Network Working Group
Request for Comments: 2239
Category: Standards Track

K. de Graaf
3Com Corporation
D. Romascanu
Madge Networks Ltd.
D. McMaster
Cisco Systems Inc.
K. McCloghrie
Cisco Systems Inc.
S. Roberts
Farallon Computing, Inc.

November 1997

Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs) using SMIv2

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 Internet Society (1997). All Rights Reserved.

Table of Contents

1 The SNMPv2 Network Management Framework 2
1.1 Object Definitions 2
2 Overview 3
2.1 Relationship to RFC 1515 3
2.2 MAU Management 3
2.3 Relationship to Other MIBs 3
2.3.1 Relationship to the MIB-II 'interfaces' group 3
2.3.2 Relationship to the 802.3 Repeater MIB 4
2.4 Management of Internal MAUs 4
3 Definitions 4
4 Acknowledgements 39
5 References 40
6 Security Considerations 41
7 Authors' Addresses 41
8 Full Copyright Statement 43

Abstract

This memo defines an portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing 10 and 100 Mb/second Medium Attachment Units (MAUs) based on IEEE Std 802.3 Section 30, "10 & 100 Mb/s Management," October 26, 1995.

1. The SNMPv2 Network Management Framework

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

- the SMI, described in RFC 1902 [6] - the mechanisms used for describing and naming objects for the purpose of management.
- the MIB-II, STD 17, RFC 1213 [5] - the core set of managed objects for the Internet suite of protocols.
- the protocol, STD 15, RFC 1157 [10] and/or RFC 1905 [9] - the protocol used for accessing managed information.

Textual conventions are defined in RFC 1903 [7], and conformance statements are defined in RFC 1904 [8].

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

1.1. Object Definitions

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

2. Overview

2.1. Relationship to RFC 1515

This MIB is intended to be a superset of that defined by RFC 1515 [11], which will go to historic status. This MIB includes all of the objects contained in that MIB, plus several new ones which provide additional capabilities. Implementors are encouraged to support all applicable conformance groups in order to make the best use of the new functionality provided by this MIB. The new objects provide management support for:

- management of 100 Mb/s devices
- auto-negotiation on interface MAUs
- jack management

2.2. MAU Management

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 [1] and [2]. 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 & 100 Mb/s Medium Attachment Units (MAUs)", and Annex 30A, "GDMO Specifications for 802.3 managed objects" of IEEE Std 802.3u-1995. That specification includes definitions for both 10Mb/s and 100Mb/s devices, and is essentially a superset of the 10Mb/s definitions given by IEEE 802.3 Section 20. This specification is intended to serve the same purpose: to provide for management of both 10Mb/s and 100Mb/s MAUs.

2.3. Relationship to Other MIBs

It is assumed that an agent implementing this MIB will also implement (at least) the 'system' group defined in MIB-II [5]. The following sections identify other MIBs that such an agent should implement.

2.3.1. Relationship to the MIB-II 'interfaces' group

The sections of this document that define interface MAU-related objects specify an extension to the 'interfaces' group of MIB-II. An agent implementing these interface-MAU related objects must also

implement the 'interfaces' group of MIB-II. 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 expected that an agent implementing the interface-MAU related objects in this MIB will also implement the Ethernet- like Interfaces MIB, RFC 1650.

(Note that repeater ports are not represented as interfaces in the sense of MIB-II's 'interfaces' group.)

2.3.2. Relationship to the 802.3 Repeater MIB

The section of this document that defines repeater MAU-related objects specifies an extension to the 802.3 Repeater MIB defined in [4]. An agent implementing these repeater-MAU related objects must also implement the 802.3 Repeater MIB.

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 to which the given MAU is connected.

2.4. 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. Definitions

```
MAU-MIB DEFINITIONS ::= BEGIN

IMPORTS
    Counter32, Integer32,
    OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE,
    OBJECT-IDENTITY, mib-2
        FROM SNMPv2-SMI
    TruthValue, TEXTUAL-CONVENTION
```

```
        FROM SNMPv2-TC
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
        FROM SNMPv2-CONF;

mauMod MODULE-IDENTITY
    LAST-UPDATED "9710310000Z"
    ORGANIZATION "IETF HUB MIB Working Group"
    CONTACT-INFO
        "WG E-mail: hubmib@hprnd.rose.hp.com

             Chair: Dan Romascanu
            Postal: Madge Networks (Israel) Ltd.
                    Atidim Technology Park, Bldg. 3
                    Tel Aviv 61131, Israel
               Tel: 972-3-6458414, 6458458
               Fax: 972-3-6487146
            E-mail: dromasca@madge.com

            Editor: Kathryn de Graaf
            Postal: 3Com Corporation
                    118 Turnpike Rd.
                    Southborough, MA  01772
                    USA
               Tel: (508)229-1627
               Fax: (508)490-5882
            E-mail: kdegraaf@isd.3com.com"
    DESCRIPTION
            "Management information for 802.3 MAUs.

            The following references are used throughout this
            MIB module:

            [IEEE 802.3 Std]
               refers to IEEE 802.3/ISO 8802-3 Information
               processing systems - Local area networks -
               Part 3: Carrier sense multiple access with
               collision detection (CSMA/CD) access method
               and physical layer specifications (1993),
               and to IEEE Std 802.3u-1995, Supplement to
               IEEE Std 802.3, clauses 22 through 29.

            [IEEE 802.3 Mgt]
               refers to IEEE 802.3u-1995, - 10 Mb/s &
               100 Mb/s Management, Section 30 -
               Supplement to IEEE Std 802.3."
    ::= { snmpDot3MauMgt 6 }
```

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


   -- textual conventions

   JackType ::= TEXTUAL-CONVENTION
       STATUS     current
       DESCRIPTION
               "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)
                  }


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

   dot3IfMauAutoNegGroup       OBJECT IDENTIFIER ::= { snmpDot3MauMgt 5 }

   -- object identities for MAU types
   --  (see rpMauType and ifMauType for usage)

   dot3MauType
       OBJECT IDENTIFIER ::= { snmpDot3MauMgt 4 }

   dot3MauTypeAUI OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "no internal MAU, view from AUI"
       ::= { dot3MauType 1 }

   dot3MauType10Base5 OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "thick coax MAU (per 802.3 section 8)"
       ::= { dot3MauType 2 }
```

```
   dot3MauTypeFoirl OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "FOIRL MAU (per 802.3 section 9.9)"
       ::= { dot3MauType 3 }

   dot3MauType10Base2 OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "thin coax MAU (per 802.3 section 10)"
       ::= { dot3MauType 4 }

   dot3MauType10BaseT OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "UTP MAU (per 802.3 section 14)"
       ::= { dot3MauType 5 }

   dot3MauType10BaseFP OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "passive fiber MAU (per 802.3 section 16)"
       ::= { dot3MauType 6 }

   dot3MauType10BaseFB OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "sync fiber MAU (per 802.3 section 17)"
       ::= { dot3MauType 7 }

   dot3MauType10BaseFL OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "async fiber MAU (per 802.3 section 18)"
       ::= { dot3MauType 8 }

   dot3MauType10Broad36 OBJECT-IDENTITY
       STATUS     current
       DESCRIPTION
               "broadband DTE MAU (per 802.3 section 11). Note
               that 10BROAD36 MAUs can be attached to interfaces
               but not to repeaters."
       ::= { dot3MauType 9 }

   ------ new since RFC 1515:

   dot3MauType10BaseTHD OBJECT-IDENTITY
       STATUS     current
```

```
        DESCRIPTION
                "UTP MAU (per 802.3 section 14), half duplex mode"
        ::= { dot3MauType 10 }

    dot3MauType10BaseTFD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "UTP MAU (per 802.3 section 14), full duplex mode"
        ::= { dot3MauType 11 }

    dot3MauType10BaseFLHD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "async fiber MAU (per 802.3 section 18), half
                duplex mode"
        ::= { dot3MauType 12 }

    dot3MauType10BaseFLFD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "async fiber MAU (per 802.3 section 18), full
                duplex mode"
        ::= { dot3MauType 13 }

    dot3MauType100BaseT4 OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "4 pair categ. 3 UTP (per 802.3 section 23)"
        ::= { dot3MauType 14 }

    dot3MauType100BaseTXHD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "2 pair categ. 5 UTP (per 802.3 section 25), half
                duplex mode"
        ::= { dot3MauType 15 }

    dot3MauType100BaseTXFD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "2 pair categ. 5 UTP (per 802.3 section 25), full
                duplex mode"
        ::= { dot3MauType 16 }

    dot3MauType100BaseFXHD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "X fiber over PMT (per 802.3 section 26), half
```

```
                duplex mode"
        ::= { dot3MauType 17 }

    dot3MauType100BaseFXFD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "X fiber over PMT (per 802.3 section 26), full
                duplex mode"
        ::= { dot3MauType 18 }

    dot3MauType100BaseT2HD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "2 pair categ. 3 UTP (per 802.3 section 32), half
                duplex mode"
        ::= { dot3MauType 19 }

    dot3MauType100BaseT2FD OBJECT-IDENTITY
        STATUS     current
        DESCRIPTION
                "2 pair categ. 3 UTP (per 802.3 section 32), full
                duplex mode"
        ::= { dot3MauType 20 }


    --
    -- 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, rpMauIndex }
        ::= { rpMauTable 1 }
```

```
   RpMauEntry ::=
       SEQUENCE {
           rpMauGroupIndex
               Integer32,
           rpMauPortIndex
               Integer32,
           rpMauIndex
               Integer32,
           rpMauType
               OBJECT IDENTIFIER,
           rpMauStatus
               INTEGER,
           rpMauMediaAvailable
               INTEGER,
           rpMauMediaAvailableStateExits
               Counter32,
           rpMauJabberState
               INTEGER,
           rpMauJabberingStateEnters
               Counter32,
           rpMauFalseCarriers
               Counter32
       }

   rpMauGroupIndex OBJECT-TYPE
       SYNTAX     Integer32 (1..2147483647)
       MAX-ACCESS read-only
       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."
       ::= { rpMauEntry 1 }

   rpMauPortIndex OBJECT-TYPE
       SYNTAX     Integer32 (1..2147483647)
       MAX-ACCESS read-only
       STATUS     current
```

```
        DESCRIPTION
                "This variable uniquely identifies the repeater
                port within group rpMauGroupIndex to which the MAU
                described by this entry is connected."
        REFERENCE
                "Reference RFC 1516, rptrPortIndex."
        ::= { rpMauEntry 2 }

    rpMauIndex OBJECT-TYPE
        SYNTAX     Integer32 (1..2147483647)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable uniquely identifies the MAU
                described by this entry from among other MAUs
                connected to the same port (rpMauPortIndex)."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.1, aMAUID."
        ::= { rpMauEntry 3 }

    rpMauType OBJECT-TYPE
        SYNTAX     OBJECT IDENTIFIER
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This object identifies the 10 or 100 Mb/s
                baseband MAU type.  An initial set of MAU types
                are defined above.  The assignment of OBJECT
                IDENTIFIERs to new types of MAUs is managed by the
                IANA.  If the MAU type is unknown, the object
                identifier

                unknownMauType OBJECT IDENTIFIER ::= { 0 0 }

                is returned.  Note that unknownMauType is a
                syntactically valid object identifier, and any
                conformant implementation of ASN.1 and the BER
                must be able to generate and recognize this
                value."
        REFERENCE
                "[IEEE 802.3 Mgt], 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.

            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, 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
                "[IEEE 802.3 Mgt], 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     INTEGER {
                       other(1),
                       unknown(2),
                       available(3),
                       notAvailable(4),
                       remoteFault(5),
                       invalidSignal(6),
                       remoteJabber(7),
                       remoteLinkLoss(8),
                       remoteTest(9)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "If the MAU is a 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 or a coax (including broadband) MAU this
                indicates whether or not loopback is detected on
                the DI circuit.  The value of this attribute
                persists between packets for MAU types AUI,
                10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP.

                The value other(1) is returned if the
                mediaAvailable state is not one of 2 through 6.

                The value unknown(2) is returned when the MAU's
                true state is unknown; for example, when it is
                being initialized.  At power-up or following a
                reset, the value of this attribute will be unknown
                for AUI, coax, 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 value of this attribute will only
                change during non-collided transmissions for AUI,
```

```
                coax, and 10BASE-FP MAUs.

                For 100BASE-T4, 100BASE-TX and 100BASE-FX the
                enumerations match the states within the
                respective link integrity state diagrams, fig 23-
                12 and 24-15 of sections 23 and 24 of [2].  Any
                MAU which implements management of auto-
                negotiation will map remote fault indication to
                remote fault.

                The value available(3) indicates that the link,
                light, or loopback is normal.  The value
                notAvailable(4) indicates link loss, low light, or
                no loopback.

                The value remoteFault(5) indicates that 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.
                The values remoteJabber(7), remoteLinkLoss(8), and
                remoteTest(9) should be used instead of
                remoteFault(5) where the reason for remote fault
                is identified in the remote signaling protocol.

                The value invalidSignal(6) indicates that an
                invalid signal has been received from the other
                end of the link.  InvalidSignal(6) applies only to
                MAUs of type 10BASE-FB.

                Where an IEEE Std 802.3u-1995 clause 22 MII is
                present, a logic one in the remote fault bit
                (reference section 22.2.4.2.8 of that document)
                maps to the value remoteFault(5), and a logic zero
                in the link status bit (reference section
                22.2.4.2.10 of that document) maps to the value
                notAvailable(4).  The value notAvailable(4) takes
                precedence over the value remoteFault(5)."
        REFERENCE
                "[IEEE 802.3 Mgt], 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)."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.5, aLoseMediaCounter."
        ::= { 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
                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
                "[IEEE 802.3 Mgt], 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, and dot3MauType100BaseFX,
                this counter will always indicate zero."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.6,
                aJabber.jabberCounter."
```

```
        ::= { rpMauEntry 9 }

    rpMauFalseCarriers 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 links.  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 until the next
                carrier event.

                This counter increments only for MAUs of type
                dot3MauType100BaseT4, dot3MauType100BaseTX, and
                dot3MauType100BaseFX.  For all other MAU types,
                this counter will always indicate zero.

                The approximate minimum time for rollover of this
                counter is 7.4 hours."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.10, aFalseCarriers."
        ::= { rpMauEntry 10 }


    -- The rpJackTable applies to MAUs attached to repeaters
    -- which have one or more external jacks (connectors).

    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
                JackType
        }

    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     JackType
        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
                Integer32,
            ifMauIndex
                Integer32,
            ifMauType
                OBJECT IDENTIFIER,
            ifMauStatus
                INTEGER,
            ifMauMediaAvailable
                INTEGER,
            ifMauMediaAvailableStateExits
                Counter32,
            ifMauJabberState
                INTEGER,
            ifMauJabberingStateEnters
                Counter32,
            ifMauFalseCarriers
                Counter32,
            ifMauTypeList
                Integer32,
            ifMauDefaultType
                OBJECT IDENTIFIER,
            ifMauAutoNegSupported
                TruthValue
        }

    ifMauIfIndex OBJECT-TYPE
        SYNTAX     Integer32 (1..2147483647)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable uniquely identifies the interface
                to which the MAU described by this entry is
                connected."
        REFERENCE
                "RFC 1213, ifIndex"
        ::= { ifMauEntry 1 }
```

```
   ifMauIndex OBJECT-TYPE
       SYNTAX     Integer32 (1..2147483647)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "This variable uniquely identifies the MAU
               described by this entry from among other MAUs
               connected to the same interface (ifMauIfIndex)."
       REFERENCE
               "[IEEE 802.3 Mgt], 30.5.1.1.1, aMAUID."
       ::= { ifMauEntry 2 }

   ifMauType OBJECT-TYPE
       SYNTAX     OBJECT IDENTIFIER
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "This object identifies the 10 or 100 Mb/s
               baseband or broadband MAU type.  An initial set of
               MAU types are defined above.  The assignment of
               OBJECT IDENTIFIERs to new types of MAUs is managed
               by the IANA.  If the MAU type is unknown, the
               object identifier

               unknownMauType OBJECT IDENTIFIER ::= { 0 0 }

               is returned.  Note that unknownMauType is a
               syntactically valid object identifier, and any
               conformant implementation of ASN.1 and the BER
               must be able to generate and recognize this value.

               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
               "[IEEE 802.3 Mgt], 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, 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
                "[IEEE 802.3 Mgt], 30.5.1.1.7, aMAUAdminState,
                30.5.1.2.2, acMAUAdminControl, and 30.5.1.2.1,
                acRESETMAU."
        ::= { ifMauEntry 4 }

    ifMauMediaAvailable OBJECT-TYPE
        SYNTAX     INTEGER {
                       other(1),
                       unknown(2),
                       available(3),
                       notAvailable(4),
                       remoteFault(5),
                       invalidSignal(6),
                       remoteJabber(7),
                       remoteLinkLoss(8),
                       remoteTest(9)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "If the MAU is a 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 or a coax (including broadband) MAU this
                indicates whether or not loopback is detected on
                the DI circuit.  The value of this attribute
                persists between packets for MAU types AUI,
                10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP.

                The value other(1) is returned if the
                mediaAvailable state is not one of 2 through 6.

                The value unknown(2) is returned when the MAU's
                true state is unknown; for example, when it is
                being initialized.  At power-up or following a
                reset, the value of this attribute will be unknown
                for AUI, coax, 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 value of this attribute will only
                change during non-collided transmissions for AUI,
```

```
                coax, and 10BASE-FP MAUs.

                For 100BASE-T4, 100BASE-TX and 100BASE-FX the
                enumerations match the states within the
                respective link integrity state diagrams, fig 23-
                12 and 24-15 of sections 23 and 24 of [2].  Any
                MAU which implements management of auto-
                negotiation will map remote fault indication to
                remote fault.

                The value available(3) indicates that the link,
                light, or loopback is normal.  The value
                notAvailable(4) indicates link loss, low light, or
                no loopback.

                The value remoteFault(5) indicates that 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.
                The values remoteJabber(7), remoteLinkLoss(8), and
                remoteTest(9) should be used instead of
                remoteFault(5) where the reason for remote fault
                is identified in the remote signaling protocol.

                The value invalidSignal(6) indicates that an
                invalid signal has been received from the other
                end of the link.  InvalidSignal(6) applies only to
                MAUs of type 10BASE-FB.

                Where an IEEE Std 802.3u-1995 clause 22 MII is
                present, a logic one in the remote fault bit
                (reference section 22.2.4.2.8 of that document)
                maps to the value remoteFault(5), and a logic zero
                in the link status bit (reference section
                22.2.4.2.10 of that document) maps to the value
                notAvailable(4).  The value notAvailable(4) takes
                precedence over the value remoteFault(5)."
        REFERENCE
                "[IEEE 802.3 Mgt], 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)."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.5, aLoseMediaCounter."
        ::= { 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
                "[IEEE 802.3 Mgt], 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).  For MAUs of type
                dot3MauTypeAUI, dot3MauType100BaseT4,
                dot3MauType100BaseTX, and dot3MauType100BaseFX,
                this counter will always indicate zero."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.6,
                aJabber.jabberCounter."
```

```
        ::= { 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 links.  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 until the next
                carrier event.

                This counter increments only for MAUs of type
                dot3MauType100BaseT4, dot3MauType100BaseTX, and
                dot3MauType100BaseFX.  For all other MAU types,
                this counter will always indicate zero.

                The approximate minimum time for rollover of this
                counter is 7.4 hours."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.5.1.1.10, aFalseCarriers."
        ::= { ifMauEntry 9 }

    ifMauTypeList OBJECT-TYPE
        SYNTAX     Integer32
        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.
                The value is a sum which 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 which has the capability
                to be only 10BASE-T would have a value of 512
                (2**9).  In contrast, a MAU which 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:

                Power  Capability
                  0      other or unknown
                  1      AUI
```

```
                 2      10BASE-5
                 3      FOIRL
                 4      10BASE-2
                 5      10BASE-T duplex mode unknown
                 6      10BASE-FP
                 7      10BASE-FB
                 8      10BASE-FL duplex mode unknown
                 9      10BROAD36
                10      10BASE-T  half duplex mode
                11      10BASE-T  full duplex mode
                12      10BASE-FL half duplex mode
                13      10BASE-FL full duplex mode
                14      100BASE-T4
                15      100BASE-TX half duplex mode
                16      100BASE-TX full duplex mode
                17      100BASE-FX half duplex mode
                18      100BASE-FX full duplex mode
                19      100BASE-T2 half duplex mode
                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     OBJECT IDENTIFIER
      MAX-ACCESS read-write
      STATUS     current
      DESCRIPTION
              "This object identifies the default administrative
              10 or 100 Mb/s 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
                "[IEEE 802.3 Mgt], 30.5.1.1.1, aMAUID, and [IEEE
                802.3 Std], 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 }



    -- The ifJackTable applies to MAUs attached to interfaces
    -- which have one or more external jacks (connectors).

    ifJackTable OBJECT-TYPE
        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 OBJECT-TYPE
        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
                JackType
        }


    ifJackIndex 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."
        ::= { ifJackEntry 1 }

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


    -- 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.

    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."
        ::= { 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

        }


    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 which 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
                "[IEEE 802.3 Mgt], 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
                "[IEEE 802.3 Mgt], 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 [IEEE
                802.3 Std]."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.6.1.1.4,
                aAutoNegAutoConfig."
        ::= { ifMauAutoNegEntry 4 }

    ifMauAutoNegCapability OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "A value that uniquely identifies the set of
                capabilities of the local auto-negotiation entity.
                The value is a sum which 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 which
                has the capability to support only 100Base-TX half
                duplex would have a value of 32768 (2**15).  In
                contrast, an interface which supports both
                100Base-TX half duplex and 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:

                Power   Capability
                  0       other or unknown
                 (1-9)    (reserved)
                 10       10BASE-T  half duplex mode
                 11       10BASE-T  full duplex mode
                 12       (reserved)
                 13       (reserved)
                 14       100BASE-T4
                 15       100BASE-TX half duplex mode
                 16       100BASE-TX full duplex mode
                 17       (reserved)
                 18       (reserved)
                 19      100BASE-T2 half duplex mode
                 20      100BASE-T2 full duplex mode

                Note that interfaces that support this MIB may
```

```
                have capabilities that extend beyond the scope of
                this MIB."
        REFERENCE
                "[IEEE 802.3 Mgt], 30.6.1.1.5,
                aAutoNegLocalTechnologyAbility."
        ::= { ifMauAutoNegEntry 5 }

    ifMauAutoNegCapAdvertised OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-write
        STATUS     current
        DESCRIPTION
                "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
                "[IEEE 802.3 Mgt], 30.6.1.1.6,
                aAutoNegAdvertisedTechnologyAbility."
        ::= { ifMauAutoNegEntry 6 }

    ifMauAutoNegCapReceived OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "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 which
                have capabilities beyond the scope of this MIB."
        REFERENCE
                "[IEEE 802.3 Mgt], 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
                "[IEEE 802.3 Mgt], 30.6.1.2.1,
                acAutoNegRestartAutoConfig."
        ::= { ifMauAutoNegEntry 8 }


    --
    -- The Basic Broadband MAU Table
    --

    broadMauBasicTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF BroadMauBasicEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "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     current
        DESCRIPTION
                "An entry in the table, containing information
                about a single broadband MAU."
        INDEX      { broadMauIfIndex, broadMauIndex }
        ::= { broadMauBasicTable 1 }

    BroadMauBasicEntry ::=
        SEQUENCE {
            broadMauIfIndex
                Integer32,
            broadMauIndex
                Integer32,
            broadMauXmtRcvSplitType
                INTEGER,
```

```
            broadMauXmtCarrierFreq
                Integer32,
            broadMauTranslationFreq
                Integer32
        }

    broadMauIfIndex OBJECT-TYPE
        SYNTAX     Integer32 (1..2147483647)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable uniquely identifies the interface
                to which the MAU described by this entry is
                connected."
        REFERENCE
                "Reference RFC 1213, ifIndex."
        ::= { broadMauBasicEntry 1 }

    broadMauIndex OBJECT-TYPE
        SYNTAX     Integer32 (1..2147483647)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable uniquely identifies the MAU
                connected to interface broadMauIfIndex that is
                described by this entry."
        REFERENCE
                "Reference IEEE 802.3 MAU Mgt, 20.2.3.2, aMAUID."
        ::= { broadMauBasicEntry 2 }

    broadMauXmtRcvSplitType OBJECT-TYPE
        SYNTAX     INTEGER {
                       other(1),
                       single(2),
                       dual(3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "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
                "Reference IEEE 802.3 MAU Mgt, 20.2.3.2,
                aBbMAUXmitRcvSplitType."
        ::= { broadMauBasicEntry 3 }

    broadMauXmtCarrierFreq OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable indicates the transmit carrier
                frequency of the 10BROAD36 MAU in MHz/4; that is,
                in units of 250 kHz."
        REFERENCE
                "Reference IEEE 802.3 MAU Mgt, 20.2.3.2,
                aBroadbandFrequencies.xmitCarrierFrequency."
        ::= { broadMauBasicEntry 4 }

    broadMauTranslationFreq OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "This variable indicates the translation offset
                frequency of the 10BROAD36 MAU in MHz/4; that is,
                in units of 250 kHz."
        REFERENCE
                "Reference IEEE 802.3 MAU Mgt, 20.2.3.2,
                aBroadbandFrequencies.translationFrequency."
        ::= { broadMauBasicEntry 5 }



    -- Notifications for use by 802.3 MAUs

    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
                "[IEEE 802.3 Mgt], 30.5.1.3.1, nJabber
```

```
                notification."
        ::= { snmpDot3MauMgt 0 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
                "[IEEE 802.3 Mgt], 30.5.1.3.1, nJabber
                notification."
        ::= { snmpDot3MauMgt 0 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 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."
        ::= { mauModObjGrps 4 }

    mauIfGrp100Mbs OBJECT-GROUP
        OBJECTS     { ifMauFalseCarriers,
                      ifMauTypeList,
                      ifMauDefaultType,
                      ifMauAutoNegSupported }
        STATUS      current
        DESCRIPTION
            "Conformance group for MAUs attached
            to interfaces with 100 Mb/s capability."
        ::= { mauModObjGrps 5 }

    mauIfGrpJack OBJECT-GROUP
```

```
        OBJECTS     { ifJackType }
        STATUS      current
        DESCRIPTION
            "Conformance group for MAUs attached
            to interfaces with managed jacks."
        ::= { mauModObjGrps 6 }

    mauIfGrpAutoNeg OBJECT-GROUP
        OBJECTS     { ifMauAutoNegAdminStatus,
                      ifMauAutoNegRemoteSignaling,
                      ifMauAutoNegConfig,
                      ifMauAutoNegCapability,
                      ifMauAutoNegCapAdvertised,
                      ifMauAutoNegCapReceived,
                      ifMauAutoNegRestart }
        STATUS      current
        DESCRIPTION
            "Conformance group for MAUs attached to
            interfaces with managed auto-negotiation."
        ::= { mauModObjGrps 7 }

    mauBroadBasic OBJECT-GROUP
        OBJECTS     { broadMauIfIndex,
                      broadMauIndex,
                      broadMauXmtRcvSplitType,
                      broadMauXmtCarrierFreq,
                      broadMauTranslationFreq }
        STATUS      current
        DESCRIPTION
            "Conformance group for broadband MAUs
            attached to interfaces.  This group
            provides a conformance specification
            for RFC 1515 implementations."
        ::= { mauModObjGrps 8 }

    -- 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      current
        DESCRIPTION
            "Compliance for MAUs attached to repeater ports."

        MODULE -- this module
            MANDATORY-GROUPS { mauRpGrpBasic }

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

            GROUP mauRpGrpJack
            DESCRIPTION
                "Implementation of this optional group is
                recommended for MAUs which 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      current
        DESCRIPTION
            "Compliance for MAUs attached to interfaces."

        MODULE -- this module
            MANDATORY-GROUPS { mauIfGrpBasic }

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

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

            GROUP mauIfGrpAutoNeg
            DESCRIPTION
                "Implementation of this group is
                mandatory for MAUs which 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 }

    END
```

4. Acknowledgements

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

Chuck Black
John Flick
Jeff Johnson
Leon Leong
Mike Lui
Dave Perkins
Geoff Thompson
Maurice Turcotte
Paul Woodruff

Special thanks as well to Dave Perkins for his excellent work on the SMICng compiler, which made it easy to take advantage of the latest SNMPv2 constructs in this MIB.

5. References

[1] IEEE 802.3/ISO 8802-3 Information processing systems - Local area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 1993.

[2] IEEE 802.3u-1995, "MAC Parameters, Physical Layer, Medium Attachment Units and Repeater for 100 Mb/s Operation, Type 100BASE-T," Sections 21 through 29, Supplement to IEEE Std 802.3, October 26, 1995.

[3] IEEE 802.3u-1995, "10 & 100 Mb/s Management," Section 30, Supplement to IEEE Std 802.3, October 26, 1995.

[4] de Graaf, K., D. Romascanu, D. McMaster and K. McCloghrie, "Definitions of Managed Objects for IEEE 802.3 Repeater Devices using SMIv2", RFC 2108, February 1997.

[5] McCloghrie, K. and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.

[6] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[7] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.

[8] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.

[9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[10] Case, J., M. Fedor, M. Schoffstall and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[11] McMaster, D., K. McCloghrie and S. Roberts, "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 1515, September 1993.

6. Security Considerations

Certain management information defined in this MIB may be considered sensitive in some network environments. Therefore, authentication of received SNMP requests and controlled access to management information should be employed in such environments. The method for this authentication is a function of the SNMP Administrative Framework, and has not been expanded by this MIB.

Several objects in this MIB allow write access. 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, or modifying the capabilities that a MAU advertizes during autonegotiation. It is recommended that implementers seriously consider whether set operations should be allowed without providing, at a minimum, authentication of request origin.

7. Authors' Addresses

Kathryn de Graaf
3Com Corporation
118 Turnpike Rd.
Southborough, MA 01772 USA

Phone: (508)229-1627
Fax: (508)490-5882
EMail: kdegraaf@isd.3com.com

Dan Romascanu
Madge Networks (Israel) Ltd.
Atidim Technology Park, Bldg. 3
Tel Aviv 61131, Israel

Phone: 972-3-6458414, 6458458
Fax: 972-3-6487146
EMail: dromasca@madge.com

Donna McMaster
Cisco Systems Inc.
170 West Tasman Drive
San Jose, CA 95134

Phone:: (408) 526-5260
EMail: mcmaster@cisco.com

Keith McCloghrie
Cisco Systems Inc.
170 West Tasman Drive
San Jose, CA 95134

Phone: (408) 526-5260
EMail: kzm@cisco.com

Sam Roberts
Farallon Computing, Inc.
2470 Mariner Square Loop
Alameda, CA 94501-1010

Phone:: (510) 814-5215
EMail: sroberts@farallon.com

8. Full Copyright Statement

Copyright (C) The Internet Society (1997). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.