csmacdmgt(30) package(4)
port100 MbpsMonitor(12)};
PRESENT IF The 100 Mb/s Monitor capability is implemented;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) managedObjectClass(3) repeaterPortObjectClass(5)};

nbPortName NAME BINDING

SUBORDINATE OBJECT CLASS oRepeaterPort;
NAMED BY SUPERIOR OBJECT CLASS
oGroup AND SUBCLASSES;
WITH ATTRIBUTE aPortID;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) nameBinding(6) portName(8)};

30A.5.2 Port attributes

aPortID ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
BEHAVIOUR bPortID;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) portID(49)};

bPortID BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.1;

aPortAdminState ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
PortAdminState;
MATCHES FOR EQUALITY;
BEHAVIOUR bPortAdminState;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) portAdminState(50)};

bPortAdminState BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.2;

aAutoPartitionState ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoPartitionState;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoPartition;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) autoPartitionState(51)};

bAutoPartition BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.3;

aReadableFrames ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableFrames;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) readableFrames(52)};

bReadableFrames BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aReadableOctets ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableOctets;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) readableOctets(53)};

bReadableOctets BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.5;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.;

aFrameCheckSequenceErrors ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFCSErrors;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(54)};

bFCSErrors BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.6;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aAlignmentErrors ATTRIBUTE

DERIVED FROM	aCMCounter;
BEHAVIOUR	bAlignmentErrors;
REGISTERED AS	{iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) alignmentErrors(55)};

bAlignmentErrors BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.7;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

aFramesTooLong ATTRIBUTE

DERIVED FROM	aCMCounter;
BEHAVIOUR	bFramesTooLong;
REGISTERED AS	{iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) framesTooLong(56)};

bFramesTooLong BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.8;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;

aShortEvents ATTRIBUTE

DERIVED FROM	aCMCounter;
BEHAVIOUR	bShortEvents;
REGISTERED AS	{iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) shortEvents(57)};

bShortEvents BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.9;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 hours;

aRunts ATTRIBUTE

DERIVED FROM	aCMCounter;
BEHAVIOUR	bRunts;
REGISTERED AS	{iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) runts(58)};

bRunts BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.10;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

aCollisions ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bCollisions;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) collisions(59)};

bCollisions BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.11;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

aLateEvents ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bLateEvents;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) lateEvents(60)};

bLateEvents BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.12;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 81 h.;

aVeryLongEvents ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bVeryLongEvents;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) veryLongEvents(61)};

bVeryLongEvents BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.13;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 198 days.;

aDataRateMismatches ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bDataRateMismatches;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) dataRateMismatches(62)};

bDataRateMismatches BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.14;

aAutoPartitions ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bAutoPartitions;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) autoPartitions(63)};

bAutoPartitions BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.15;

aIsolates ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bIsolates;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) isolates(64)};

bIsolates BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.16;

aSymbolErrorDuringPacket ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bSymbolErrorDuringPacket;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) symbolErrorDuringPacket(65)};

bSymbolErrorDuringPacket BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.17;

aLastSourceAddress ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802CommonDefinitions.MACAddress;
MATCHES FOR EQUALITY;

BEHAVIOUR bLastSourceAddress;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) lastSourceAddress(66)};

bLastSourceAddress BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.18;

aSourceAddressChanges ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bSourceAddressChanges;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) sourceAddressChanges(67)};

bSourceAddressChanges BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.19;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 81 h.;

30A.5.3 Port actions

acPortAdminControl ACTION

BEHAVIOUR bPortAdminControl;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) portAdminControl(8)};

bPortAdminControl BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.2.1;

30A.6 MAU managed object class

30A.6.1 MAU, formal definition

oMAU MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY
pMAUBasic PACKAGE
ATTRIBUTES aMAUID GET,
aMAUType GET-SET,

```
                                        aMAUTypeList            GET,
                                        aMediaAvailable         GET,
                                        aJabber                 GET,
                                        aMAUAdminState          GET;
                    NOTIFICATIONS       nJabber;
                ;
        ;
        CONDITIONAL PACKAGES
                pMAUControl             PACKAGE
                    ACTIONS             acResetMAU,
                                        acMAUAdminControl;
                    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mauControlPkg(13)};
                    PRESENT IF          The pMAUControl package is implemented.;

                pMediaLossTracking      PACKAGE
                    ATTRIBUTES          aLoseMediaCounter       GET;
                    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mediaLossTrackingPkg(14)};
                    PRESENT IF          MAU TypeValue = AUI or if the
                                        pMediaLossTracking package is implemented.;

                pBroadbandDTEMAU        PACKAGE
                    ATTRIBUTES          aBbMAUXmitRcvSplitType  GET,
                                        aBroadbandFrequencies   GET;
                    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) broadbandMAUPkg(15)};
                    PRESENT IF          The MAU is of type 10BROAD36.;

                p100MbpsMonitor         PACKAGE
                    ATTRIBUTES          aFalseCarriers          GET;
                    REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mau100MbpsMonitor(16)};
                    PRESENT IF          The MAU is capable of 100 Mb/s operation.;
        REGISTERED AS                   {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        managedObjectClass(3) mauObjectClass(6)};
nbMAU-repeaterName                      NAME BINDING

        SUBORDINATE OBJECT CLASS        oMAU;
        NAMED BY SUPERIOR OBJECT CLASS                  --(of oRepeaterPort)
                                        oRepeaterPort   AND SUBCLASSES;
                                                        --{1.2.840.10006.30.3.5}
        WITH ATTRIBUTE                  aMAUID;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                            mau-repeaterName(9)};

nbMAU-dteName                           NAME BINDING

        SUBORDINATE OBJECT CLASS        oMAU;
        NAMED BY SUPERIOR OBJECT CLASS                  --(of oPHYEntity)
                                        oPHYEntity  AND SUBCLASSES
                                                        --{1.2.840.10006.30.3.2};
        WITH ATTRIBUTE                  aMAUID;
        REGISTERED AS       {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
```

mau-dteName(10)};

30A.6.2 MAU attributes

aMAUID ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bMAUID;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauID(68)};

bMAUID BEHAVIOUR

DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.1;

aMAUType ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.TypeValue;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMAUType;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauType(69)};

bMAUType BEHAVIOUR

DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.2;

aMAUTypeList ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.TypeList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMAUTypeList;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauTypeList(70)};

bMAUTypeList BEHAVIOUR

DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.3;

aMediaAvailable ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. MediaAvailState;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMediaAvailable;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauMediaAvailable(71)};

bMediaAvailable BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.4;

aLoseMediaCounter ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLoseMediaCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauLoseMediaCounter(72)};

bLoseMediaCounter BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.5;

aJabber ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Jabber;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJabberAttribute;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) jabber(73)};

bJabberAttribute BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.6;

aMAUAdminState ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMAUAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauAdminState(74)};

bMAUAdminState BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.7;

aBbMAUXmitRcvSplitType ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
BbandXmitRcvSplitType;
MATCHES FOR EQUALITY;
BEHAVIOUR bBbMAUXmitRcvSplitType;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandSplitType(75)};

bBbMAUXmitRcvSplitType BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.8;

aBroadbandFrequencies ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BbandFrequency;
MATCHES FOR EQUALITY;
BEHAVIOUR bBroadbandFrequencies;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandFrequencies(76)};

bBroadbandFrequencies BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.9;

aFalseCarriers ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bFalseCarriers;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) falseCarriers(77)};

bFalseCarriers BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.10;

30A.6.3 MAU actions

acResetMAU ACTION

BEHAVIOUR bResetMAU;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) resetMAU(9)};

bResetMAU BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.2.1;

acMAUAdminControl ACTION

BEHAVIOUR bMAUAdminControl;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) mauAdminCtrl(10)};

bMAUAdminControl BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.2.2;

30A.6.4 MAU notifications

nJabber NOTIFICATION

BEHAVIOUR bJabberNotification;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.Jabber;
;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10) jabber(5)};

bJabberNotification BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.3.1;

30A.7 AutoNegotiation managed object class

30A.7.1 AutoNegotiation, formal definition

oAutoNegotiation MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

pAutoNeg PACKAGE

ATTRIBUTES

aAutoNegID	GET,
aAutoNegAdminState	GET,
aAutoNegRemoteSignaling	GET,
aAutoNegAutoConfig	GET-SET,
aAutoNegLocalTechnologyAbility	GET,
aAutoNegAdvertisedTechnologyAbility	GET-SET,
aAutoNegReceivedTechnologyAbility	GET,
aAutoNegLocalSelectorAbility	GET,
aAutoNegAdvertisedSelectorAbility	GET-SET,
aAutoNegReceivedSelectorAbility	GET;

ACTIONS

acAutoNegRestartAutoConfig,
acAutoNegAdminControl;

;
;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)

managedObjectClass(3) autoNegObjectClass(7)};

nbAutoNeg-mauName NAME BINDING

SUBORDINATE OBJECT CLASS oMAU;
NAMED BY SUPERIOR OBJECT CLASS --(of oMAU)
oMAU AND SUBCLASSES;
--{1.2.840.10006.30.3.6}
WITH ATTRIBUTE aMAUID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6) autoNeg-mauName(11)};

30A.7.2 Auto-Negotiation attributes

aAutoNegID ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegID;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegID(78)};

bAutoNegID BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.1;

aAutoNegAdminState ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAdminState;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAdminState(79)};

bAutoNegAdminState BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.2;

aAutoNegRemoteSignaling ATTRIBUTE

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegRemoteSignalingDetect;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegRemoteSignaling;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegRemoteSignaling(80)};

bAutoNegRemoteSignaling BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.3;

aAutoNegAutoConfig ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegAutoConfig;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bAutoNegAutoConfig;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAutoConfig(81)};

bAutoNegAutoConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.4;

aAutoNegLocalTechnologyAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegLocalTechnologyAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegLocalTechnologyAbility(82)};

bAutoNegLocalTechnologyAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.5;

aAutoNegAdvertisedTechnologyAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegAdvertisedTechnologyAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAdvertisedTechnologyAbility(83)};

bAutoNegAdvertisedTechnologyAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.6;

aAutoNegReceivedTechnologyAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegReceivedTechnologyAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)

autoNegReceivedTechnologyAbility(84)};

bAutoNegReceivedTechnologyAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.7;

aAutoNegLocalSelectorAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegSelectorList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegLocalSelectorAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegLocalSelectorAbility(85)};

bAutoNegLocalSelectorAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.8;

aAutoNegAdvertisedSelectorAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegSelectorList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegAdvertisedSelectorAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAdvertisedSelectorAbility(86)};

bAutoNegAdvertisedSelectorAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.9;

aAutoNegReceivedSelectorAbility ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegSelectorList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAutoNegReceivedSelectorAbility;
REGISTERED AS	{iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegReceivedSelectorAbility(87)};

bAutoNegReceivedSelectorAbility BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.1.10;

30A.7.3 AutoNegotiation actions

acAutoNegRestartAutoConfig ACTION

BEHAVIOUR bAutoNegRestartAutoConfig;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) autoNegRestartAutoConfig(11)};

bAutoNegRestartAutoConfig BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.2.1;

acAutoNegAdminControl ACTION

BEHAVIOUR bAutoNegAdminControl;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.AutoNegAdminState;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) autoNegAdminCtrl(12)};

bAutoNegAdminControl BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.2.2;

30A.8 ResourceTypeID managed object class

30A.8.1 ResourceTypeID, formal definition

— Implementation of this managed object in accordance with the definition contained in IEEE Std
— 802.1F-1993 is a conformance requirement of this standard.
— NOTE—A single instance of the Resource Type ID managed object exists within the oMACEntity
— managed object class, a single instance of the Resource Type ID managed object exists within
— the oRepeater managed object class, and a single instance of the Resource Type ID managed
— object exists within the oMAU managed object class conditional on the presence of an MII.
— The managed object itself is contained in IEEE Std 802.1F-1993, therefore only name bindings
— appear in this standard;

nbResourceTypeID-mac NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":oResourceTypeID;
NAMED BY SUPERIOR OBJECT CLASS oMACEntity;
WITH ATTRIBUTE "IEEE802.1F":aResourceTypeIDName;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) nameBinding(6) resourceTypeID-mac(12)};

nbResourceTypeID-repeater NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":oResourceTypeID;
NAMED BY SUPERIOR OBJECT CLASS
oRepeater AND SUBCLASSES;
WITH ATTRIBUTE "IEEE802.1F":aResourceTypeIDName;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) nameBinding(6) resourceTypeID-repeater(13)};

nbResourceTypeID-mau NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":oResourceTypeID;
NAMED BY SUPERIOR OBJECT CLASS
oMAU AND SUBCLASSES;
WITH ATTRIBUTE "IEEE802.1F":aResourceTypeIDName;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) nameBinding(6) resourceTypeID-mau(14)};

Annex 30B

(normative)

GDMO and ASN.1 definitions for management

30B.1 Common attributes template

aCMCounter ATTRIBUTE

DERIVED FROM	"ISO/IEC 10165-5":genericWrappingCounter;
BEHAVIOUR	bCMCounter;
REGISTERED AS	{iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) cmCounter(88)};

bCMCounter BEHAVIOUR

DEFINED AS	Wraps at one of two sizes. Size is conditional. Wraps at 32 bits, that is this counter reaches its maximum value at $2^{32}-1$ (i.e., approximately 4.294×10^{9}) and then rolls over to zero on the next increment, if maximum increment rate from zero causes a rollover in 58 min or more. Wraps at 64 bits, that is this counter reaches its maximum value at $2^{64}-1$ (i.e., approximately $1.844\ldots \times 10^{19}$) and then rolls over to zero on the next increment, if maximum increment rate from zero would cause a 32 bit counter to roll over in less than 58 min. The counter that this is derived from initializes to zero. Initialization to zero is not a requirement of this standard;

30B.2 ASN.1 module for CSMA/CD managed objects

This ASN.1 module defines the ASN.1 types and subtypes that are referred to immediately after the WITH ATTRIBUTE SYNTAX construct in this clause's uses of the attribute template defined in ISO/IEC 10165-4: 1992, Guidelines for the definition of managed objects (GDMO).

IEEE802Dot3-MgmtAttributeModule {iso(1) member-body(2) us(840) 802dot3(10006) global(1) asn1Module(2) commonDefinitions(0) version(2)} DEFINITIONS IMPLICIT TAGS::= BEGIN

EXPORTS--*everything*

IMPORTS--*implicitly imports ISO 8824: 1990*

```
MACAddress
    FROM IEEE802CommonDefinitions
    {iso(1) member-body(2) us(840) ieee802dot1partF(10011)
    asn1Module(2) commonDefinitions(0) version1(0)};


AdminState::= ENUMERATED {
    other                   (1),    --undefined
```

```
        unknown             (2),    --initializing, true state not yet known
        operational         (3),    --powered and connected
        standby             (4),    --inactive but on
        shutdown            (5)     --similar to power down
        }

AttemptArray::= SEQUENCE OF aCMCounter--array [1..attempt limit - 1]

AutoNegAdminState::= ENUMERATED {
        disabled            (1),
        enabled             (2)
        }

AutoNegAutoConfig:=ENUMERATED {
        other               (1),
        configuring         (2),
        complete            (3),
        disabled            (4),
        parallel detect fail (5)
        }

AutoNegRemoteSignalingDetect::=ENUMERATED {
        detected            (1),
        notdetected         (2)
        }

AutoNegSelector::=ENUMERATED {
        other               (1),    --undefined
        ethernet            (2),    --802.3
        isoethernet         (3)     --802.9
        }

AutoNegSelectorList::=SEQUENCE OF AutoNegSelector

AutoNegTechnology::=ENUMERATED {
        global              (0),    --reserved for future use.
        other               (1),    --undefined
        unknown             (2),    --initializing, true ability not yet known.
        10BASE-T            (14),   --10BASE-T as defined in clause 14
        100BASE-T4          (23),   --100BASE-T4 as defined in clause 23
        100BASE-TX          (25),   --100BASE-TX as defined in clause 25
        10BASE-TFD          (142),  --Full-duplex 10BASE-T
        100BASE-TXFD        (252),  --Full-duplex 100BASE-TX
        isoethernet         (8029)  --802.9 ISLAN-16T
        }
AutoNegTechnologyList::=SEQUENCE OF AutoNegTechnology
AutoPartitionState::= ENUMERATED {
        autoPartitioned     (1),
        notAutoPartitioned  (2)
        }

BbandFrequency::= SEQUENCE {
        xmitCarrierFrequency    [1]     INTEGER , --Frequency in MHz times 4 (250 kHz resolution)
        translationFrequency    [2]     INTEGER   --Frequency in MHz times 4 (250 kHz resolution)
        }
```

```
BbandXmitRcvSplitType::= ENUMERATED {
      other                 (1),      --undefined
      single                (2),      --single-cable system
      dual                  (3)       --dual-cable system, offset normally zero
      }

BitString ::= BIT STRING (SIZE (1..1024))

Jabber::= SEQUENCE {
      jabberFlag            [1]       JabberFlag,
      jabberCounter         [2]       JabberCounter
      }

JabberFlag::= ENUMERATED {
      other                 (1),      --undefined
      unknown               (2),      --initializing, true state not yet known
      normal                (3),      --state is true or normal
      fault                 (4)       --state is false, fault or abnormal
      }

JabberCounter::= INTEGER (0..2^32 –1)

MauTypeList::= SEQUENCE OF TypeValue

MediaAvailState::= ENUMERATED {
      other                 (1),      --undefined
      unknown               (2),      --initializing, true state not yet known
      available             (3),      --link or light normal, loopback normal
      not available         (4),      --link loss or low light, no loopback
      remote fault          (5),      --remote fault with no detail
      invalid signal        (6),      --invalid signal, applies only to 10BASE-FB
      remote jabber         (7),      --remote fault, reason known to be jabber
      remote link loss      (8),      --remote fault, reason known to be far-end link loss
      remote test           (9)       --remote fault, reason known to be test
      }

MIIDetect::= ENUMERATED {
      unknown                 (1),
      presentNothingConnected (2),
      presentConnected        (3),
      absent                  (4)
      }

MulticastAddressList::= SEQUENCE OF MACAddress

OneOfName ::= INTEGER (1..1024)

PhyTypeList::= SEQUENCE OF PhyTypeValue

PhyTypeValue::= ENUMERATED {
      other                 (1),      --undefined:
      unknown               (2),      --initializing, true state or type not yet known
```

```
        none                (3),    --MII present and nothing connected
        10 Mb/s             (7),    --clause 7 10 Mb/s Manchester
        100BASE-T4          (23),   --clause 23 100 Mb/s 8B/6T
        100BASE-X           (24)    --clause 24 100 Mb/s 4B/5B
        }

PortAdminState::= ENUMERATED {
        disabled            (1),
        enabled             (2)
        }

RepeaterHealthData ::= OCTET STRING (SIZE (0..255))

RepeaterHealthInfo::= SEQUENCE {
        repeaterHealthState     [1]     RepeaterHealthState,
        repeaterHealthText      [2]     RepeaterHealthText  OPTIONAL,
        repeaterHealthData      [3]     RepeaterHealthData  OPTIONAL
        }

RepeaterHealthState::= ENUMERATED {
        other               (1),    --undefined or unknown
        ok                  (2),    --no known failures
        repeaterFailure     (3),    --known to have a repeater-related failure
        groupFailure        (4),    --known to have a group-related failure
        portFailure         (5),    --known to have a port-related failure
        generalFailure      (6)     --has a failure condition, unspecified type
        }

RepeaterType::= ENUMERATED {
        other               (1),    --See 20.2.2.3:
        unknown             (2),    --initializing, true state or type not yet known
        10 Mb/s             (9),    --clause 9 10 Mb/s Baseband repeater
        100 Mb/sClassI      (271),  --clause 27 class I 100 Mb/s Baseband repeater
        100 Mb/sClassII     (272),  --clause 27 class II 100 Mb/s Baseband repeater
        802.9a              (99)    --Integrated services repeater
        }

RepeaterHealthText ::= PrintableString (SIZE (0..255))

TrueFalse ::= BOOLEAN

TypeList::= SEQUENCE OF TypeValue

TypeValue::= ENUMERATED {
        global              (0),    --undefined
        other               (1),    --undefined
        unknown             (2),    --initializing, true state not yet known
        AUI                 (7),    --no internal MAU, view from AUI
        10BASE5             (8),    --Thick coax MAU as specified in clause 8
        FOIRL               (9),    --FOIRL MAU as specified in 9.9
        10BAS               (10),   --Thin coax MAU as specified in clause 10
        10BROAD36           (11),   --Broadband DTE MAU as specified in clause 11
        10BASE-T            (14),   --UTP MAU as specified in clause 14
        10BASE-FP           (16),   --Passive fiber MAU, specified in clause 16
        10BASE-FB           (17),   --Synchronous fiber MAU, specified in clause 17
```

```
        10BASE-FL       (18),   --Asynchronous fiber  MAU, specified in clause 18
        100BASE-T4      (23),   --Four-pair Category 3 UTP as specified in clause 23
        100BASE-TX      (25),   --Two-pair Category 5 UTP as specified in clause 25
        100BASE-FX      (26),   --X fiber over PMD as specified in clause 26
        802.9a          (99)    --Integrated services MAU as specified in IEEE Std 802.9
                                  ISLAN-16T
        }

    END
```