            OSI Internet Management: Management Information Base

Status of this Memo

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

Abstract

   This RFC documents a MIB for use with CMIP, either over pure OSI
   stacks or with the CMIP over TCP specification. It redefines objects
   comprised by the second revision of the Management Information Base
   for Network Management of TCP/IP-based internets: MIB-II so as to
   conform to the OSI structure of management information. This
   document is a product of the IETF OIM working group.

Table of Contents

1. Introduction

        This memo defines the management information base (MIB) for use with
        the OSI network management protocol in TCP/IP based internets. It
        formats the Management Information Base (MIB-II) in OSI templates and
        adds variables necessary for use with the OSI management protocol.
        Together with internet memos that define agreements for using the OSI
        management protocol to manage TCP/IP internets (RFC 1189) [3], the
        management information base: MIB-II, and OSI standards defining the
        structure of management information (ISO/IEC DIS 10165) [4,5,6],
        these documents provide an OSI compatible architecture and system for
        managing TCP/IP based internets and in particular the Internet
        community.

        This OSI version of the MIB specification is an incremental
        refinement of MIB-II. It has been designed according to the
        following policy: first, maintain compatibility with MIB-II by
        including objects in MIB-II without changes to their registered
        identifiers, syntax, and behaviour;   second, format the MIB-II using
        the OSI structure of information defined templates; third, add
        attributes necessary for identification using the OSI management
        protocol; fourth, identify object instances in a manner which is
        compatible with development of a common application programming
        interface for both CMIP and SNMP; and finally, define notifications
        (events), including those defined by SNMP (RFC 1157) [14], deemed
        necessary for management.

        It is expected that the MIB will grow over time to accommodate the
        changing needs of the Internet.

        As reported in RFC 1109, Report of the Second Ad Hoc Network
        Management Review Group [13], the requirements for SNMP and OSI
        frameworks were more different than anticipated. Therefore, the
        requirement for compatibility between the SMI/MIB for both frameworks
        was suspended. This action allowed both frameworks to independently
        define the MIB according to their SMI requirements and views of
        management.   The SNMP response included development of MIB-II. The
        OSI response is this document which builds on the excellent work done
        in MIB-II, and adds to it attributes necessary to map into the OSI

SMI.

The OIM network management framework consists of: Open Systems
Interconnection - Structure of Management Information: Parts 1-4,
ISO/IEC DIS 10165 [4,5,6], with slight modifications to maintain
compatibility with RFC 1155 [11];   OSI Internet Management:
Management Information Base, this memo; and the OSI Common Management
Information Service (CMIS) [7] and Common Management Information
Protocol (CMIP) [8].

2. Additions to MIB-II

Additions to MIB-II fall into the following categories: the addition
of the managed object class "top" from which all object classes are
derived; additions for object instance identification purposes
according to the OSI SMI;   additions for generation of asynchronous
events; the addition of name bindings which define the binding
between objects in the containment (naming) hierarchy; and the
addition of proposed changes to RFC 1158 that are expected to be
accepted by the IETF [16].

2.1 Structure of the OIM Subtree

The oim subtree structure is defined below. Note that the "cmot"
subtree has been relabeled to "oim".

```
cmot            OBJECT IDENTIFIER ::= {mib 9}
oim             OBJECT IDENTIFIER ::= {cmot}
cmotVersion     OBJECT IDENTIFIER ::= {oim 1}
cmotACSEInfo    OBJECT IDENTIFIER ::= {oim 2}
cmotSystemId    OBJECT IDENTIFIER ::= {oim 3}
misc            OBJECT IDENTIFIER ::= {oim 4}
objects         OBJECT IDENTIFIER ::= {oim 5}
attributes      OBJECT IDENTIFIER ::= {oim 6}
events          OBJECT IDENTIFIER ::= {oim 7}
nameforms       OBJECT IDENTIFIER ::= {oim 8}
actions         OBJECT IDENTIFIER ::= {oim 9}
```

2.2 Supporting ISO Definitions

The OSI defined managed object class "top" [5], as defined by OSI,
has been added to the MIB. This managed object class is the root of
the inheritance hierarchy. The class "top" contains attributes that
are inherited by all objects in the MIB. Pending the progression of
ISO/IEC DIS 10165 to an International Standard status, "top" and its
associated attributes are registered in Appendix 3.

2.3 Additions for Object Instance Identification

   The conventions for identifying objects in the OIM MIB-II for use
   with CMIP are defined below.

2.3.1 Distinguished Names

   The distinguished name of an object shall consist of a sequence of
   relative distinguished names, one for each object on the containment
   path from the root to the object. Each relative distinguished name
   shall contain exactly one attribute, the "naming" attribute for the
   corresponding class, as specified by a NAME BINDING template. The
   CMIP ObjectInstance shall be encoded using the distinguishedName
   choice.

   Object classes for which there is only a single instance normally
   implemented in a managed system, other than the "system" managed
   object class, shall use an empty string for their naming attribute
   value, i.e., a string of length zero.

   The format of the naming attribute for the "system" managed object
   class and object classes which normally have multiple instances,
   e.g., table entries, are defined in the attribute templates for the
   naming attributes of those classes. The formats for naming
   attributes of table entries are compatible with instance
   identification conventions used by SNMP, thereby facilitating the
   development of a common application programming interface that may
   interface to either protocol.

   For example, a distinguished name designating a particular routing
   table entry (of class ipRouteEntry) might be

        {
                { sysName = "troi.mitre.org"}
                { ipId = ""}
                { ipRoutingTableId = "" }
                { ipRouteEntryId = "129.83.2.17" }
        }.

   The naming attributes for each OIM MIB-II managed object class listed
   in section 5 may be determined by reference to the NAME BINDING
   templates listed in section 8. The naming attribute definitions can
   be found in the alphabetical listing of all attributes in section 6.

   The following attributes have been defined for MIB-II object classes
   for purposes of object instance identification.

```
           Attribute              Identifier              Object

     ifTableId               {attributes 1}          ifTable
     atTableId               {attributes 2}          atTable
     atEntryId               {attributes 3}          atEntry
     ipId                    {attributes 4}          ip
     ipAddrTableId           {attributes 5}          ipAddrTable
     ipRoutingTableId        {attributes 6}          ipRoutingtable
     ipNetToMediaTableId     {attributes 7}          ipNetToMediaTable
     ipNetToMediaEntryId     (attributes 8}          ipNetToMediaEntry
     icmpId                  {attributes 9}          icmp
     tcpId                   {attributes 10}         tcp
     tcpConnTableId          {attributes 11}         tcpConnTable
     tcpConnId               {attributes 12}         tcpConnEntry
     udpId                   {attributes 13}         udp
     udpTable
     udpEntryId              {attributes 15}         udpEntry
     egpId                   {attributes 16}         egp
     egpNeighTableId         {attributes 17}         egpneighTable
     snmpId                  {attributes 18}         snmp
     ipAdEntryId             {attributes 19}         ipAddrEntry
     ipRouteEntryId          {attributes 20}         ipRouteEntry
     ifEntryId               {attributes 21}         ifEntry
     egpNeighEntryId         {attributes 22}         egpNeighEntry
     ifId                    {attributes 23}         interfaces
```

2.4 Events

   Events have been defined in accordance with the OSI SMI. These
   events include those defined for SNMP. The OSI SMI requires that
   events (notifications) are part of a specific managed object class
   definition. Accordingly the SNMP defined events have been assigned
   to specific objects within the MIB.

```
           Event                    Identifier            Object

     coldStartEvent              {events 0}            system
     warmStartEvent              {events 1}            system
     linkDownEvent               {events 2}            ifEntry
     linkUpEvent                 {events 3}            ifEntry
     snmpAuthentFailureEvent     {events 4}            snmp
     egpNeighborLossEvent        {events 5}            egpNeighEntry
     entSpecificEvent            {events 6}            system
```

2.5 Containment (Naming) Tree Definition

   The OSI SMI requires that object classes be bound into the
   containment hierarchy for purposes of naming. The binding must

specify for the managed object class: a) the object which is superior
to it in the containment hierarchy; and b) an attribute in the
managed object class that is used to distinguish instances of the
object at a given level in the containment hierarchy. The
containment tree begins with "root". Agents in managed systems will
consider root to be immediately above the "system" object in the
containment hierarchy. They may ignore any relative distinguished
names that precede the relative distinguished name for "system".

The name bindings for objects in the OIM MIB-II are specified in
section 8.

2.6 Changes from RFC 1158

The proposed successor to RFC 1158 [16] contains several minor
additions that have been incorporated into this document.

> o   The sysLocation attribute of the system managed object
>   class is changed to GET-REPLACE.

> o   The ipRouteMetric5 and ipRouteInfo attributes are
>   added to the ipRouteEntry managed object class.

> o   The textual descriptor of the snmpEnableAuthTraps is
>   changed to snmpEnableAuthenTraps.

> o   The PhysAddress textual convention is introduced to
>   represent media addresses.

> o   The definition of sysServices is clarified.

> o   New ifType values (29-32) are defined. In
>   addition, the textual-descriptor for the DS1 and E1
>   interface types is corrected.

> o   The definition of ipForwarding is clarified.

> o   The definition of ipRouteType is clarified.

> o   The tcpConnState attribute is now GET-REPLACE, to
>   support deletion of the TCB associated with a TCP
>   connection. The definition of this object is
>   clarified to explain this usage.

> o   The definition of egpNeighEventTrigger is clarified.

> o   The definition of several of the variables in the new
>   snmp group are clarified. In addition, the

snmpInBadTypes and snmpOutReadOnlys objects are no
longer present. (However, the object identifiers associated
with those objects are reserved to prevent future use.)

   o The definition of snmpInReadOnlys is clarified.

   o The ipRoutingDiscards attribute is added to the ip object.

   o The optional use of an implementation-dependent, small
      positive integer is disallowed when identifying
      instances of the IP address and routing tables.

3. Format of OSI Templates

   The format of OSI templates is defined in ISO/IEC DIS 10165-4 [6],
   Open Systems Interconnection - Structure of Management Information -
   Part 4: Guidelines for Managed Object Definition.

   The template definitions relevant to this document are included in
   Appendix 1.

4. Object Class Definitions

   The Internet SMI objects are recast into OSI template form using the
   following conventions:

     o MIB II groups become OSI object classes,
       except for the "at" group which has no attributes.

     o Tables become OSI object classes.

     o Table entries become OSI object classes.

     o All other object types become OSI attributes and are
       assigned to object classes according to their association
       in MIB II.

   The templates for the OIM MIB-II object classes are listed in
   alphabetical order below.

   atEntry MANAGED OBJECT CLASS
            DERIVED FROM top;
            CHARACTERIZED BY
               atEntryPkg PACKAGE
                    -- see MIB II [16] for attribute semantics
               ATTRIBUTES
                            atEntryId   GET,

```
                                atIfIndex            GET-REPLACE,
                                atPhysAddress        GET-REPLACE,
                                atNetAddress         GET-REPLACE;;;
                  REGISTERED AS  { atTable 1};

      atTable  MANAGED OBJECT CLASS
               DERIVED FROM  top;
               CHARACTERIZED BY
                  atTablePkg PACKAGE
                            -- see MIB II [16] for attribute semantics
                  ATTRIBUTES
                      atTableId                      GET;;;
                  REGISTERED AS  { at 1};

      egp MANAGED OBJECT CLASS
               DERIVED FROM  top;
               CHARACTERIZED BY
                  egpPkg PACKAGE
                      -- see MIB II [16] for attribute semantics
                  ATTRIBUTES
                                  egpId              GET,
                                  egpInMsgs          GET,
                                  egpInErrors        GET,
                                  egpOutMsgs         GET,
                                  egpOutErrors       GET,
                                egpAs                GET;;;
                  REGISTERED AS   { mib 8};

      egpNeighEntry MANAGED OBJECT CLASS
               DERIVED FROM  top;
               CHARACTERIZED BY
                  egpNeighEntryPkg PACKAGE
                      -- see MIB II [16] for attribute semantics
                  ATTRIBUTES
                                  egpNeighEntryId     GET,
                                  egpNeighState       GET,
                                  egpNeighAddr        GET,
                                  egpNeighAs          GET,
                                  egpNeighInMsgs      GET,
                                  egpNeighInErrs      GET,
                                  egpNeighOutMsgs     GET,
                                  egpNeighOutErrs     GET,
                                  egpNeighInErrMsgs   GET,
                                  egpNeighOutErrMsgs  GET,
                                  egpNeighStateUps    GET,
                                  egpNeighStateDowns  GET,
                                  egpNeighIntervalHello GET,
                                  egpNeighIntervalPoll  GET,
```

```
                                egpNeighMode            GET,
                                egpNeighEventTrigger    GET-REPLACE;
                 NOTIFICATIONS
                                egpNeighborLossEvent;;;
            REGISTERED AS   { egpNeighTable 1};

     egpNeighTable MANAGED OBJECT CLASS
            DERIVED FROM   top;
            CHARACTERIZED BY
               egpNeighTablePkg PACKAGE
                   -- see MIB II [16] for attribute semantics
               ATTRIBUTES
                                egpNeighTableId     GET;;;
            REGISTERED AS   { egp 5};

     icmp MANAGED OBJECT CLASS
            DERIVED FROM   top;
            CHARACTERIZED BY
               icmpPkg PACKAGE
                   -- see MIB II [16] for attribute semantics
               ATTRIBUTES
                                icmpId GET,
                                icmpInMsgs              GET,
                                icmpInErrors            GET,
                                icmpInDestUnreachs      GET,
                                icmpInTimeExcds         GET,
                                icmpInParmProbs         GET,
                                icmpInSrcQuenchs        GET,
                                icmpInRedirects         GET,
                                icmpInEchos             GET,
                                icmpInEchoReps          GET,
                                icmpInTimestamps        GET,
                                icmpInTimestampReps GET,
                                icmpInAddrMasks         GET,
                                icmpInAddrMaskReps      GET,
                                icmpOutMsgs             GET,
                                icmpOutErrors           GET,
                                icmpOutDestUnreachs GET,
                                icmpOutTimeExcds        GET,
                                icmpOutParmProbs        GET,
                                icmpOutSrcQuenchs       GET,
                                icmpOutRedirects        GET,
                                icmpOutEchos            GET,
                                icmpOutEchoReps         GET,
                                icmpOutTimestamps       GET,
                                icmpOutTimestampReps GET,
                                icmpOutAddrMasks        GET,
                                icmpOutAddrMaskReps GET;;;
```

```
                        REGISTERED AS  { mib 5};

     ifEntry   MANAGED OBJECT CLASS
                DERIVED FROM  top;
                CHARACTERIZED BY
                    ifEntryPkg PACKAGE
                        -- see MIB II [16] for attribute semantics
                    ATTRIBUTES
                                ifEntryId            GET,
                                    ifIndex          GET,
                                    ifDescr          GET,
                                    ifType           GET,
                                    ifMtu            GET,
                                    ifSpeed          GET,
                                    ifPhysAddress    GET,
                                    ifAdminStatus    GET-REPLACE,
                                    ifOperStatus     GET,
                                    ifLastChange     GET,
                                    ifInOctets       GET,
                                    ifInUcastPkts    GET,
                                    ifInNUcastPkts   GET,
                                    ifInDiscards     GET,
                                    ifInErrors       GET,
                                    ifInUnknownProtos GET,
                                    ifOutOctets      GET,
                                    ifOutUcastPkts   GET,
                                    ifOutNUcastPkts  GET,
                                    ifOutDiscards    GET,
                                    ifOutErrors      GET,
                                    ifOutQLen        GET,
                                    ifSpecific       GET;
                    NOTIFICATIONS
                                            linkUpEvent,
                                            linkDownEvent;;;
                REGISTERED AS  { ifTable 1};

     ifTable MANAGED OBJECT CLASS
                DERIVED FROM  top;
                CHARACTERIZED BY
                    ifTablePkg PACKAGE
                        -- see MIB II [16] for attribute semantics
                    ATTRIBUTES
                                ifTableId               GET;;;
                 REGISTERED AS  { interface 2};

     interfaces MANAGED OBJECT CLASS
                DERIVED FROM  top;
                CHARACTERIZED BY
```

```
                    interfacesPkg PACKAGE
                        -- see MIB II [16] for attribute semantics
                    ATTRIBUTES
                                    ifId                    GET,
                                    ifNumber                GET;;;
                    REGISTERED AS   { mib 2};

    ip MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipPkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                                    ipId                    GET,
                                    ipForwarding            GET-REPLACE,
                                    ipDefaultTTL            GET-REPLACE,
                                    ipInReceives            GET,
                                    ipInHdrErrors           GET,
                                    ipInAddrErrors          GET,
                                    ipForwDatagrams         GET,
                                    ipInUnknownProtos       GET,
                                    ipInDiscards            GET,
                                    ipInDelivers            GET,
                                    ipOutRequests           GET,
                                    ipOutDiscards           GET,
                                    ipOutNoRoutes           GET,
                                    ipReasmTimeout          GET,
                                    ipReasmReqds            GET,
                                    ipReasmOKs              GET,
                                    ipReasmFails            GET,
                                    ipFragOKs               GET,
                                    ipFragFails             GET,
                                    ipFragCreates           GET,
                                    ipRoutingDiscards       GET;;;
                REGISTERED AS   { mib 4};

    ipAddrEntry MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipAddrEntryPkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                                    ipAdEntryId             GET,
                                    ipAdEntAddr             GET,
                                    ipAdEntIfIndex          GET,
                                    ipAdEntNetMask          GET,
                                    ipAdEntBcastAddr        GET,
                                    ipAdEntReasmMaxSize GET;;;
```

```
            REGISTERED AS  { ipAddrTable 1};

    ipAddrTable MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipAddrTablePkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                                   ipAddrTableId           GET;;;
            REGISTERED AS  { ip 20};

    ipNetToMediaEntry        MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipNetToMediaEntryPkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                            ipNetToMediaEntryId      GET,
                            ipNetToMediaIfIndex      GET-REPLACE,
                            ipNetToMediaPhysAddress  GET-REPLACE,
                            ipNetToMediaNetAddress   GET-REPLACE,
                            ipNetToMediaType         GET-REPLACE;;;
            REGISTERED AS  { ipNetToMediaTable 1};

    ipNetToMediaTable MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipNetToMediaTablePkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                            ipNetToMediaTableId      GET;;;
            REGISTERED AS  { ip 22};

    ipRouteEntry MANAGED OBJECT CLASS
            DERIVED FROM  top;
            CHARACTERIZED BY
                ipRouteEntryPkg PACKAGE
                    -- see MIB II [16] for attribute semantics
                ATTRIBUTES
                            ipRouteEntryId    GET,
                            ipRouteDest       GET-REPLACE,
                            ipRouteIfIndex    GET-REPLACE,
                            ipRouteMetric1    GET-REPLACE,
                            ipRouteMetric2    GET-REPLACE,
                            ipRouteMetric3    GET-REPLACE,
                            ipRouteMetric4    GET-REPLACE,
                            ipRouteMetric5    GET-REPLACE,
                            ipRouteNextHop    GET-REPLACE,
```

```
                                 ipRouteType            GET-REPLACE,
                                 ipRouteProto           GET,
                                 ipRouteAge             GET-REPLACE,
                                 ipRouteMask            GET-REPLACE,
                                 ipRouteInfo            GET-REPLACE;;;
                    REGISTERED AS   { ipRoutingTable 1};


        ipRoutingTable MANAGED OBJECT CLASS
                DERIVED FROM  top;
                CHARACTERIZED BY
                    ipRoutingTablePkg PACKAGE
                        -- see MIB II [16] for attribute semantics
                    ATTRIBUTES
                                 ipRoutingTableId       GET;;;
                    REGISTERED AS   { ip 21};


        snmp MANAGED OBJECT CLASS
                DERIVED FROM   top;
                CHARACTERIZED BY
                    snmpPkg PACKAGE
                        -- see MIB II [16] for attribute semantics
                    ATTRIBUTES
                                 snmpId                   GET,
                                 snmpInPkts               GET,
                                 snmpInBadVersions        GET,
                                 snmpInBadCommunityNames  GET,
                                 snmpInBadCommunityUses   GET,
                                 snmpInASNParseErrs       GET,
                                 snmpInTooBigs            GET,
                                 snmpInNoSuchNames        GET,
                                 snmpInBadValues          GET,
                                 snmpInReadOnlys          GET,
                                 snmpInGenErrs            GET,
                                 snmpInTotalReqVars       GET,
                                 snmpInTotalSetVars       GET,
                                 snmpInGetRequests        GET,
                                 snmpInGetNexts           GET,
                                 snmpInSetRequests        GET,
                                 snmpInGetResponses       GET,
                                 snmpInTraps              GET,
                                 snmpOutPkts              GET,
                                 snmpOutTooBigs           GET,
                                 snmpOutNoSuchNames       GET,
                                 snmpOutBadValues         GET,
                                 snmpOutGenErrs           GET,
                                 snmpOutGetRequests       GET,
                                 snmpOutGetNexts          GET,
                                 snmpOutSetRequests       GET,
```

```
                              snmpOutGetResponses      GET,
                              snmpOutTraps             GET,
                              snmpEnableAuthenTraps    GET-REPLACE;
                 NOTIFICATIONS
                              snmpAuthentFailureEvent;;;
                 REGISTERED AS   {mib 11};

    system MANAGED OBJECT CLASS
           DERIVED FROM top;
           CHARACTERIZED BY
               systemPkg PACKAGE
                        -- SEE MIB II [16] for attribute semantics
                   ATTRIBUTES
                           sysDescr         GET,
                           sysObjectId      GET,
                           sysUpTime        GET,
                           sysContact       GET-REPLACE,
                           sysName          GET-REPLACE,
                           -- MAJOR HEALTH WARNING sysName is the
                           -- naming attribute and should not be
                           -- modified via remote management without
                           -- coordinated local system changes
                           sysLocation      GET-REPLACE,
                           sysServices      GET;
                   NOTIFICATIONS
                           coldStartEvent,
                           warmStartEvent,
                           entSpecificEvent;;;
    REGISTERED AS    {mib 1};

    tcp MANAGED OBJECT CLASS
            DERIVED FROM   top;
            CHARACTERIZED BY
               tcpPkg PACKAGE
                   -- see MIB II [16] for attribute semantics
               ATTRIBUTES
                           tcpId                GET,
                           tcpRtoAlgorithm      GET,
                           tcpRtoMin            GET,
                           tcpRtoMax            GET,
                           tcpMaxConn           GET,
                           tcpActiveOpens       GET,
                           tcpPassiveOpens      GET,
                           tcpAttemptFails      GET,
                           tcpEstabResets       GET,
                           tcpCurrEstab         GET,
                           tcpInSegs            GET,
                           tcpOutSegs           GET,
```

```
                             tcpRetransSegs        GET,
                             tcpInErrs             GET,
                             tcpOutRsts            GET;;;
                 REGISTERED AS { mib 6};


     tcpConnEntry MANAGED OBJECT CLASS
             DERIVED FROM  top;
             CHARACTERIZED BY
                 tcpConnEntryPkg PACKAGE
                     -- see MIB II [16] for attribute semantics
                 ATTRIBUTES
                             tcpConnId            GET,
                             tcpConnState         GET-REPLACE,
                             tcpConnLocalAddress  GET,
                             tcpConnLocalPort     GET,
                             tcpConnRemAddress    GET,
                             tcpConnRemPort       GET;;;
                 REGISTERED AS { tcpConnTable 1 };



     tcpConnTable MANAGED OBJECT CLASS
             DERIVED FROM   top;
             CHARACTERIZED BY
                 tcpConnTablePkg PACKAGE
                     -- see MIB II [16] for attribute semantics
                 ATTRIBUTES
                             tcpConnTableId       GET;;;
                 REGISTERED AS { tcp 13 };

     udp MANAGED OBJECT CLASS
             DERIVED FROM   top;
             CHARACTERIZED BY
                 udpPkg PACKAGE
                     -- see MIB II [16] for attribute semantics
                 ATTRIBUTES
                             udpId                GET,
                             udpInDatagrams       GET,
                             udpNoPorts           GET,
                             udpInErrors          GET,
                             udpOutDatagrams      GET;;;
                 REGISTERED AS  { mib 7};

      udpEntry MANAGED OBJECT CLASS
             DERIVED FROM  top;
             CHARACTERIZED BY
                 udpEntryPkg PACKAGE
                     -- see MIB II [16] for attribute semantics
                 ATTRIBUTES
```

```
                                udpEntryId           GET,
                                udpLocalAddress      GET,
                                udpLocalPort         GET;;;
                REGISTERED AS { udpTable 1 };


    udpTable MANAGED OBJECT CLASS
            DERIVED FROM   top;
            CHARACTERIZED BY
               udpTablePkg PACKAGE
                    -- see MIB II [16] for semantics
                  ATTRIBUTES
                                udpTableId           GET;;;
              REGISTERED AS { udp 5 };
```

5. Attribute Definitions

   Attribute templates for the attributes referenced in the managed
   object class templates of section 5 are listed here in alphabetical
   order.


```
atEntryId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
            MATCHES FOR       Equality;
       -- identifies the atEntry instance
       -- <index>".1." <address>, where <index is the decimal
       -- representation of atIfIndex and <address> is atNetAddress
       -- represented in "dot notation".  The "1" subidentifier
       -- indicates that the address is an IP address.
           REGISTERED AS {attributes 3};


atIfIndex ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
            MATCHES FOR       Equality;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {atEntry 1};


atNetAddress ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.NetworkAddress;
            MATCHES FOR       Equality;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {atEntry 3};


atPhysAddress ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.PhysAddress;
            MATCHES FOR       Equality;
```

                    -- SEE MIB II [16] for attribute semantics
                 REGISTERED AS {atEntry 2};

atTableId ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   OIM-Module.PrintString;
                 MATCHES FOR   Equality;
                    -- This attribute identifies the instance of
                    -- the address table which is being used.
                    -- An empty string is used for a single
                    -- instance.
                 REGISTERED AS {attributes 2};

egpAs   ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX OIM-Module.Integer;
                 MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                 REGISTERED AS {egp 6};

egpId   ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   OIM-Module.PrintString;
                 MATCHES FOR   Equality;
                    -- EGP entity naming attribute.
                    -- An empty string is used for a single instance.
                 REGISTERED AS {attributes 16};

egpInErrors ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                 REGISTERED AS {egp 2};

egpInMsgs ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                 REGISTERED AS {egp 1};

egpNeighAddr   ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX RFC1065-SMI.IpAddress;
                 MATCHES FOR   Equality;
                    -- SEE MIB II [16] for attribute semantics
                 REGISTERED AS {egpNeighEntry 2};

egpNeighAs   ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX OIM-Module.Integer;
                 MATCHES FOR   Equality, Ordering;

```
                       -- SEE MIB II [16] for attribute semantics
                   REGISTERED AS {egpNeighEntry 3};


egpNeighEntryId   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                   OIM-Module.PrintString;
                MATCHES FOR       Equality;
                   -- identifies the egp neighbor table entry.
                   -- egpNeighAddr encoded in "dot notation".
                       REGISTERED AS {attributes 22};


egpNeighEventTrigger    ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                   OIM-Module.EgpNeighEventTrigger;
                MATCHES FOR       Equality;
                   -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {egpNeighEntry 15};


egpNeighInErrMsgs        ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR       Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {egpNeighEntry 8};


egpNeighInErrs  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
                MATCHES FOR       Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {egpNeighEntry  5};


egpNeighInMsgs  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR       Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {egpNeighEntry 4};


egpNeighIntervalHello   ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                    OIM-Module.Integer;
          MATCHES FOR       Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {egpNeighEntry 12};


egpNeighIntervalPoll    ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                   OIM-Module.Integer;
                MATCHES FOR       Equality, Ordering;
```

```
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 13};


egpNeighMode     ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX  OIM-Module.EgpNeighMode;
                 MATCHES FOR       Equality;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 14};


egpNeighOutErrMsgs      ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                 MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 9};


egpNeighOutErrs ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                 MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 7};


egpNeighOutMsgs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
REGISTERED AS {egpNeighEntry 6};


egpNeighState ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX  OIM-Module.EgpNeighState;
                 MATCHES FOR       Equality;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 1};


egpNeighStateDowns      ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                 MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 11};


egpNeighStateUps        ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                 MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {egpNeighEntry 10};
```

```
egpNeighTableId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
            MATCHES FOR      Equality;
              -- This attribute uniquely identifies the
              -- egp neighbor table
              -- An empty string is used for a single instance.
          REGISTERED AS {attributes 17};


egpOutErrors ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {egp 4};


egpOutMsgs  ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {egp 3};


icmpId     ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
            MATCHES FOR      Equality;
              -- This attribute uniquely identifies the
              -- ICMP object instance
              -- An empty string is used for a single instance.
          REGISTERED AS {attributes 9};


icmpInAddrMaskReps ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 13};


icmpInAddrMasks ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 12};


icmpInDestUnreachs ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
```

```
                    -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 3};


icmpInEchoReps ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 9};


icmpInEchos  ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 8};


icmpInErrors ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 2};


icmpInMsgs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 1};


icmpInParmProbs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 5};


icmpInRedirects ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 7};


icmpInSrcQuenchs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {icmp 6};
```

```
icmpInTimeExcds ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 4};


icmpInTimestampReps ATTRIBUTE
      WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
      MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
   REGISTERED AS {icmp 11};


icmpInTimestamps ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 10};


icmpOutAddrMaskReps ATTRIBUTE
          WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
          MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
       REGISTERED AS {icmp 26};


icmpOutAddrMasks ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 25};


icmpOutDestUnreachs ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 16};


icmpOutEchoReps  ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {icmp 22};
```

```
icmpOutEchos ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 21};


icmpOutErrors ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 15};


icmpOutMsgs ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 14};


icmpOutParmProbs ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 18};


icmpOutRedirects ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 20};


icmpOutSrcQuenchs ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 19};


icmpOutTimeExcds ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {icmp 17};
```

icmpOutTimestampReps ATTRIBUTE
      WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
      MATCHES FOR   Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
   REGISTERED AS {icmp 24};

icmpOutTimestamps ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {icmp 23};

ifAdminStatus ATTRIBUTE
              WITH ATTRIBUTE SYNTAX OIM-Module.IfAdminStatus;
              MATCHES FOR   Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 7};

ifDescr ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                OIM-Module.PrintString;
              MATCHES FOR   Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 2};

ifEntryId ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 OIM-Module.PrintString;
              MATCHES FOR   Equality;
                 -- The decimal representation of ifIndex.
           REGISTERED AS {attributes 21};

ifId   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX OIM-Module.PrintString;
              MATCHES FOR   Equality;
                 -- An empty string is used for a single instance.
           REGISTERED AS {attributes 23};

ifIndex   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX OIM-Module.Integer;
              MATCHES FOR   Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 1};

ifInDiscards ATTRIBUTE
              WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;

```
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 13};


ifInErrors ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 14};


ifInNUcastPkts ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 12};


ifInOctets ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 10};


ifInUcastPkts ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 11};


ifInUnknownProtos ATTRIBUTE
      WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
     MATCHES FOR       Equality, Ordering;
        -- SEE MIB II [16] for attribute semantics
  REGISTERED AS {ifEntry 15};


ifLastChange ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  RFC1065-SMI.TimeTicks;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 9};


ifMtu ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
           MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ifEntry 4};


ifNumber  ATTRIBUTE
           WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
```

                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {interfaces 1};

ifOperStatus ATTRIBUTE
                WITH ATTRIBUTE SYNTAX OIM-Module.IfOperStatus;
                MATCHES FOR   Equality;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ifEntry 8};

ifOutDiscards ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ifEntry 19};

ifOutErrors ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;

                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ifEntry 20};

ifOutNUcastPkts ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
           MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ifEntry 18};

ifOutOctets ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;

                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ifEntry 16};

ifOutQLen ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Gauge;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ifEntry 21};

ifOutUcastPkts ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
           MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ifEntry 17};

```
ifPhysAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OIM-Module.PhysAddress;
        MATCHES FOR      Equality;
            -- SEE MIB II [16] for attribute semantics
    REGISTERED AS {ifEntry 6};


ifSpecific ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.ObjectId;
            MATCHES FOR      Equality;
            -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {ifEntry 22};


ifSpeed ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Gauge;
            MATCHES FOR      Equality, Ordering;
            -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {ifEntry 5};


ifTableId  ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
            MATCHES FOR      Equality;
              -- an empty string
         REGISTERED AS {attributes 1};


ifType ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.IfType;
          MATCHES FOR      Equality;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ifEntry 3};


ipAddrTableId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
          MATCHES FOR      Equality;
            -- This attribute provides a unique id
            -- which identifies the address table.
            -- An empty string is used for a single instance.
         REGISTERED AS {attributes 5};


ipAdEntAddr ATTRIBUTE
            WITH ATTRIBUTE SYNTAX   RFC1065-SMI.IpAddress;
            MATCHES FOR      Equality;
              -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {ipAddrEntry 1};


ipAdEntBcastAddr ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
```

```
        MATCHES FOR        Equality;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ipAddrEntry 4};


ipAdEntIfIndex ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
            MATCHES FOR        Equality;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {ipAddrEntry 2};


ipAdEntNetMask ATTRIBUTE
          WITH ATTRIBUTE SYNTAX   RFC1065-SMI.IpAddress;
          MATCHES FOR        Equality;
              -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ipAddrEntry 3};


ipAdEntReasmMaxSize ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  OIM-Module.Integer64k;
    MATCHES FOR        Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
  REGISTERED AS {ipAddrEntry 5};


ipAdEntryId  ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                OIM-Module.PrintString;
             MATCHES FOR        Equality;
                -- This attribute uniquely identifies the ip address
                -- table entry.  The format is:
                -- <addr> , where <addr> is ipAdEntAddr
                -- represented in "dot notation".
          REGISTERED AS {attributes 19};


ipDefaultTTL ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
            MATCHES FOR        Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {ip 2};


ipForwarding ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  OIM-Module.IpForwarding;
            MATCHES FOR        Equality;
                -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {ip 1};


ipForwDatagrams ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
        MATCHES FOR        Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
```

          REGISTERED AS {ip 6};

ipFragCreates ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 19};

ipFragFails ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 18};

ipFragOKs ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 17};

ipId ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                  OIM-Module.PrintString;
                MATCHES FOR   Equality;
                   -- identifies the Ip entity.
                   -- An empty string is used for a single instance
             REGISTERED AS {attributes 4};

ipInAddrErrors ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 5};

ipInDelivers ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 9};

ipInDiscards ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ip 8};

ipInHdrErrors ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;

```
                    MATCHES FOR       Equality, Ordering;
                        -- SEE MIB II [16] for attribute semantics
                  REGISTERED AS {ip 4};


ipInReceives ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR       Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {ip 3};


ipInUnknownProtos ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
        MATCHES FOR       Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
      REGISTERED AS {ip 7};


ipNetToMediaEntryId ATTRIBUTE
          WITH ATTRIBUTE SYNTAX
                    OIM-Module.PrintString;
            MATCHES FOR       Equality;
                    -- identifies an instance of IpNetToMediaEntry
                    -- <interface> "." <address>, where <interface> is the
                    -- decimal representation of ipNetToMediaIfIndex and
                    -- <address> is ipNetToMediaNetAddress in
                    -- "dot notation".
        REGISTERED AS {attributes 8};


ipNetToMediaIfIndex ATTRIBUTE
          WITH ATTRIBUTE SYNTAX    OIM-Module.Integer;
            MATCHES FOR       Equality;
                    -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ipNetToMediaEntry 1};


ipNetToMediaNetAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
            RFC1065-SMI.IpAddress;
        MATCHES FOR       Equality;
                    -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ipNetToMediaEntry 3};


ipNetToMediaPhysAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
            OIM-Module.PhysAddress;
        MATCHES FOR       Equality;
                    -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {ipNetToMediaEntry 2};
```

```
ipNetToMediaTableId  ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               OIM-Module.PrintString;
      MATCHES FOR       Equality;
               -- This is the distinguishing attribute for the
               -- Ip Net to Media Table.
               -- An empty string is used for a single instance.
   REGISTERED AS {attributes 7};


ipNetToMediaType ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                      OIM-Module.IpNetToMediaType;
              MATCHES FOR       Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ipNetToMediaEntry 4};


ipOutDiscards ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ip 11};


ipOutNoRoutes ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ip 12};


ipOutRequests ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ip 10};


ipReasmFails ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ip 16};


ipReasmOKs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
              MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {ip 15};


ipReasmReqds ATTRIBUTE
              WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
```

```
                  MATCHES FOR       Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {ip 14};


ipReasmTimeout ATTRIBUTE
            WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
             MATCHES FOR       Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {ip 13};

 ipRouteAge ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
               MATCHES FOR       Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ipRouteEntry 10};


ipRouteDest ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.IpAddress;
                MATCHES FOR       Equality;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ipRouteEntry 1};


ipRouteEntryId  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                  OIM-Module.PrintString;
                MATCHES FOR       Equality;
                   -- This attribute identifies the ip route table entry.
                   -- The format is <dest>, where <dest> is
                   -- ipRouteDest represented in "dot notation.
REGISTERED AS {attributes 20};


ipRouteIfIndex ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                MATCHES FOR       Equality;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ipRouteEntry 2};


ipRouteInfo ATTRIBUTE
               WITH ATTRIBUTE SYNTAX   OIM-Module.ObjectId;
                MATCHES FOR       Equality;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ipRouteEntry 13};


ipRouteMask ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.IpAddress;
                MATCHES FOR       Equality;
                   -- SEE MIB II [16] for attribute semantics
             REGISTERED AS {ipRouteEntry 11};
```

        ipRouteMetric1 ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                        MATCHES FOR   Equality, Ordering;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 3};

        ipRouteMetric2 ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                        MATCHES FOR   Equality, Ordering;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 4};

        ipRouteMetric3 ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                        MATCHES FOR   Equality, Ordering;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 5};

        ipRouteMetric4 ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                        MATCHES FOR   Equality, Ordering;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 6};

        ipRouteMetric5 ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer;
                        MATCHES FOR   Equality, Ordering;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 12};

        ipRouteNextHop ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX   RFC1065-SMI.IpAddress;
                       MATCHES FOR   Equality;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 7};

        ipRouteProto ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX OIM-Module.IpRouteProto;
                       MATCHES FOR   Equality;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 9};

        ipRouteType ATTRIBUTE
                     WITH ATTRIBUTE SYNTAX OIM-Module.IpRouteType;
                       MATCHES FOR   Equality;
                           -- SEE MIB II [16] for attribute semantics
                     REGISTERED AS {ipRouteEntry 8};

```
ipRoutingDiscards ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {ip 23};


ipRoutingTableId  ATTRIBUTE
         WITH ATTRIBUTE SYNTAX
             OIM-Module.PrintString;
         MATCHES FOR       Equality;
              -- unique id which identifies the
              -- Ip Routing Table instance.
              -- An empty string is used for a single instance.
        REGISTERED AS {attributes 6};


snmpEnableAuthenTraps   ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                      OIM-Module.SnmpEnableAuthenTraps;
            MATCHES FOR       Equality;
              -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {snmp 30};


snmpId  ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
             OIM-Module.PrintString;
            MATCHES FOR       Equality;
                     -- naming attribute for snmp
                     -- An empty string is used for a single instance.
         REGISTERED AS {attributes 18};


snmpInASNParseErrs      ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               RFC1065-SMI.Counter;
            MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {snmp 6};


snmpInBadCommunityNames ATTRIBUTE
    WITH ATTRIBUTE SYNTAX
         RFC1065-SMI.Counter;
    MATCHES FOR       Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
  REGISTERED AS {snmp 4};


snmpInBadCommunityUses  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX
               RFC1065-SMI.Counter;
    MATCHES FOR       Equality, Ordering;
```

                         -- SEE MIB II [16] for attribute semantics
       REGISTERED AS {snmp 5};

snmpInBadValues ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                          RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 10};

snmpInBadVersions   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 3};

snmpInGenErrs   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 12};

snmpInGetNexts ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 16};

snmpInGetRequests   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 15};

snmpInGetResponses   ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
        MATCHES FOR   Equality, Ordering;
                  -- SEE MIB II [16] for attribute semantics
       REGISTERED AS {snmp 18};

snmpInNoSuchNames   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;

                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 9};

    snmpInPkts   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX   RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 1};

    snmpInReadOnlys ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                          RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 11};

    snmpInSetRequests   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 17};

    snmpInTooBigs   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 8};

    snmpInTotalReqVars   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 13};

    snmpInTotalSetVars   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 14};

    snmpInTraps   ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
                REGISTERED AS {snmp 19};

snmpOutBadValues   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                        RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 22};

snmpOutGenErrs ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                        RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 24};

snmpOutGetNexts ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                  RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 26};

snmpOutGetRequests   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                  RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 25};

snmpOutGetResponses   ATTRIBUTE
          WITH ATTRIBUTE SYNTAX
                    RFC1065-SMI.Counter;
          MATCHES FOR   Equality, Ordering;
                    -- SEE MIB II [16] for attribute semantics
      REGISTERED AS {snmp 28};

snmpOutNoSuchNames   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                  RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 21};

snmpOutPkts   ATTRIBUTE
              WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
              MATCHES FOR   Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {snmp 2};

```
snmpOutSetRequests      ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {snmp 27};


snmpOutTooBigs  ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                          RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {snmp 20};


snmpOutTraps    ATTRIBUTE
              WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
              MATCHES FOR      Equality, Ordering;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {snmp 29};


sysContact ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 OIM-Module.PrintString;
             MATCHES FOR                 Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {system 4};


sysDescr ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 OIM-Module.PrintString;
             MATCHES FOR                 Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {system 1};


sysLocation     ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 OIM-Module.PrintString;
             MATCHES FOR                 Equality;
                 -- SEE MIB II [16] for attribute semantics
           REGISTERED AS {system 6};


sysName         ATTRIBUTE
              WITH ATTRIBUTE SYNTAX
                 OIM-Module.PrintString;
             MATCHES FOR                 Equality;
                 -- SEE MIB II [16] for attribute semantics
                 -- The nodes domain name.
           REGISTERED AS {system 5};
```

```
sysObjectId      ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.ObjectId;
          MATCHES FOR                 Equality;
             -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {system 2};


sysServices      ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  OIM-Module.Integer128;
          MATCHES FOR                 Equality;
             -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {system 7};


sysUpTime        ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.TimeTicks;
          MATCHES FOR               Equality, Ordering;
             -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {system 3};


tcpActiveOpens ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
          MATCHES FOR      Equality, Ordering;
             -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {tcp 5};


tcpAttemptFails  ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              RFC1065-SMI.Counter;
          MATCHES FOR      Equality, Ordering;
             -- SEE MIB II [16] for attribute semantics
        REGISTERED AS {tcp 7};


tcpConnId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
          MATCHES FOR      Equality;
            -- identifies an instance of tcpConnEntry
            -- <laddr> "." <lport> "." <raddr> "." <rport>, where
            -- <laddr> is the "dot notation" representation of
            -- tcpConnLocalAddress,
            -- <lport> is the decimal representation of
            -- tcpConnLocalPort,
            -- <raddr> is the "dot notation" representation of
            -- tcpConnRemAddress, and
            --<rport> is the decimal representation of
            -- tcpConnRemPort.
        REGISTERED AS {attributes 12};
```

tcpConnLocalAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX RFC1065-SMI.IpAddress;
     MATCHES FOR   Equality;
        -- SEE MIB II [16] for attribute semantics
REGISTERED AS {tcpConnEntry 2};

tcpConnLocalPort ATTRIBUTE
     WITH ATTRIBUTE SYNTAX   OIM-Module.Integer64k;
     MATCHES FOR   Equality, Ordering;
        -- SEE MIB II [16] for attribute semantics
REGISTERED AS {tcpConnEntry 3};

tcpConnRemAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX RFC1065-SMI.IpAddress;
     MATCHES FOR   Equality;
        -- SEE MIB II [16] for attribute semantics
REGISTERED AS {tcpConnEntry 4};

tcpConnRemPort ATTRIBUTE
            WITH ATTRIBUTE SYNTAX   OIM-Module.Integer64k;
            MATCHES FOR   Equality, Ordering;
               -- SEE MIB II [16] for attribute semantics
REGISTERED AS {tcpConnEntry 5};

tcpConnState ATTRIBUTE
            WITH ATTRIBUTE SYNTAX OIM-Module.TcpConnState;
            MATCHES FOR   Equality;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {tcpConnEntry 1};

tcpConnTableId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
              OIM-Module.PrintString;
            MATCHES FOR   Equality;
              -- This attribute defines a unique index of
              -- tcpConnEntry in the TCP Connection Table
              -- An empty string is used for a single instance.
          REGISTERED AS {attributes 11};

tcpCurrEstab ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Gauge;
            MATCHES FOR   Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {tcp 9};

tcpEstabResets ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
            MATCHES FOR   Equality, Ordering;

                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 8};

tcpId   ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   OIM-Module.PrintString;
                 MATCHES FOR   Equality;
                   -- This attribute uniquely identified a tcp
                   -- object instance
                   -- An empty string is used for a single instance.
              REGISTERED AS {attributes 10};

tcpInErrs ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 14};

tcpInSegs ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 10};

tcpMaxConn ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX OIM-Module.Integer;
                 MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 4};

tcpOutRsts ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 15};

tcpOutSegs ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;
                   -- SEE MIB II [16] for attribute semantics
              REGISTERED AS {tcp 11};

tcpPassiveOpens ATTRIBUTE
                 WITH ATTRIBUTE SYNTAX
                   RFC1065-SMI.Counter;
                 MATCHES FOR   Equality, Ordering;

```
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {tcp 6};


tcpRetransSegs ATTRIBUTE
                WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
                MATCHES FOR      Equality, Ordering;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {tcp 12};


tcpRtoAlgorithm ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    OIM-Module.TcpRtoAlgorithm;
                MATCHES FOR      Equality;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {tcp 1};


tcpRtoMax  ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
                MATCHES FOR      Equality;
                -- SEE MIB II [16] for attribute semantics
         REGISTERED AS {tcp 3};


tcpRtoMin ATTRIBUTE
                WITH ATTRIBUTE SYNTAX  OIM-Module.Integer;
                MATCHES FOR      Equality;
                -- SEE MIB II [16] for attribute semantics
            REGISTERED AS {tcp 2};


udpEntryId ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    OIM-Module.PrintString;
                MATCHES FOR      Equality;
                -- This attribute defines an entry
                -- for the UDP Listener Table
                -- <port> "." <address>, where <port> is the decimal
                -- representation of udpLocalPort and <address> is
                -- udpLocalAddress represented in "dot notation".
            REGISTERED AS {attributes 15};


udpId      ATTRIBUTE
                WITH ATTRIBUTE SYNTAX
                    OIM-Module.PrintString;
                MATCHES FOR      Equality;
                -- This attribute provides a unique id to
                -- identify the UDP entity.
                -- An empty string is used for a single instance.
            REGISTERED AS {attributes 13};
```

```
udpInDatagrams ATTRIBUTE
            WITH ATTRIBUTE SYNTAX RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udp 1};


udpInErrors ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udp 3};


udpLocalAddress ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               RFC1065-SMI.IpAddress;
            MATCHES FOR      Equality;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udpEntry 1};


udpLocalPort ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               OIM-Module.Integer64k;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udpEntry 2};


udpNoPorts ATTRIBUTE
            WITH ATTRIBUTE SYNTAX  RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udp 2};


udpOutDatagrams ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               RFC1065-SMI.Counter;
            MATCHES FOR      Equality, Ordering;
              -- SEE MIB II [16] for attribute semantics
          REGISTERED AS {udp 4};


udpTableId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX
               OIM-Module.PrintString;
            MATCHES FOR      Equality;
              -- This attribute defines a unique id
              -- for the UDP Listener Table
              -- An empty string is used for a single instance.
          REGISTERED AS {attributes 14};
```

6. Notifications

   Notification templates for the notifications referenced in the
   managed object class templates of section 5 are listed here in
   alphabetical order.

   coldStartEvent NOTIFICATION
                   -- clause 4.1.6.1 of RFC1157
                   -- emitted by system
           MODE   NON-CONFIRMED;
           WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo
   REGISTERED AS {events 0};

   egpNeighborLossEvent NOTIFICATION
                   -- clause 4.1.6.6 of RFC1157
                                   -- with generic-trap = 5
                                   -- The first VarBindList element
                                   -- contains the name and value of the
                                   -- egpNeighAddr of the affected
                                   -- neighbor.
                                   -- emitted by egpNeighEntry
           MODE   NON-CONFIRMED;
           WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
   REGISTERED AS {events 5};

   linkDownEvent NOTIFICATION
                   -- clause 4.1.6.3 of RFC1157
                   -- emitted by ifEntry
                   -- with generic-trap = 2
           MODE   NON-CONFIRMED;
           WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
   REGISTERED AS {events 2};

   linkUpEvent NOTIFICATION
                   -- clause 4.1.6.4 of RFC1157
                   -- emitted by ifEntry
                   -- with generic-trap = 3
           MODE   NON-CONFIRMED;
           WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
   REGISTERED AS {events 3};

   snmpAuthentFailureEvent NOTIFICATION
                   -- clause 4.1.6.5 of RFC1157
                   -- emitted by snmp
           MODE   NON-CONFIRMED;
           WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
   REGISTERED AS {events 4};

```
     entSpecificEvent NOTIFICATION
                    -- clause 4.1.6.7 of RFC1157
                    -- emitted by system
             MODE    NON-CONFIRMED;
             WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
     REGISTERED AS {events 6};


     warmStartEvent NOTIFICATION
                    -- clause 4.1.6.2 of RFC1157
                    -- emitted by system
             MODE    NON-CONFIRMED;
             WITH INFORMATION SYNTAX OIM-Module.SnmpTrapInfo;
     REGISTERED AS {events 1};
```

7. The Containment Hierarchy

   Name Binding templates that define the containment hierarchy for the
   OIM MIB-II are listed here in alphabetical order.   The containment
   hierarchy within a managed system begins at the "system" managed
   object.


```
     atEntry-nf  NAME BINDING
             SUBORDINATE OBJECT CLASS     atEntry;
             NAMED BY SUPERIOR OBJECT CLASS   atTable;
             WITH ATTRIBUTE          atEntryId;
            CREATE;
            DELETE        only-if-no-contained-objects;
             REGISTERED AS { nameForms 5};


     atTable-nf  NAME BINDING
             SUBORDINATE OBJECT CLASS     atTable;
             NAMED BY SUPERIOR OBJECT CLASS   system;
             WITH ATTRIBUTE          atTableId;
             REGISTERED AS { nameForms 4};


     egp-nf  NAME BINDING
             SUBORDINATE OBJECT CLASS     egp;
             NAMED BY SUPERIOR OBJECT CLASS       system;
             WITH ATTRIBUTE         egpId;
             REGISTERED AS { nameForms 20};


     egpNeighEntry-nf  NAME BINDING
             SUBORDINATE OBJECT CLASS     egpNeighEntry;
             NAMED BY SUPERIOR OBJECT CLASS      egpNeighTable;
             WITH ATTRIBUTE          egpNeighEntryId;
            CREATE;
            DELETE        only-if-no-contained-objects;
```

```
                REGISTERED AS { nameForms 22};

   egpNeighTable-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      egpNeighTable;
                NAMED BY SUPERIOR OBJECT CLASS       egp;
                WITH ATTRIBUTE        egpNeighTableId;
                REGISTERED AS { nameForms 21};

   icmp-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      icmp;
                NAMED BY SUPERIOR OBJECT CLASS       system;
                WITH ATTRIBUTE        icmpId;
                REGISTERED AS { nameForms 13};

   interfaces-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      interfaces;
                NAMED BY SUPERIOR OBJECT CLASS   system;
                WITH ATTRIBUTE        ifId;
                REGISTERED AS { nameForms 24};


   ifEntry-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      ifEntry;
                NAMED BY SUPERIOR OBJECT CLASS   ifTable;
                WITH ATTRIBUTE        ifEntryId;
              CREATE;
              DELETE       only-if-no-contained-objects;
               REGISTERED AS { nameForms 3};

   ifTable-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      ifTable;
                NAMED BY SUPERIOR OBJECT CLASS  interfaces;
                WITH ATTRIBUTE   ifTableId;
                REGISTERED AS { nameForms 2};

   ip-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      ip;
                NAMED BY SUPERIOR OBJECT CLASS   system;
                WITH ATTRIBUTE        ipId;
                REGISTERED AS       { nameForms 6};

   ipAddrEntry-nf  NAME BINDING
                SUBORDINATE OBJECT CLASS      ipAddrEntry;
                NAMED BY SUPERIOR OBJECT CLASS  ipAddrTable;
                WITH ATTRIBUTE        ipAdEntryId;
              CREATE;
              DELETE       only-if-no-contained-objects;
               REGISTERED AS { nameForms 8};
```

```
ipAddrTable-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      ipAddrTable;
          NAMED BY SUPERIOR OBJECT CLASS    ip;
          WITH ATTRIBUTE           ipAddrTableId;
          REGISTERED AS { nameForms 7};


ipNetToMediaEntry-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      ipNetToMediaEntry;
          NAMED BY SUPERIOR OBJECT CLASS  ipNetToMediaTable;
          WITH ATTRIBUTE          ipNetToMediaEntryId;
        CREATE;
        DELETE         only-if-no-contained-objects;
          REGISTERED AS { nameForms 12};


ipNetToMediaTable-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      ipNetToMediaTable;
          NAMED BY SUPERIOR OBJECT CLASS  ip;
          WITH ATTRIBUTE          ipNetToMediaTableId;
          REGISTERED AS { nameForms 11};



ipRouteEntry-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      ipRouteEntry;
          NAMED BY SUPERIOR OBJECT CLASS  ipRoutingTable;
          WITH ATTRIBUTE          ipRouteEntryId;
        CREATE;
        DELETE         only-if-no-contained-objects;
          REGISTERED AS { nameForms 10};


ipRoutingTable-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      ipRoutingTable;
          NAMED BY SUPERIOR OBJECT CLASS    ip;
          WITH ATTRIBUTE          ipRoutingTableId;
          REGISTERED AS { nameForms 9};


snmp-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      snmp;
          NAMED BY SUPERIOR OBJECT CLASS      system;
          WITH ATTRIBUTE          snmpId;
          REGISTERED AS { nameForms 23};


system-nf   NAME BINDING
          SUBORDINATE OBJECT CLASS      system;
          NAMED BY SUPERIOR OBJECT CLASS  root;
          WITH ATTRIBUTE   sysName;
          REGISTERED AS { nameForms 1 };
```

```
tcp-nf   NAME BINDING
         SUBORDINATE OBJECT CLASS     tcp;
         NAMED BY SUPERIOR OBJECT CLASS       system;
         WITH ATTRIBUTE          tcpid;
         REGISTERED AS { nameForms 14};

tcpConnEntry-nf  NAME BINDING
         SUBORDINATE OBJECT CLASS     tcpConnEntry;
         NAMED BY SUPERIOR OBJECT CLASS       tcpConnTable;
         WITH ATTRIBUTE          tcpConnId;
         REGISTERED AS { nameForms 16};

tcpConnTable-nf  NAME BINDING
         SUBORDINATE OBJECT CLASS     tcpConnTable;
         NAMED BY SUPERIOR OBJECT CLASS       tcp;
         WITH ATTRIBUTE          tcpConnTableId;
         REGISTERED AS { nameForms 15};

udp-nf   NAME BINDING
         SUBORDINATE OBJECT CLASS     udp;
         NAMED BY SUPERIOR OBJECT CLASS       system;
         WITH ATTRIBUTE          udpId;
         REGISTERED AS { nameForms 17};

udpEntry-nf  NAME BINDING
         SUBORDINATE OBJECT CLASS     udpEntry;
         NAMED BY SUPERIOR OBJECT CLASS       udpTable;
         WITH ATTRIBUTE          udpEntryId;
         REGISTERED AS { nameForms 19};

udpTable-nf  NAME BINDING
         SUBORDINATE OBJECT CLASS     udpTable;
         NAMED BY SUPERIOR OBJECT CLASS       udp;
         WITH ATTRIBUTE          udpTableId;
         REGISTERED AS { nameForms 18};
```

8. ASN.1 Definitions

   The ASN.1 syntax referenced by the attribute templates in section 6
   are either defined in RFC1065-SMI, the Attribute-ASN1Module of
   ISO/IEC DIS 10165-2, or in the OIM-Module defined in this section.
   This module includes some syntax definitions taken from the RFC1158
   since they were not included in that document inside an ASN.1 module
   and hence could not be externally referenced.

```
OIM-Module        {iso org(3) dod(6) internet(1) mgmt(2) mib(1)
                        oim(9) misc(4) 1}
DEFINITIONS ::=
BEGIN
-- EXPORTS      Everything

IMPORTS ObjectClass, ObjectInstance
            FROM CMIP-1{joint-iso-ccitt ms(9) cmip(1)
                            modules(0) protocol(3)}
            ObjectName, ObjectSyntax
            FROM RFC1065-SMI;

mib                  OBJECT IDENTIFIER ::= {iso org(3) dod(6)
                                    internet(1) mgmt(2) 1}
at                   OBJECT IDENTIFIER ::= {mib 3}
oim                  OBJECT IDENTIFIER ::= {mib 9}
cmotVersion    OBJECT IDENTIFIER ::= {oim 1}
cmotACSEInfo   OBJECT IDENTIFIER ::= {oim 2}
cmotSystemId   OBJECT IDENTIFIER ::= {oim 3}
misc                 OBJECT IDENTIFIER ::= {oim 4}
objects        OBJECT IDENTIFIER ::= {oim 5}
attributes           OBJECT IDENTIFIER ::= {oim 6}
events         OBJECT IDENTIFIER ::= {oim 7}
nameforms      OBJECT IDENTIFIER ::= {oim 8}
actions        OBJECT IDENTIFIER ::= {oim 9}

--   Generic and MIB specific syntax

EgpNeighEventTrigger    ::= INTEGER {
                                        start(1),
                                        stop(2)
                                        }

EgpNeighMode    ::= INTEGER {
                                active(1),
                                passive(2)
                                }

EgpNeighState   ::= INTEGER {
```

```
                                        idle(1),
                                        acquisition(2),
                                        down(3),
                                        up(4),
                                        cease(5)
                                        }

Integer          ::= INTEGER

Integer128       ::= INTEGER (0..127)

Integer64k       ::= INTEGER (0..65535)

IfAdminStatus    ::= INTEGER {
                                        up(1),  -- ready to pass packets
                                        down(2),
                                        testing(3)  -- in some test mode
                                        }

IfOperStatus     ::= INTEGER {
                                        up(1),  -- ready to pass packets
                                        down(2),
                                        testing(3)  -- in some test mode
                                        }

IfType INTEGER {
                                other(1),  -- none of the following
                                regular1822(2),
                                hdh1822(3),
                                ddn-x25(4),
                                rfc877-x25(5),
                                ethernet-csmacd(6),
                                iso88023-csmacd(7),
                                iso88024-tokenBus(8),
                                iso88025-tokenRing(9),
                                iso88026-man(10),
                                starLan(11),
                                proteon-10Mbit(12),
                                proteon-80Mbit(13),
                                hyperchannel(14),
                                fddi(15),
                                lapb(16),
                                sdlc(17),
                                ds1(18),
                                e1(19),  -- european equivalent
                                        -- of T-1
                                basicISDN(20),
                                primaryISDN(21),
```

```
                                    -- proprietary serial
                             propPointToPointSerial(22),
                             ppp(23),
                             softwareLoopback(24),
                             eon(25),  -- CLNP over IP
                             ethernet-3Mbit(26),
                             nsip(27),  -- XNS over IP
                             slip(28),  -- generic SLIP
                         ultra(29),  -- ULTRA technologies
                         ds3(30),  -- T-3
                         sip(31),  -- SMDS
                         frame-relay(32)
                }

IpForwarding    ::= INTEGER {
                         forwarding(1),  -- acting as a gateway
                         not-forwarding(2)  -- NOT acting as
                                         -- a gateway
                             }

IpNetToMediaType ::= INTEGER {
                             other(1),  -- none of the
                                     -- following
                             invalid(2),  -- an invalidated
                                       -- mapping
                             dynamic(3),
                             static(4)  }

IpRouteProto    ::= INTEGER {
                             other(1),  -- none of the
                                     -- following
                             local(2),  -- configured entries
                             netmgmt(3),  -- mgmt protocol
                             icmp(4),  -- obtained via ICMP,
                                     -- e.g., redirect
                             egp(5),
                             ggp(6),
                             hello(7),
                             rip(8),
                             is-is(9),
                             es-is(10),
                             ciscoIgrp(11),
                             bbnSpfIgp(12),
                             ospf(13),
                             bgp(14)
                             }
```

```
IpRouteType      ::= INTEGER {
                                     other(1),  -- none of the
                                             -- following
                                     invalid(2),  -- an invalidated
                                             -- route
                                     direct(3),  -- route to directly
                                             -- connected
                                             -- subnetwork
                                     indirect(4)  -- route to a
                                             -- non-local
                                             -- host/network/
                                             -- subnet
                                     }
```

```
ObjectId          ::= OBJECT IDENTIFIER

OctetString       ::= OCTET STRING

PhysAddress       ::= OCTET STRING

PrintString       ::= DisplayString (SIZE (0..255))

DisplayString     ::= OCTET STRING
--
-- DisplayString is restricted to the NVT ASCII character set as
-- defined in pages 10-11 of RFC 854 "TELNET Protocol
-- Specification", May 1983 [15].
--
--  SNMP trap syntax

SnmpEnableAuthenTraps   ::= INTEGER {
                                       enabled(1),
                                       disabled(2)
                                       }

SnmpTrapInfo ::= SEQUENCE {
        enterprise                  -- type of device generating
                                    -- event
                                    -- see sysObjectId
                                    OBJECT IDENTIFIER,
        agent-addr            -- address of device generating
                                    -- the event
                                    NetworkAddress,
        generic-trap INTEGER {coldStart (0),
                                    warmStart (1),
                                    linkDown (2),
                                    linkUp (3),
                                    authenticationFailure (4),
                                    egpNeighborLoss (5),
                                    enterpriseSpecific (6) },
        specific-trap INTEGER   OPTIONAL, -- enterprise specific
        variable-bindings   VarBindList OPTIONAL}

VarBindList ::= SEQUENCE OF VarBind

VarBind ::=      SEQUENCE {
                                    name    ObjectName,
                                    ovalue  ObjectSyntax}

TcpConnState     ::= INTEGER {
                                    closed(1),
                                    listen(2),
```

```
                                     synSent(3),
                                     synReceived(4),
                                     established(5),
                                     finWait1(6),
                                     finWait2(7),
                                     closeWait(8),
                                     lastAck(9),
                                     closing(10),
                                     timeWait(11),
                                     deleteTCP(12) }


TcpRtoAlgorithm  ::= INTEGER {

                                     other(1),  -- none of the
                                                -- following
                                     constant(2),  -- a constant rto
                                     rsre(3),  -- MIL-STD-1778,
                                                -- Appendix B
                                     vanj(4)  -- Van Jacobsons alg.
                                       }

     END
```

9. Acknowledgements

   The editor acknowledges the contributions of the members of the OIM
   working group. Particular thanks are made to the following for their
   comments and assistance:

        Marvin Solomon, University of Wisconsin
        Nancy Hall, University of Wisconsin
        Subhendu Ghatak, University of Tennessee at Knoxville

References

    1. ISO 8824: Information Processing - Open System Interconnection -
        Specification of Abstract Syntax Notation One (ASN.1), February
        1989.

    2. ISO/IEC 7498-4, Information Processing Systems- Open Systems
        Interconnection - Basic Reference Model Part 4 - OSI Management
        Framework.

    3. Warrier, U., Besaw, L., and LaBarre, L., and B. Handspicker, "The
        Common Management Information Services and Protocol for the
        Internet (CMOT and CMIP)", RFC 1189, Netlabs, Hewlett-Packard,
        The Mitre Corporation, Digital Equipment Corporation, October
        1990.

    4. ISO/IEC DIS 10165-1, Information Processing Systems - Open
        Systems Interconnection-Structure of Management Information -
        Part 1: Management Information Model, July 1990.

    5. ISO/IEC DIS 10165-2 (ISO/IEC JTC1/SC21 N4072), Information
        Processing Systems -Open Systems Interconnection - Structure of
        Management Information - Part 2: Definition of Management
        Information, July 1990.

    6. ISO/IEC DIS 10165-4 (ISO/IEC JTC1/SC21 N4065), Information
        Processing Systems - Open Systems Interconnection - Structure of
        Management Information - Part 4: Guidelines for the Definition of
        Managed Objects, June 1990.

    7. ISO 9595, Information Processing Systems - Open Systems
        Interconnection - Management Information Service Definition -
        Common Management Information Service, November 1990.

    8. ISO 9596, Information Processing Systems - Open Systems
        Interconnection - Management Information Protocol Specification -
        Common Management Information Protocol, November 1990.

    9. ISO 8649, Information Processing Systems - Open Systems
        Interconnection, Service Definition for the Association Control
        Service Element.

   10. ISO 9072-1, Information Processing Systems - Text Communication,
        Remote Operations: Model, Notation and Service Definition,
        Gloucester, Nov 1987.

   11. Rose, M., and K. McCloghrie, "Structure and Identification of
        Management Information for TCP/IP-based internets", RFC 1155,

Performance Systems International, Hughes LAN Systems, May 1990.

   12. Rose, M., Editor, "Management Information Base for Network
       Management of TCP/IP-based internets: MIB-II", RFC 1158,
       Performance Systems International, May 1990.

   13. Cerf, V., "Report of the Second Ad Hoc Network Management Review
       Group", RFC 1109, NRI, August 1989.

   14. Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple
       Network Management Protocol (SNMP)", RFC 1157, SNMP Research,
       Performance Systems International, Performance Systems
       International, MIT Laboratory for Computer Science, May 1990.

   15. Postel, J., and J. Reynolds, "Telnet Protocol Specification", RFC
       854, USC/Information Sciences Institute, May 1983.

   16. McCloghrie, K., and M. Rose, "Management Information Base for
       Network Management of TCP/IP-based internets: MIB-II", RFC 1213,
       Hughes LAN Systems, Performance Systems International, March
       1991.

   17. Working Implementation Agreements for Open Systems
       Interconnection Protocols, December 1990.

Appendix 1: Notational Tools for Managed Object Definition

   This section provides descriptions of the notational tools used to
   define the templates defined in this memorandum. The text is
   excerpted and paraphrased from reference 6. Only the templates used
   in this document are included here. For a complete description of
   the notational tools specified for OSI management, see reference 6.

A1.1 Overview of Notational Tools

   The "Templates" defined in this clause provide a common set of tools
   and a common format for the representation of the various aspects of
   a managed object class definition and its associated naming
   structure. Formal definitions of each template will be found in
   clauses A1.3 -A1.11 inclusive; the syntactic conventions used in
   these formal definitions are specified in clause A1.2. Examples of
   the use of these tools may be found in Annex A (of 10165-4).

   The structure and behaviour of a managed object class is primarily
   defined by means of the Managed Object Class Template. This template
   identifies the inheritance relationships that exist between the class
   and other managed object classes, the packages of class specific
   behaviour, the attributes that are associated with the class, the
   notifications that the class may issue and the operations that may be
   performed upon the class. In order to allow re-use of parts of this
   specification in the specification of other managed object classes,
   additional templates are defined to provide for the specification of
   attributes (individual and group), behaviour, actions, notifications
   and conditional packages. These other templates are "called up" by
   the Managed Object Class template by means of the referencing
   mechanism defined in clause A1.2; this mechanism allows references to
   be made in one standard to specifications that are contained in other
   standards, hence allowing "generic" specifications to be made
   available for use in managed object class definitions in addition to
   local specifications. Templates may also be included "in-line" if so
   desired.

   The naming of a managed object class is defined by means of the Name
   Binding template. This template identifies the managed object class
   being named and defines the relative distinguished name that will be
   used to name instances of the class in the context of a particular
   superior class. This template also provides for the specification of
   relationships that exist between two object classes as a consequence
   of a particular name binding.

A1.2 Conventions Used in Template Definitions

   The templates definitions contained in this clause are based on the
   following syntactic conventions:

   - Strings surrounded by [ ] delimit parts of the template that may
     either be present or absent in each instance of use of the
     template. If the closing brace is followed by an asterisk,
     e.g., [ ]*, the contents of the braces may appear zero or more
     times. The circumstances under which these parts of the
     definition may be omitted or repeated are dependent upon the
     template type;

   - Strings surrounded by < > delimit strings that must be replaced in
     each instance of use of the template. The structure and meaning
     of the replacement string is dependent upon the string type;

   - Upper case strings denote Keywords which are required to be
     present in each instance of use of the template, unless they are
     enclosed in [ ];

   - The solidus, |, is used to separate alternative strings in the
     template;

   - Spaces and carriage returns are significant only as string
     delimiters;

   - In order to simplify the structure of the templates, particularly
     where the same syntactic structure is repeated in the template
     definitions, supporting syntactic definitions may be defined. If
     any such supporting syntactic definitions are required in order to
     complete the template definition, these appear following the
     keywords "supporting productions" at the end of the template
     definition. These supporting definitions take the general form

                 definition-label-> syntactic-definition

   where definition-label allows the definition to be referenced by the
   template definition and syntactic-definition gives the expansion of
   the definition, using the above notational rules;

   - The templates themselves have the general structure:

                 <template-label>TEMPLATE NAME

         [ELEMENT NAME [<element-definition>] ;]*

         [REGISTERED AS <object-identifier> ;]

           [supporting productions

           [<definition-label> -> <syntactic-definition >]*]

     Each instance of use of a template therefore declares a template-
     label by which that instance may be referenced from other templates,
     and if the REGISTERED AS clause is present, assigns a value of an
     ASN.1 OBJECT IDENTIFIER under which the instance has been registered;

     - Semicolons are used to mark the end of each distinct element in
       the template. In an instance of use of the template, any text
       inserted between a semicolon and the next valid keyword is
       assumed to be a comment;

     - The notation used for representing object identifiers is the
       normal notation defined in ASN.1 for object identifier values;
       i.e., the production:

           object-identifier-> <ASN.l ObjectIdentifierValue notation>

           is assumed to be a supporting production for all template
           definitions in this document;

     - Further definitions, such as the nature of the element definitions,
       follow the syntactic definition of the template;

     - Template labels shall be unique within the standard or document
       in which they are declared. Where a template-label is declared in
       document A and referenced in document B, the reference in
       document B shall be prefixed by the globally unique name of
       document A. In the case of documents named by an
       internationally recognized naming authority such as
       [CCITT|ISO/IEC], the registered name of the document shall be
       used as the identifier, such as [Recommendation X.722|ISO/IEC
       10165-4]. Where a globally unique name is not already available,
       it is permissible to assign the value of an OBJECT IDENTIFIER to
       the referenced document, and use this value as a globally unique
       document name. The syntax of a template-label, defined using the
       above notation, is defined as follows:

           [<document-identifier> :] < label-string>

           document-identifier-> "<standard-name>" | <object-identifier>

     A label-string may include any number of the following characters:

           -         upper or lower case alphabetic characters;

                  -        the digits 0-9;

                  -        the hyphen character (-);

    in any order, commencing with a lower case character. For example,
    the following template-label:

         "ISO/IEC 10165/4":exampleObjectClass

    constitutes a globally unique label for the definition of
    exampleObjectClass contained in Annex A of 10165-4.

    Label references that are not prefixed by a document-identifier are
    assumed to refer to labels declared in the document in which the
    reference appears.

    - Whenever a template-label is present in a template as a
       pointer to another template, it may be replaced by the entire
       text of the referenced template itself (including the
       template's label).

A1.3 Managed Object Class Template

A1.3.1 Overview

    The Managed Object Class template forms the basis of the formal
    definition of a managed object. Elements in the template allow the
    class to be placed at the appropriate node of the inheritance tree,
    the various attributes of the class to be specified, and the
    behaviour of the class to be defined. The major elements of the
    definition are:

    - Inheritance. Each managed object class defines the
      superclass(es) from which it has been derived. All
      characteristics of the superclass(es) are inherited by the
      subclass; the subclass definition may add to these
      characteristics (refinement) but may not remove any
      characteristics of the superclass. Ultimately, all classes are
      subclasses of TOP;

    - Allomorphs. If the class supports allomorphism, the set of
      classes that are legitimate allomorphic views of the class must
      be defined. These classes must all be superclasses of the class
      being defined;

    - Mandatory Packages. The managed object class definition
      includes the packages of behaviour, attributes, operations and
      notifications that provide a complete specification of the

behaviour that is common to all instances of the class;

- Conditional Packages. The managed object class definition
  includes the specification of packages of behaviour, attributes,
  operations and notifications that are present or absent in
  instances of that class as a consequence of a specified condition;

Note: Attributes, Operations and Notifications that form part of a
class definition may only be omitted from an instance of that class
if they are defined as features of a Conditional Package and are
omitted in accordance with the condition defined for that package.

- Class Naming. The managed object class definition must include a
  class name which may be used to refer to the class in CMIP. This
  is achieved by registration of an Object Identifier value which
  corresponds to the class.

A1.3.2 Template Structure

```
<class-label> MANAGED OBJECT CLASS

        [DERIVED FROM    <class-label>
              [, <class-label>]*;
        ]
        [ALLOMORPHIC SET        <class-label>
        [, < class-label > ] *;
        ]
  [CHARACTERIZED BY <package-label>[,<package-label>]*;
  ]
  [CONDITIONAL PACKAGES <package-label> PRESENT IF
                                  <condition-definition>
                                  [,<package-label> PRESENT IF
                                   <condition-definition>]*;
  ]
  [PARAMETERS <parameter-label> [,<parameter-label>]*;
  ]

  REGISTERED AS < object-identifier>;
```

A1.3.3 Supporting Definitions

DERIVED FROM <class-label>[,<class-label>]*

The DERIVED FROM clause shall be present in all managed object class
definitions other than "top". It is therefore the case that "top" is
a superclass of all classes other than itself.

The class-label identifies a managed object class from which the

managed object class has been derived; i.e., a managed object class
which is one of the object class's parents in the inheritance
hierarchy. As multiple inheritance is permitted, a managed object
class may have one or more parent classes.

The process of inheritance (specialization) requires all the
characteristics of the superclass(es) other than DERIVED FROM and
ALLOMORPHIC SET to be included in the definition of the subclass. If
it is intended that the subclass be allomorphic, the definition of
the subclass shall include a ALLOMORPHIC SET clause that explicitly
defines the set of classes that the subclass is allomorphic to.

Where multiple inheritance results in the same element definition
being multiply imported (as could happen, for example, if two parent
superclasses contain the same attribute), the subclass is assumed to
contain a single copy of the definition concerned.

Characteristics that are inherited from a superclass shall not be
repeated in the documentation of the subclass unless one of the
techniques described in ISO/IEC 10165-4 for extending or modifying an
element of the superclass is being used. The DERIVED FROM clause is
therefore presumed to automatically import all inheritable elements
of definition from the superclass definition(s). There may be
augmented or modified by elements defined within the CHARACTERISED
BY, CONDITIONAL PACKAGES, and PARAMETER constructs.

ALLOMORPHIC SET <class-label>[,<class-label>]*

The Allomorphic Set allows a set of superclasses to be identified
that are "backwards compatible" with the managed object class. Thus,
if managed object class A identifies classes B and C as members of
its allomorphic set, it is possible to operate on an instance of
class A as if it were an instance of class B or C. The definition of
allomorphic forms allows, for example, the definition of enhanced
versions of a managed object class that are backwards compatible with
previous versions. The class label shall identify the class-label of
a managed object class definition that is a superclass of the managed
object class that is being defined.

CHARACTERIZED BY <package-label>[,<package-label]*

This construct, if present, allows one or more mandatory packages of
behaviour, attributes, operations, and notifications to be included
in the managed object class definition, in addition to those that are
present as a result of the Derived From construct.

CONDITIONAL PACKAGES <package-label> PRESENT IF <condition-
definition>[, <package-label> PRESENT IF <condition-definition>]*

Present if one or more conditional packages are to be included in the
managed object class definition. The package-label identifies the
package definition that is applicable. The condition-definition is a
description of the condition that, if true, requires that the package
be included in an instance of the class. The condition shall refer
to a documented feature of the definition of the underlying resource
or a related standard which is permitted to be present or absent in
an implementation.

PARAMETERS <parameter-label>[,<parameter-label>]*

If present, this construct permits the definition of parameters to be
included in the object class definition in addition to any inherited
parameters.

REGISTERED AS <object-identifier>

The object identifier value provides a globally unique identifier for
the object class definition. This value is used in the management
protocol when it is necessary to identify the object class.

A1.4 Package Template

This template allows a package of behaviour definitions, attributes,
operations and notifications to be defined for subsequent insertion
into a managed object class template under the Characterized By and
Conditional Packages constructs. The major elements of the
definition are:

- Behaviour. The managed object class definition provides a
  complete specification of the behaviour of the object. This
  includes:

  - The effect of the Operations upon the managed object;

  - Any constraints that are placed on operations in order to
    satisfy consistency rules, and in particular, the rules under
    which Creation and Deletion of managed objects may be
    performed and the consequences of these operations;

  - A specification of how instances of a managed object class
    interact with each other, related, managed objects of the same
    or different classes.

  - A complete definition of any other aspects of the
    behaviour of the managed object class.

  - Contained attributes. The set of attributes that the

package contains must be defined;

- Operations and Notifications. The package definition
  specifies which notifications instances of the class that make
  use of this package shall be able to generate, which operations
  instance of the class shall be capable of performing, and in the
  case of attribute related operations, which attributes shall be
  available to be operated upon.

Note: The operations identified in the package definition are the
operation types defined in the Information Model (Get, Replace, Set
to Default,...etc). In the case of Actions and Notifications,
further definitions are required in order to characterise their
functionality, as described in clauses A1.10, A1.11. The create and
delete operations are defined as part of the Name Binding template
described in clause A1.6, a creation and deletion of an object is
closely bound to the containment relationship between superior and
subordinate objects, rather than to all instances of a managed object
class.

A1.4.1 Template Structure

< package-label> PACKAGE

[BEHAVIOUR DEFINITIONS < behaviour-definition-label>
        [, < behaviour-definition-label >] *;
]
[ATTRIBUTES<attribute-label> <propertylist>[<parameter-label>]
        [,< attribute-label> < propertylist> [< parameter-label>]*]*;
]
[ATTRIBUTE GROUPS   <group-label> [<attribute-label>]*
                          [, < group-label > [< attribute-label >]*]*;
]
[ACTIONS   <action-label>[<parameter-label>]*
                          [, < action-label > [< parameter-label >]*]*;
]
[NOTIFICATIONS   <notification-label> [<parameter-label>]*
                  [, < notification-label > [< parameter-label >]*]*;
]
[REGISTERED AS < object-identifier>];

supporting productions

propertylist -> [REPLACE WITH DEFAULT]
        [DEFAULT VALUE < value-definition >]
        [PERMITTED VALUES <value-set-syntax-label>]
        [REQUIRED VALUES   <value-set-syntax-label>]
        [GET | REPLACE | GET-REPLACE]

          [ADD | REMOVE | ADD-REMOVE]

A1.4.2 Supporting Definitions

     BEHAVIOUR DEFINITIONS < definition-label> [,<definition-label>]*

     Behaviour Definitions allow the behaviour (semantics) of the managed
     object class to be completely described. These definitions relate
     the external view of the object (its operations and notifications) to
     its internal operation. The definition-label identifies an instance
     of use of the Behaviour template.

     Note: It should not be assumed that the behaviour defined by this
     clause is testable using existing conformance test technology.

     ATTRIBUTES. <attribute-label><propertylist> [<parameter-label>]*
          [,<attribute-label><propertylist> [<parameter-label>]*]*

     This allows attributes to be included in the package definition. The
     propertylist that follows each attribute label defines the set of
     operations that may be performed on the managed object with reference
     to the attribute, and defines any default, permitted or required
     value(s) associated with the attribute. The Replace With Default
     property is included if the property has a default value that may be
     set by means of the Replace With Default Value operation. The
     value-definition used to specify the default value shall be a value
     reference name, using the Externaltypereference notation defined in
     ISO 8824.

     If the PERMITTED VALUES property is present, the value-set- syntax-
     label specifies any restrictions on the possible values that the
     attribute may take. The value-set-syntax-label shall be a type
     reference name, using the ExternaltypeReference notation defined in
     ISO 8824. The form of the specification referenced shall be a
     subtype of the attribute syntax type, defined using the ASN.1 subtype
     notation.

     Note: The Permitted Values construct is required only in attribute
     definitions where it is necessary to specify a restriction on the
     value set permitted by the specification of the Attribute Syntax,
     e.g., when modifying an existing attribute specification.

     If the REQUIRED VALUES property is present, the value-set- syntax-
     label specifies any restrictions on the possible values that the
     attribute may take. The value-set-syntax-label shall be a type
     reference name, using the ExternaltypeReference notation defined in
     ISO 8824. The form of the specification referenced shall be a
     subtype of the attribute syntax type, defined using the ASN.1 subtype

notation.

Note: This property defines the value set required for conformance.

The parameter-labels allow parameters to be associated with the
operations permitted on the attribute.

ATTRIBUTE GROUPS <group-label> [<attribute-label>]*[,<group-label>
[<attribute-label>]*]*

This allows a set of attributes groups to be identified as part of
the package. The original definition of an attribute group may be
extended by the addition of further attribute-labels.

ACTIONS <action-label>[<parameter-label>]*
        [,<action-label>[<parameter-label>]*]*

If present, the action-labels identify the set of action definitions
that are included in the package. The behaviour definitions shall
specify the effect of these Actions upon managed objects.

The parameter-labels allow parameters to be associated with the
Action.

NOTIFICATIONS <notification-label>[<parameter-label>]*
                [,<notification-label>[<parameter-label>]*]*

Present if any Notifications are included in the package. The
notification labels identify the Notification definitions that are
applicable. The behaviour definitions shall specify the
circumstances under which those Notifications are generated by a
managed object.

The parameter-labels allow parameters to be associated with the
Notification.

REGISTERED AS <object-identifier>

The object identifier value, if present, provides a globally unique
identifier for the package definition, and registers the groupings of
behaviour definitions, attributes, attribute groups, actions and
notifications that the package defines. This value is required in
cases where the package is referenced by a conditional packages
construct in an object class template and the package contains a
behaviour construct or more than one element, in which case the value
of the object identifier is included in the conditional packages
attribute of any instances of the object class that are created with
the package present.

A1.5 Parameter Template

   This template permits the specification and registration of parameter
   syntaxes and associated behaviour that may be associated with
   particular operations and notifications within the managed object
   class and conditional package templates defined in clauses A1.3 and
   A1.4. This mechanism is not used in the OIM MIB-II.

A1.6 Name Binding Template

A1.6.1 Overview

   This template allows alternative naming structures to be defined for
   managed objects of a given managed object class by means of name
   bindings. The name binding allows an attribute to be selected as the
   naming attribute for a given subordinate class/superior class pair.

   The attribute selected must be an attribute of the subordinate object
   that is present in all instances of the specified managed object
   class. This attribute is used for the purpose of constructing the
   relative distinguished name (RDN) of subordinate objects of that
   class. An RDN is constructed from the OBJECT IDENTIFIER assigned to
   that attribute type and the value of the instance of the attribute.
   The Distinguished Name of the subordinate object is obtained by
   appending its RDN to the Distinguished Name of its superior object.

   Name bindings are not considered to be part of the definition of
   either of the managed object classes that they reference. A given
   managed object class may have more than one name binding associated
   with it. The set of name bindings defines the set of possible naming
   relationships with superior managed objects and the set of managed
   object classes from which subordinate object classes may be
   instantiated.

   Note: Name Bindings may be specified for a managed object class
   subsequent to the specification of the managed object class itself.

A1.6.2 Template Structure

   <name-binding-label> NAME BINDING

   SUBORDINATE OBJECT CLASS <class-label>;
   NAMED BY
   SUPERIOR OBJECT CLASS   <class-label>;
   WITH ATTRIBUTE   <attribute-label>;
   [BEHAVIOUR   <behaviour-definition-label>
                                   [,<behaviour-definition-label>]*;
   ]

```
   [CREATE [<create-modifier>[,<create-modifier>]]
                                        [<parameter-label>]*;
   ]
   [DELETE                 <delete-modifier>[<parameter-label>]*;
   ]
```

   REGISTERED AS < object-identifier>;

   supporting productions

   create-modifier->   with-reference-object |
                          with-automatic-instance-naming

   delete-modifier ->   only-if-no-contained-objects |
                                 deletes-contained-objects

A1.6.3 Supporting Definitions

   SUBORDINATE OBJECT CLASS <class-label>

   This defines the managed object class being named. The name of an
   instance of the subordinate managed object class is constructed by
   concatenating the distinguished name of the superior managed object
   class with the relative distinguished name of the subordinate managed
   object class.

   NAMED BY SUPERIOR OBJECT CLASS <class-label>

   This defines a managed object class that may contain instances of the
   subordinate managed object class.

   WITH ATTRIBUTE <attribute label>

   This defines the attribute that shall be used, in the context of this
   name binding, to construct the relative distinguished name for the
   subordinate managed object class. Values of this attribute shall be
   represented by single-valued data types, complying with the
   restrictions specified in ISO 10165-1; if no naturally suitable
   attribute is available for use as a naming attribute, object
   designers are encouraged to provide a naming attribute of type
   Graphic String.

   BEHAVIOUR <behaviour-definition-label>

   If present, this clause permits any behavioural impact imposed as a
   consequence of the name binding to be defined. The behaviour-
   definition-label identifies the constraint definition concerned; the
   constraint is therefore expressed using the behaviour template.

CREATE [<create-modifier>[,<create-modifier>]][<parameter-label>]*

Present if it is permitted to create new instances of the managed
object class referenced by the SUBORDINATE OBJECT CLASS construct in
the context of this name binding, by means of systems management
operations. The create-modifier values specify the options available
on creation. The permitted create- modifiers are as follows:

   - with reference-object. If present, a reference object may be
     specified on creation as a source of default values and to
     specify choice of Conditional Packages;

   - with automatic instance naming. If present, the CREATE
     request may omit to specify the instance name of the new
     object instance. The behaviour definitions shall specify what
     course of action is taken when there is a choice of Name Forms
     that may be applied to the new object instance.

If neither create-modifier is specified, the instance name and all
necessary default values shall be specified in the CREATE request.

The parameter-labels allows parameters to be associated with the
CREATE operation.

DELETE <delete-modifier> [<parameter-label>]*

Present if it is permitted to delete instances of the managed object
class referenced by the SUBORDINATE OBJECT CLASS construct in the
context of this name binding. The delete- modifier indicates the
behaviour of a managed object of that class if the managed object is
deleted. The permitted delete-modifiers are:

   - only-if-no-contained-objects. If specified, any contained
     managed objects must be explicitly deleted prior to deletion of
     the containing managed object, i.e., a DELETE request will cause an
     error if there are contained managed objects;

   - delete-contained-objects. If specified, a DELETE request
     applied to an instance of the managed object class has the effect
     of deleting contained objects also.

If there are constraints on deletion relative to other relationships
or conditions, these are specified as part of the behaviour of the
managed object class.

The parameter-labels allows parameters to be associated with the
DELETE operation.

A1.7 Attribute Template

A1.7.1 Overview

   This template is used to define individual attribute types. These
   definitions may be further combined by the Attribute Group template
   where attribute groups are required. The major elements of the
   definition are:

      - Derivation. The definition of an attribute type may modify or
        constrain the definition of another attribute type.

      - Attribute Syntax. The definition of an attribute type must
        include the definition of the syntax that will be used to convey
        values of the attribute in CMIP. This definition is achieved by
        means of a reference to an ASN.l Type Definition. The
        definition of an attribute syntax indicates whether the
        attribute value is a single or set-valued attribute type. If
        the base type is SET OF, the attribute is a set-valued type,
        otherwise it is a single-valued type;

      - Value Matching. The definition of an attribute type may
        include the valid ways in which the value of an instance of the
        type may be tested, i.e., whether the attribute may be tested
        for equality, magnitude, etc.. Value matching on some attribute
        types may require the specification of how a matching rule is
        defined to operate, as part of the attribute's behaviour
        definition. The absence of any matching rules in the attribute
        definition implies that matching of values is undefined;

      - Behaviour. The attribute definition may include definition of
        attribute specific behaviour; i.e., behaviour that applies to an
        attribute type regardless of which managed object class contains
        instances of tho attribute type;

      - Attribute type name. An Object Identifier value shall be
        allocated to each attribute that is to be included in the
        definition of a managed object class. This value is used in CMIP
        to identify the attribute.

A1.7.2 Template Structure

        <attribute-label> ATTRIBUTE

        derived-from-construct | with attribute-syntax-construct
        [MATCHES FOR   <qualifier>[,<qualifier>]* ;]
        [BEHAVIOUR   <behaviour-definition-label>
                         [,<behaviour-definition-label>]* ;

```
    ]
    [PARAMETERS      <parameter-label>
                     [,<parameter-label>]*
    ]

    [REGISTERED AS < object-identifier> ;]

    supporting productions

    qualifier -> Equality | Ordering | Substrings
                 | Set Comparison | Set Intersection

    derived-from-construct -> DERIVED FROM  <attribute-label > ;

    with-attribute-syntax-construct -> WITH ATTRIBUTE SYNTAX
                                       <syntax-label> ;
```

A1.7.3 Supporting Definitions

    DERIVED FROM <attribute-label>

    If this element is present, the attribute definition takes as a
    starting point all aspects of the definition referenced by
    attribute-label, including any that it may in turn have derived from
    other attribute definitions. The rules for interpreting the effect
    of presence of any of the other elements of the attribute template
    under these circumstances are as follows:

       - WITH ATTRIBUTE SYNTAX is not permitted to be present. The
         attribute syntax shall be the attribute syntax of the attribute
         from which this attribute has been derived.;

       - MATCHES FOR: The resultant set of matching rules shall be the
         logical OR of the matching rules specified by this construct
         with any derived matching rules;

       - BEHAVIOUR is assumed to extend any derived behaviour
         definitions;

       - REGISTERED AS is assumed to replace any registration derived
         from other definitions.

    This derivation mechanism permits:

       - The definition of an attribute based on an existing attribute
         definition;

       - The addition of further constraints to an existing attribute

        definition.

    WITH ATTRIBUTE SYNTAX <syntax label>

    This element, present only if the DERIVED FROM construct is absent,
    identifies the ASN.1 data type that describes how instances of the
    attribute value are carried in protocol. The syntax-label shall be
    an ASN.l Externaltypereference.

    The ASN.1 data type also defines the data type of the attribute
    itself. If the base type of the syntax is set-of, the attribute is a
    set-valued attribute. All other ASN.l data types, including set,
    sequence of, and sequence type, define single-valued attribute types.

    MATCHES FOR <qualifier>[,<qualifier>]*

    This element defines the types of test that may be applied to a value
    of the attribute as part of a Filter operation. Matching for the
    presence of an attribute is implicitly permitted for all attributes.
    For all other types of matching, if this clause is not present,
    matching is undefined and is therefore not permitted on the
    attribute. The options are:

        - Equality. If present, the attribute value may be tested for
          equality against a given value;

        - Ordering. If present, the attribute value may be tested against
          a given value in order to determine which has the greater
          value;

        - Substrings. If present, the attribute value may be tested
          against a given substring value in order to determine its
          presence or absence in the attribute value;

        - Set Comparison. If present, the attribute value may be tested
          against a given value in order to determine set equality or
          subset/superset relationships between the values;

        - Set Intersection. If present, the attribute value may be tested
          against a given value in order to determine the presence or
          absence of a non-null set intersection between the two values.

    BEHAVIOUR <behaviour-definition-label>
                 [,<behaviour-definition-label>]*

    Any behaviour that is generic to this attribute type may be defined
    by means of this behaviour clause. The behaviour definition shall
    include any additional specification that is required in order to

define how the chosen set of matching rules are applied to the
attribute definition. Behaviour that is specific to the managed
object class is defined in the behaviour clause of the managed object
class template.

Note: It should not be assumed that the behaviour defined by this
clause is testable using existing conformance test technology.

REGISTERED AS <object-identifier>

If present, the object-identifier provides a globally unique
identifier for the attribute definition; this includes all elements
referenced either directly, or indirectly by the Derived From, With
Attribute Syntax, Matches For and Behaviour constructs, where
present. This value is used in the management protocol when it is
necessary to identify the attribute type. If this construct is
omitted, the attribute definition shall not be referenced in a
managed object class definition without further constraints being
applied in another attribute definition.

A1.8 Notification Template

A1.8.1 Overview

This template is used to define the behaviour and syntax associated
with a particular Notification type. The main features of the
definition are as follows:

   - Behaviour. The definition of a Notification type must specify
     the circumstances under which a notification of the type is
     generated;

   - Mode of operation. The definition of a notification type shall
     indicate whether the notification may be confirmed,
     unconfirmed or both;

   - Syntax Definitions. The definition of the Notification type
     must specify any syntax that will be used to convey the CMIS
     Event Information and Event Reply parameters in CMIP. These
     syntaxes are defined by means of ASN.1 data types. The
     template also permits the allocation of attribute values to
     fields in the syntax;

   - Notification naming. The value of the Object Identifier
     associated with the Notification definition is used to identify
     the Event type in CMIP.

A1.8.2 Template Structure

      <notification-label> NOTIFICATION

      BEHAVIOUR   <behaviour-definition-label>
                                      [,<behaviour-definition-label>]*;
      MODE   <confirmation-mode>;
      [PARAMETERS   <parameter-label>[,<parameter-label]*;
      ]
      [WITH INFORMATION SYNTAX <syntax-label>
                      [AND ATTRIBUTE IDS <field-name><attribute-label>
                                      [,<field-name><attribute-label]*];
      ]
      [WITH REPLY SYNTAX <syntax-label>;]

      REGISTERED AS <object-identifier>;

      supporting productions

      confirmation-mode -> CONFIRMED | NON-CONFIRMED | CONFIRMED AND
      NON-CONFIRMED

A1.8.3 Supporting Definitions

   BEHAVIOUR <behaviour-definition-label>

   This defines the behaviour of the notification, the data that must be
   specified with the notification, the results that the notification
   may generate and their meaning. The behaviour- definition-label
   references a behaviour description defined by use of the Behaviour
   Template.

   Note: It should not be assumed that the behaviour defined by this
   clause is testable using existing conformance test technology.

   MODE CONFIRMED | NON-CONFIRMED | CONFIRMED AND NON-CONFIRMED

   This defines the allowable mode of operation of the notification
   type, as follows:

      - CONFIRMED: The notification type shall operate in the
           confirmed mode only;

      - NON-CONFIRMED: The notification type shall operate in the
           non-confirmed mode only;

      - CONFIRMED AND NON-CONFIRMED: The notification type may
           operate in either confirmed mode or non-confirmed mode.

PARAMETERS <parameter-label>[,<parameter-label>]*

The parameter-labels allow parameters to be associated with with the
behaviour of the attribute type.

WITH INFORMATION SYNTAX <syntax-label> [AND ATTRIBUTE IDS
<field-name><attribute-label> [,<field-name><attribute-label>]*;]

This construct identifies the ASN.l data type that describes the
structure of the notification information that is carried in
management protocol, and permits the association of attribute
identifiers with named fields in the abstract syntax. The syntax-
label shall be a type reference name, using the Externaltypereference
notation defined in ISO 8824. If absent, there is no information
associated with the notification invocation. If the AND ATTRIBUTE
IDS option is used, the field- name label shall be a label defined
within the abstract syntax referenced by the syntax that appears in
the construct. The data type that is labeled by the field-name is
used to carry values of the attribute referenced by attribute-label.
The ASN.1 data type of the attribute shall be the same as the data
type referenced by field-name.

WITH REPLY SYNTAX <syntax-label>

If a syntax-label is present, this identifies the ASN.1 data type
that describes the structure of the notification reply that is
carried in management protocol. The syntax-label shall be a type
reference name, using the Externaltypereference notation defined in
ISO 8824. If absent, there is no information associated with the
notification reply.

Appendix 2: New Objects: Internet SMI Object Type Macros

```
atEntryId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
   "<index>.1.<address>, where <index is the decimal representation
    of atIfIndex and <address> is atNetAddress represented in
    dot notation.  The 1 is a subidentifier indicating that an IP
    address follows. "
              ::= { attributes 3}


atTableId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
               "An empty string."
              ::= { attributes 2}



egpId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
               "An empty string."
              ::= { attributes 16}


egpNeighEntryId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
                    "egpNeighAddr encoded in dot notation."
              ::= { attributes 22}


egpNeighTableId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
               "An empty string."
              ::= { attributes 17}


icmpId OBJECT-TYPE
              SYNTAX  DisplayString
```

```
                    ACCESS  read-only
                    STATUS  mandatory
                   DESCRIPTION
                     "An empty string."
                   ::= { attributes 9}

   ifId OBJECT-TYPE
                   SYNTAX  DisplayString
                   ACCESS  read-only
                   STATUS  mandatory
                   DESCRIPTION
                     "An empty string."
                   ::= { attributes 23}

   ifEntryId OBJECT-TYPE
                   SYNTAX  DisplayString
                   ACCESS  read-only
                   STATUS  mandatory
                   DESCRIPTION
                     "The decimal representation of ifIndex."
                   ::= { attributes  21}

   ifTableId OBJECT-TYPE
                   SYNTAX  DisplayString
                   ACCESS  read-only
                   STATUS  mandatory
                   DESCRIPTION
                     "An empty string."
                   ::= { attributes 1}

   ipAdEntryId OBJECT-TYPE
                   SYNTAX  DisplayString
                   ACCESS  read-only
                   STATUS  mandatory
                   DESCRIPTION
                     "ipAdEntAddr encoded in dot notation."
                   ::= { attributes 19}

   ipAddrTableId OBJECT-TYPE
                   SYNTAX  DisplayString
                   ACCESS  read-only
                   STATUS  mandatory
                   DESCRIPTION
                     "An empty string."
                   ::= { attributes 5}
```

```
ipId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
              "An empty string."
           ::= { attributes 4}


ipNetToMediaEntryId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
    "<interface>.<address>, where <interface> is the decimal
    representation of ipNetToMediaIndex and <address> is
    ipNetToMediaNetAddress in dot notation."
           ::= { attributes 8}


ipNetToMediaTableId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
              "An empty string."
           ::= { attributes 7}


ipRouteEntryId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
        "<dest> , where <dest> is ipRouteDest represented in dot
        notation."
           ::= { attributes 20}


ipRoutingTableId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
              "An empty string."
           ::= { attributes 6}


snmpId OBJECT-TYPE
            SYNTAX  DisplayString
            ACCESS  read-only
            STATUS  mandatory
           DESCRIPTION
```

```
                     "An empty string."
                 ::= { attributes 18}


   tcpConnId OBJECT-TYPE
                 SYNTAX  DisplayString
                 ACCESS  read-only
                 STATUS  mandatory
               DESCRIPTION
           "<laddr> . <lport> . <raddr> . <rport>, where
           <laddr> is the dot notation representation of
                 tcpConnLocalAddress,
           <lport> is the decimal representation of tcpConnLocalPort,
           <raddr> is the dot notation representation of
                 tcpConnRemAddress, and
           <rport> is the decimal representation of tcpConnRemPort."
                 ::= { attributes 12}


   tcpConnTableId OBJECT-TYPE
                 SYNTAX  DisplayString
                 ACCESS  read-only
                 STATUS  mandatory
               DESCRIPTION
                  "An empty string."
                 ::= { attributes 11}


   tcpId OBJECT-TYPE
                 SYNTAX  DisplayString
                 ACCESS  read-only
                 STATUS  mandatory
               DESCRIPTION
                  "An empty string."
                 ::= { attributes 10}


   updEntryId OBJECT-TYPE
                 SYNTAX  DisplayString
                 ACCESS  read-only
                 STATUS  mandatory
               DESCRIPTION
           "<index> . <address>, where <index> is the decimal
           representation of atIfIndex and <address> is atNetAddress
           represented in dot notation."
                 ::= { attributes 15}


   udpId OBJECT-TYPE
                 SYNTAX  DisplayString
                 ACCESS  read-only
                 STATUS  mandatory
               DESCRIPTION
```

```
                "An empty string."
              ::= { attributes 13}


   udpTableId OBJECT-TYPE
              SYNTAX  DisplayString
              ACCESS  read-only
              STATUS  mandatory
            DESCRIPTION
                "An empty string."
              ::= { attributes 14}
```

Appendix 3: Supporting Definitions

   The definition of the object class "top", from which all object
   classes are derived, is taken from ISO/IEC DIS 10165-2: Definition of
   Management Information [5]. However, pending progression of that
   document to an International Standard, the object class "top" and its
   associated attributes have been registered here under the oim
   registration arc.

   top   MANAGED OBJECT CLASS
           CHARACTERIZED BY
               topPackage PACKAGE
           BEHAVIOUR DEFINITIONS topBehaviour;
           ATTRIBUTES
                   objectClass   GET,
                   name   GET,
                   allomorphs   GET,
                   nameBindings   GET,
                   packages   GET;;;
   REGISTERED AS {objects 1};

   topBehaviour   BEHAVIOUR
           DEFINED AS
   This is the top level of managed object class hierarchy and
   every other managed object class is a specialization of either
   this generic class (top) or a specialization of a subclass of top.;

   allomorphs ATTRIBUTE
           WITH ATTRIBUTE SYNTAX
                   Top-Syntax.Allomorphs;
                   -- A set of allormorphic superclass identifiers
           MATCHES FOR Set Comparison, Set Intersection;
   REGISTERED AS {attributes 30};

   name ATTRIBUTE
           WITH ATTRIBUTE SYNTAX InformationFramework.
                                   RelativeDistinguishedName;
                   -- defined in Directory standards
           MATCHES FOR Equality;
   REGISTERED AS {attributes 31};

   nameBindings ATTRIBUTE
           WITH ATTRIBUTE SYNTAX
                   Top-Syntax.NameBindings;
                   -- A set of valid namebindings for this object.
           MATCHES FOR   Set Comparison, Set Intersection;
   REGISTERED AS {attributes 32};

```
objectClass     ATTRIBUTE
        WITH ATTRIBUTE SYNTAX CMIP-1.ObjectClass;
        MATCHES FOR Equality, Ordering;
REGISTERED AS {attributes 33};

packages ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
                Top-Syntax.Packages;
                -- The set of optional packages defined for the
                -- class that are included in this instantiation of
                -- the object.
        MATCHES FOR Set Comparison, Set Intersection;
REGISTERED AS {attributes 34};


Top-Syntax {iso(1) org(3) dod(6) internet(1) mgmt(1) mib(1)
            oim(9) misc(4) 2}
DEFINITIONS ::=
BEGIN

-- from ISO/IEC DIS 10165-2:Definition of Management Information

Allomorphs ::=  SET OF OBJECT IDENTIFIER

NameBindings ::=        SET OF OBJECT IDENTIFIER

Packages ::=    SET OF OBJECT IDENTIFIER

-- From Directory InformationFramework

RelativeDistinguishedName ::= SET OF AttributeValueAssertion

AttributeValueAssertion ::= SEQUENCE {AttributeType,
                                      AttributeValue}

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY

-- From CMIP-1

ObjectClass ::= CHOICE {globalForm     [0] OBJECT IDENTIFIER,
                                localForm    [1] INTEGER }
END
```

Security Considerations

   Security issues are not discussed in this memo.

Author's Address

    Lee LaBarre
    The MITRE Corporation
    Burlington Road
    Bedford, MA 01730

    Phone: (617) 271-8507

    EMail: cel@mbunix.mitre.org