERENCE "[IEEE802.3], Section 18"
   ::= { dot3MauType 8 }

dot3MauType10Broad36 OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "broadband DTE MAU.
   Note that 10BROAD36 MAUs can be attached to
   interfaces but not to repeaters."
   REFERENCE "[IEEE802.3], Section 11"
   ::= { dot3MauType 9 }

------ new since RFC 1515:
dot3MauType10BaseTHD OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "UTP MAU, half duplex mode"
   REFERENCE "[IEEE802.3], Section 14"
   ::= { dot3MauType 10 }

dot3MauType10BaseTFD OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "UTP MAU, full duplex mode"
   REFERENCE "[IEEE802.3], Section 14"
   ::= { dot3MauType 11 }

dot3MauType10BaseFLHD OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "async fiber MAU, half duplex mode"
   REFERENCE "[IEEE802.3], Section 18"
   ::= { dot3MauType 12 }

dot3MauType10BaseFLFD OBJECT-IDENTITY
   STATUS current
   DESCRIPTION "async fiber MAU, full duplex mode"
REFERENCE "[IEEE802.3], Section 18"
::= { dot3MauType 13 }

dot3MauType100BaseT4 OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "4 pair category 3 UTP"
  REFERENCE "[IEEE802.3], Section 23"
::= { dot3MauType 14 }

dot3MauType100BaseTXHD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "2 pair category 5 UTP, half duplex mode"
  REFERENCE "[IEEE802.3], Section 25"
::= { dot3MauType 15 }

dot3MauType100BaseTXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "2 pair category 5 UTP, full duplex mode"
  REFERENCE "[IEEE802.3], Section 25"
::= { dot3MauType 16 }

dot3MauType100BaseFXHD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "X fiber over PMT, half duplex mode"
  REFERENCE "[IEEE802.3], Section 26"
::= { dot3MauType 17 }

dot3MauType100BaseFXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "X fiber over PMT, full duplex mode"
  REFERENCE "[IEEE802.3], Section 26"
::= { dot3MauType 18 }

dot3MauType100BaseT2HD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "2 pair category 3 UTP, half duplex mode"
  REFERENCE "[IEEE802.3], Section 32"
::= { dot3MauType 19 }

dot3MauType100BaseT2FD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "2 pair category 3 UTP, full duplex mode"
  REFERENCE "[IEEE802.3], Section 32"
::= { dot3MauType 20 }

------ new since RFC 2239:
dot3MauType1000BaseXHD OBJECT-IDENTITY
  STATUS current
DESCRIPTION "PCS/PMA, unknown PMD, half duplex mode"
REFERENCE "[IEEE802.3], Section 36"
::= { dot3MauType 21 }

dot3MauType1000BaseXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "PCS/PMA, unknown PMD, full duplex mode"
  REFERENCE "[IEEE802.3], Section 36"
::= { dot3MauType 22 }

dot3MauType1000BaseLXHD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Fiber over long-wavelength laser, half duplex mode"
  REFERENCE "[IEEE802.3], Section 38"
::= { dot3MauType 23 }

dot3MauType1000BaseLXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Fiber over long-wavelength laser, full duplex mode"
  REFERENCE "[IEEE802.3], Section 38"
::= { dot3MauType 24 }

dot3MauType1000BaseSXHD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Fiber over short-wavelength laser, half duplex mode"
  REFERENCE "[IEEE802.3], Section 38"
::= { dot3MauType 25 }

dot3MauType1000BaseSXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Fiber over short-wavelength laser, full duplex mode"
  REFERENCE "[IEEE802.3], Section 38"
::= { dot3MauType 26 }

dot3MauType1000BaseCXHD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Copper over 150-Ohm balanced cable, half duplex mode"
  REFERENCE "[IEEE802.3], Section 39"
::= { dot3MauType 27 }

dot3MauType1000BaseCXFD OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Copper over 150-Ohm balanced cable, full
duplex mode
REFERENCE "[IEEE802.3], Section 39"
::= { dot3MauType 28 }

dot3MauType1000BaseTHD OBJECT-IDENTITY
STATUS current
DESCRIPTION "Four-pair Category 5 UTP, half duplex mode"
REFERENCE "[IEEE802.3], Section 40"
::= { dot3MauType 29 }

dot3MauType1000BaseTFD OBJECT-IDENTITY
STATUS current
DESCRIPTION "Four-pair Category 5 UTP, full duplex mode"
REFERENCE "[IEEE802.3], Section 40"
::= { dot3MauType 30 }

------ new since RFC 2668:
dot3MauType10GigBaseX OBJECT-IDENTITY
STATUS current
DESCRIPTION "X PCS/PMA, unknown PMD."
REFERENCE "[IEEE802.3], Section 48"
::= { dot3MauType 31 }

dot3MauType10GigBaseLX4 OBJECT-IDENTITY
STATUS current
DESCRIPTION "X fiber over WWDM optics"
REFERENCE "[IEEE802.3], Section 53"
::= { dot3MauType 32 }

dot3MauType10GigBaseR OBJECT-IDENTITY
STATUS current
DESCRIPTION "R PCS/PMA, unknown PMD."
REFERENCE "[IEEE802.3], Section 49"
::= { dot3MauType 33 }

dot3MauType10GigBaseER OBJECT-IDENTITY
STATUS current
DESCRIPTION "R fiber over 1550 nm optics"
REFERENCE "[IEEE802.3], Section 52"
::= { dot3MauType 34 }

dot3MauType10GigBaseLR OBJECT-IDENTITY
STATUS current
DESCRIPTION "R fiber over 1310 nm optics"
REFERENCE "[IEEE802.3], Section 52"
::= { dot3MauType 35 }

dot3MauType10GigBaseSR OBJECT-IDENTITY
STATUS  current
DESCRIPTION "R fiber over 850 nm optics"
REFERENCE  "[IEEE802.3], Section 52"
::= { dot3MauType 36 }

dot3MauType10GigBaseW OBJECT-IDENTITY
STATUS  current
DESCRIPTION "W PCS/PMA, unknown PMD."
REFERENCE  "[IEEE802.3], Section 49 and 50"
::= { dot3MauType 37 }

dot3MauType10GigBaseEW OBJECT-IDENTITY
STATUS  current
DESCRIPTION "W fiber over 1550 nm optics"
REFERENCE  "[IEEE802.3], Section 52"
::= { dot3MauType 38 }

dot3MauType10GigBaseLW OBJECT-IDENTITY
STATUS  current
DESCRIPTION "W fiber over 1310 nm optics"
REFERENCE  "[IEEE802.3], Section 52"
::= { dot3MauType 39 }

dot3MauType10GigBaseSW OBJECT-IDENTITY
STATUS  current
DESCRIPTION "W fiber over 850 nm optics"
REFERENCE  "[IEEE802.3], Section 52"
::= { dot3MauType 40 }

------- new since RFC 3636:
dot3MauType10GigBaseCX4 OBJECT-IDENTITY
STATUS  current
DESCRIPTION "X copper over 8 pair 100-Ohm balanced cable"
REFERENCE  "[IEEE802.3], Section 54"
::= { dot3MauType 41 }

dot3MauType2BaseTL OBJECT-IDENTITY
STATUS  current
DESCRIPTION "Voice grade UTP copper, up to 2700m, optional PAF"
REFERENCE  "[IEEE802.3], Sections 61 and 63"
::= { dot3MauType 42 }

dot3MauType10PassTS OBJECT-IDENTITY
STATUS  current
DESCRIPTION "Voice grade UTP copper, up to 750m, optional PAF"
REFERENCE  ",[IEEE802.3], Sections 61 and 62"
::= { dot3MauType 43 }
dot3MauType100BaseBX10D OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber OLT, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 58"
  ::= { dot3MauType 44 }

dot3MauType100BaseBX10U OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber ONU, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 58"
  ::= { dot3MauType 45 }

dot3MauType100BaseLX10 OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Two single-mode fibers, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 58"
  ::= { dot3MauType 46 }

dot3MauType1000BaseBX10D OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber OLT, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 59"
  ::= { dot3MauType 47 }

dot3MauType1000BaseBX10U OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber ONU, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 59"
  ::= { dot3MauType 48 }

dot3MauType1000BaseLX10 OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "Two single-mode fibers, long wavelength, 10km"
  REFERENCE "[IEEE802.3], Section 59"
  ::= { dot3MauType 49 }

dot3MauType1000BasePX10D OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber EPON OLT, 10km"
  REFERENCE "[IEEE802.3], Section 60"
  ::= { dot3MauType 50 }

dot3MauType1000BasePX10U OBJECT-IDENTITY
  STATUS current
  DESCRIPTION "One single-mode fiber EPON ONU, 10km"
  REFERENCE "[IEEE802.3], Section 60"
  ::= { dot3MauType 51 }
6. Security Considerations

The IANA-MAU-MIB does not define any management objects. Instead, it defines a set of textual conventions which are used by the MAU-MIB and may be used by other MIB modules to define management objects. Meaningful security considerations can only be written for MIB modules that define management objects.

There are a number of management objects defined in the MAU-MIB that have a MAX-ACCESS clause of read-write. Setting these objects can have a serious effect on the operation of the network, including:

- enabling or disabling a MAU
- changing a MAU’s default type
- enabling, disabling, or restarting autonegotiation
- modifying the capabilities that a MAU advertizes during autonegotiation.

Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

Some of the readable objects in the MAU-MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. In some environments, it may be undesirable to allow unauthorized parties to access statistics or status information about individual links in a network. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.
SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in the MAU-MIB module.

It is RECOMMENDED that the implementors consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Furthermore, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of the MAU-MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

7. IANA Considerations

This document defines first version of the IANA-maintained IANA-MAU-MIB module. It is intended that each new MAU type, Media Available state, Auto Negotiation capability, and/or Jack type defined by the IEEE 802.3 working group and approved for publication in a revision of IEEE Std 802.3 will be added to the IANA-maintained MIB module, provided that it is suitable for being managed by the base objects in the MAU-MIB module.

For each new MAU type added, a short description of the MAU technology and, wherever possible, a reference to a publicly available specification SHOULD be specified. An Expert Review, as defined in RFC 2434 [RFC2434], is REQUIRED, for each modification.

8. Acknowledgments

This document was produced by the IETF Ethernet Interfaces and Hub MIB Working Group, whose efforts were greatly advanced by the contributions of the following people:

Mike Heard
John Flick
Dan Romascanu

This document is based on the Proposed Standard MAU MIB, RFC 3636 [RFC3636], edited by John Flick of Hewlett-Packard, and produced by
the Ethernet Interfaces and Hub MIB Working Group. It extends that
document by moving the object identities and textual conventions for
MAU types into a IANA-maintained MIB module. In addition, support is
provided for the EFM and 10GBASE-CX4 MAUs as defined in [IEEE802.3ah]
and [IEEE802.3ak] respectively.

RFC 3636, in turn, was based on the Proposed Standard MAU MIB, RFC
2668 [RFC2668], edited by John Flick of Hewlett-Packard and Andrew
Smith, then of Extreme Networks, and produced by the Ethernet
Interfaces and Hub MIB Working Group. It extends that document by
providing support for 10 Gb/s MAUs as defined in [IEEE802.3ae].

RFC 2668, in turn, was based on the Proposed Standard MAU MIB, RFC
2239 [RFC2239], edited by Kathryn de Graaf, then of 3Com, and Dan
Romascanu, then of Madge Networks, and produced by the Ethernet
Interfaces and Hub MIB Working Group. It extended that document by
providing support for 1000 Mb/sec MAUs, PAUSE negotiation and remote
fault status as defined in [IEEE802.3].

RFC 2239, in turn, was based on the Proposed Standard MAU MIB, RFC
1515 [RFC1515], edited by Donna McMaster, then of SynOptics
Communications, Keith McCloughrie, then of Hughes LAN Systems, and Sam
Roberts, then of Farallon Computing, and produced by the Hub MIB
Working Group. It extends that document by providing support for 100
Mb/sec MAUs, full duplex MAUs, auto-negotiation, and jack management
as defined in [IEEE802.3].

9. References

9.1. Normative References

[IEEE802.3] IEEE, "IEEE Standard for Information technology -
Telecommunications and information exchange between
systems - Local and metropolitan area networks -
Specific requirements - Part 3: Carrier sense multiple
access with collision detection (CSMA/CD) access
method and physical layer specifications", IEEE
Std 802.3-2005, December 2005.

[IEEE802.3ae] IEEE, "IEEE Standard for Information technology -
Telecommunications and information exchange between
systems - Local and metropolitan area networks -
Specific requirements - Part 3: Carrier sense multiple
access with collision detection (CSMA/CD) access
method and physical layer specifications - Media
Access Control (MAC) Parameters, Physical Layer
and Management Parameters for 10 Gb/s Operation", IEEE
Std 802.3ae-2002, August 2002.


9.2. Informative References


[RFC3636] Flick, J., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 3636, September 2003.


Author’s Address

Edward Beili
Actelis Networks
Bazel 25
Petach-Tikva
Israel

Phone: +972-3-924-3491
EMail: edward.beili@actelis.com
Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.