ISO/CCITT and Internet Management Coexistence (IIMC):

Translation of Internet MIBs to ISO/CCITT GDMO MIBs

(IIMCIMIBTRANS)

February, 1994

Lee LaBarre (Editor)

The MITRE Corporation
Burlington Road
Bedford, MA 01730
cel@mbunix.mitre.org

Status of this Memo

This document provides information to the network and systems management community. This document is intended as a contribution to ongoing work in the area of multi-protocol management coexistence and interworking. This document is part of a package; see also [IIMCOMIBTRANS] [IIMCMIB-II] [IIMCPROXY] and [IIMCSEC]. Distribution of this document is unlimited. Comments should be sent to the Network Management Forum IIMC working group (iimc@thumper.bellcore.com).

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the 1id-abstracts.txt listing contained in the internet-drafts Shadow Directories on ds.internic.net, nic.nordu.net, ftp.nisc.sri.com, munnari.oz.au to learn the current status of any Internet Draft.

Abstract

This document is intended to facilitate the multi-protocol management coexistence and interworking for networks that are managed using the ISO/CCITT Common Management Information Protocol (CMIP) and networks that are managed using the Internet Simple Network Management Protocol (SNMP). This document contains translation and registration procedures that are applicable to translation of Internet MIBs, defined according to the Internet Structure of Management Information (SMI), into ISO/CCITT SMI-defined MIBs. This document also defines and registers generic management information that may be used with the translation procedures to facilitate interoperability.

Table of Contents

List of Figures

List of Tables

REVISION HISTORY

Issue 1.0, October 1993

This is the first issue of this document. The internet draft <draft-labarre-internetmib-iso-04>, dated February, 1994, is identical in content to Issue 1.0, October 1993. It has been reformatted for posting as an internet draft.

1. INTRODUCTION

This section provides an overview of ISO/CCITT and Internet Management Coexistence (IIMC) activities, insight into the problem being addressed by IIMC, and a brief introduction to the strategy adopted by IIMC: use of translated MIBs in either a proxy or native implementation. The section concludes by describing the scope of this document, and terms and conventions used by this document.

1.1 PROBLEM STATEMENT

The need for enterprise network management has been addressed by development of network management standards within various communities, most notably the ISO/CCITT and Internet communities.

- The ISO/CCITT community developed the Common Management Information Protocol (CMIP) [5], and related SMI documents [7,8,9].
- The Internet community developed the Simple Network Management Protocol (SNMP) [12], and its successor, SNMPv2 [19]. The Internet SMI is defined in [11] and [17].

These standards share a nearly common management model, but diverge due to differing management philosophies. Although functionally similar, the Internet and ISO/CCITT protocols and SMIs differ in terms of their complexity and specific operations. Business requirements for end-to-end enterprise management include the need to integrate the management of many different devices, potentially owned or administered by many independent organizations. This requires components to be accessed by ISO/CCITT management, Internet management, and proprietary management mechanisms in a manner which presents a unified view of the network, despite protocol and SMI differences.

For example, many telecommunications and computer vendors, represented by organizations such as the Network Management Forum (NMF), and the U.S. government, as specified in the Government Network Management Profile (GNMP) Version 1.0 [27], have based their enterprise management model on the ISO/CCITT management model. These organizations are particularly interested in integrated management of devices that use the Internet management. This interest is primarily due to the widespread commercial implementation and use of such devices, especially devices that use the Internet TCP/IP protocol suite.

1.2 OVERVIEW OF IIMC

The ISO/CCITT and Internet Management Coexistence (IIMC) package includes the following documents.

IIMCIMIBTRANS	Translation of Internet MIBs to ISO/CCITT GDMO MIBs
IIMCOMIBTRANS	Translation of ISO/CCITT GDMO MIBs to Internet MIBs [25]
IIMCMIB-II	Translation of Internet MIB-II (RFC 1213) to ISO/CCITT GDMO MIB [22]

IIMCPROXY	ISO/CCITT to Internet Management Proxy [23]
IIMCSEC	ISO/CCITT to Internet Management Security [24]

These documents together comprise a package aimed at integrating ISO/CCITT-based and Internet-based management systems.

IIMC specifications address the problem that end-to-end management requires an integrated, unified view of the managed network, despite differences in management protocol and information structure. Integrated management can be facilitated by the development of "proxy" mechanisms which translate between functionally equivalent service, protocol, and SMI differences to create this unified view. MIB translation procedures can be used to support proxy management, as well as to take advantage of existing MIB definition and avoid duplication of effort. In this way, commercial investment in both ISO/CCITT and Internet-based management technologies can be preserved through deployment of common methods and tools which support integration.

This overall strategy was outlined in a joint publication developed by the NM Forum and X/Open entitled "ISO/CCITT and Internet Management: Coexistence and Interworking Strategy" [26]. The documents included in the IIMC package are the next level of detailed specifications which implement several of the methodologies identified in the strategy. Additional specifications may be defined in the future.

1.3 MIB TRANSLATION PROCEDURES

The foundation of IIMC is provided by a pair of Management Information Base (MIB) translation procedures.

- IIMCIMIBTRANS (this document) specifies translation procedures for converting MIBs from Internet MIB macro format into ISO/CCITT GDMO template format.

- IIMCOMIBTRANS [25] specifies translation procedures for converting MIBs from ISO/CCITT GDMO template format into Internet MIB macro format.

The IIMC approach is to specify direct translation procedures which yield a pair of functionally-equivalent MIBs, as shown in Figure 1.

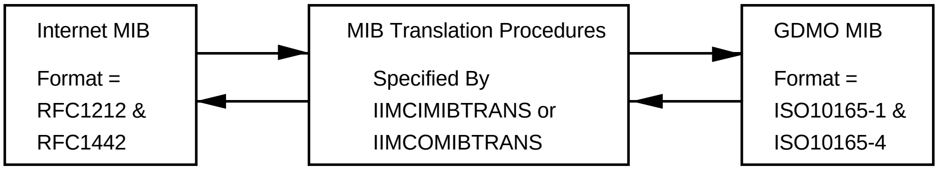

Figure 1. MIB translation.

MIBs translated by these procedures may be used to take advantage of existing MIB definitions when business needs require deployment in a different management environment. Translated MIBs may also be used to provide uniformity when multiple management environments are supported by a single system (e.g., dual stack managers). Finally, IIMC MIB translation procedures may be used to support service emulation by a proxy.

1.4 NATIVE MANAGEMENT MODEL

The basic model for ISO/CCITT and Internet management is illustrated in the following diagram.

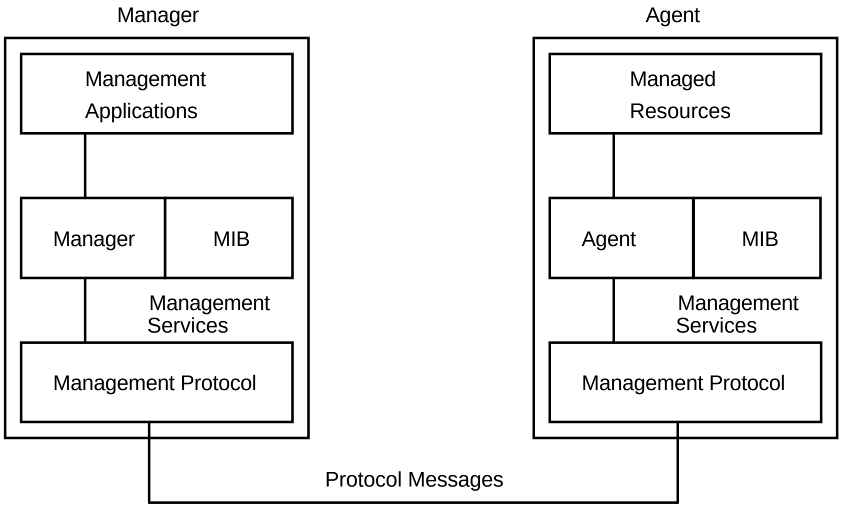

Figure 2. Native management.

Within IIMC documents, this model is referred to as the "native" management model. MIBs translated using IIMC procedures can be used by "native" agent implementations. For example, an ISO/CCITT agent can make visible TCP/IP managed resources using the translated GDMO version of the Internet MIB-II [14] specified by [22]. Dual-stack managers or agents may also be implemented which support both the original MIB and the translated MIB generated using IIMC-specified procedures.

1.5 PROXY MANAGEMENT MODEL

The basic model for ISO/CCITT to Internet proxy management is illustrated in the following diagram. This proxy is specified by [23]. A similar approach could also be taken to specify an Internet to ISO/CCITT proxy, although no such IIMC document is currently specified.

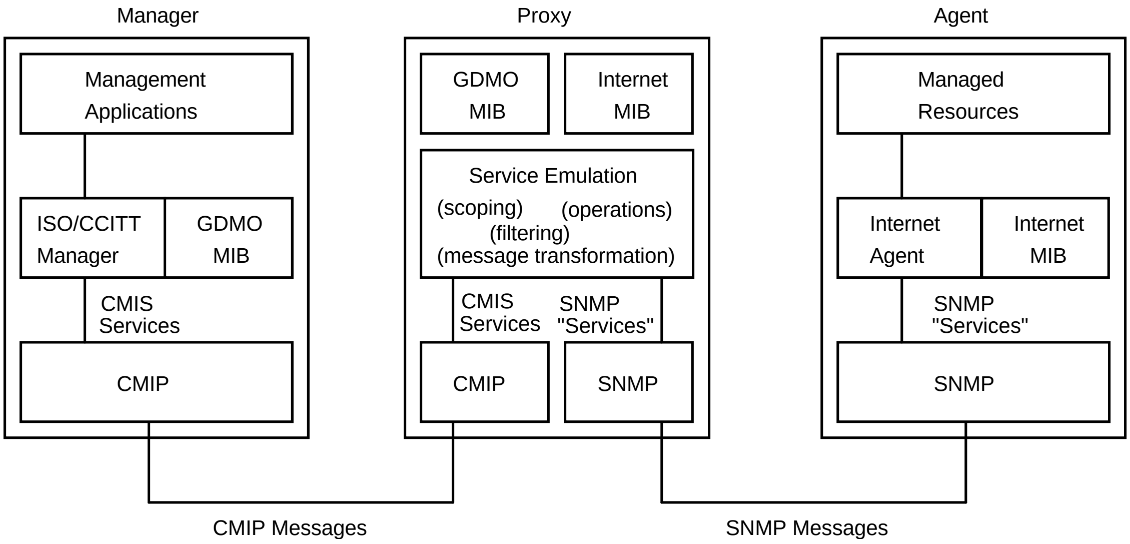

Figure 3. Proxy management.

This ISO/CCITT to Internet proxy provides emulation of CMIS services by mapping to the corresponding SNMP message(s) necessary to carry out the service request. The service emulation allows management of Internet objects by an ISO/CCITT manager. The left hand side of the proxy behaves like an ISO/CCITT agent, communicating with the ISO/CCITT manager using CMIP protocols. The right hand side of the proxy behaves like an Internet manager, communicating with the Internet agent using SNMP protocols.

The proxy relies on the existence of a pair of directly-related MIB definitions, where the Internet MIB has been translated into ISO/CCITT GDMO using the procedures specified in IIMCIMIBTRANS. The proxy uses these MIB definitions and rules to provide run-time translation of management information carried in service requests and responses.

The proxy is designed with a specified interface between the proxy and the underlying protocol stacks, and so deals primarily in terms of CMIS services and SNMP "services". The proxy emulates services such as CMIS scoping and filtering, processing of CMIP operations, and forwarding/logging of CMIS notifications by performing a mapping process which must be tailored for each protocol (for example, SNMPv1 and SNMPv2 are variants of the same protocol mapping process).

1.6 SCOPE OF THIS DOCUMENT

A major reason for the rapid commercialization of devices manageable via the Internet management protocol is due to the speed with which the vendors in the Internet community have been able to develop MIBs based on the Internet SMI. To capitalize on this continuing Internet MIB development and their deployment in commercial devices, communities interested in integrated management via ISO/CCITT-Internet proxies require that procedures be defined for translation of Internet MIBs into ISO/CCITT GDMO MIBs, i.e., MIBs defined according to the ISO/CCITT SMI Guidelines for Definition of Managed Objects [9]. Communities interested in using ISO/CCITT management protocols to directly manage resources using the Internet defined MIB elements are also interested in MIB translation procedures. Such MIB translations may also minimize the independent development of ISO/CCITT GDMO MIBs for the same resources and thereby reduce the incompatibilities with the Internet MIBs.

Translation procedures which may be automated to a high degree, and include unambiguous automated registration procedures, are of particular interest to the communities interested in using GDMO translations of Internet MIBs. This document (IIMCIMIBTRANS) defines such procedures.

This document also defines generic SNMP trap to CMIS notification mappings, common naming conventions, and ASN.1 modules applicable to translated Internet MIBs.

Many of the procedures defined in this document may be subject to automation. Comments are provided concerning possible aids to automation; however, it is not the intent of this document to provide fully automated translation algorithms.

1.7 TERMS AND CONVENTIONS

This document assumes that the reader is familiar with the ISO/CCITT SMI and Internet SMI, and the terminology of each. The term SNMP will be used throughout the document to indicate either SNMPv1 or SNMPv2, unless a distinction needs to be made.

Other conventions used during the translation process are described in Section 3.2.

2. REGISTRATION AND NAMING PROCEDURES

Registration and naming procedures are crucial to the unique identification of management information. Registration assures the uniqueness of management information element types, while naming provides a way of distinguishing instances of a type and locating them within the MIB.

2.1 REGISTRATION PROCEDURES

The registration authority for management definitions produced during the translation process shall be the Network Management Forum.

> *WARNING: Object Identifiers produced during the translation process are only provisionally assigned. The object identifiers are not registered until approved by the registration authority. In cases of conflict with existing registered object identifiers, the registration authority will manually assign object identifiers.*

Registration procedures specify that changes in the syntax or semantics of registered entities require them to be registered as new entities. The process of converting Internet MIBs into the ISO/CCITT GDMO MIBs inevitably introduces subtle semantic changes in how data can be operated on, and changes in the content of the MIB element. For example, ISO/CCITT attributes that are converted from Internet Object Types acquire matching rules for use in filtering operations. ISO/CCITT object classes that are created from Internet groups acquire semantics related to their inheritance of new attributes from the "top" managed object class. The end result is that all the new ISO/CCITT object classes, attributes, and notifications created during the translation process must be registered. In addition, name bindings for inserting object instances into the naming hierarchy must be registered.

2.1.1 Automated Registration Procedures

Registration procedures are critical to the goals of automating the translation of Internet MIBs to ISO/CCITT GDMO format, and the efficient implementation of ISO/CCITT-Internet proxies. Registration involves assignment of an ASN.1 Object Identifier (OID) to the entity. Management entities defined according to the principles of the Internet SMI may be registered under the IAB's "internet" arc, or registered under an arc in another organization's proprietary registration subtree.

Since OIDs can be guaranteed to be unique across organizations only within the context of the uppermost registration hierarchy, this document uses the entire OID to prevent ambiguity. The effect of the registration procedure specified in this document is to graft the entire OID to another part of the registration tree, without changing the OID structure.

This document allocates an arc in the registration hierarchy for use in automated registration of management elements defined according to IIMC procedures defined in this document. The arc is named "iimcAutoTrans", and is registered in section 4.2.

The following paragraphs describe IIMC registration procedures to facilitate automated translation and assure uniqueness of registered ASN.1 object identifiers for ISO/CCITT object classes and attributes derived from internet entities; and a registration procedure for their name bindings, document identifiers, and ASN.1 modules.

2.1.1.1 Translated Object Classes and Attributes Registration

Follow the procedure below for OIDs associated with Internet groups, conceptual tables, conceptual table entries, and objects.

a) Determine the sequence of sub-identifiers of the OID assigned to the internet management entity, beginning at the root of the registration tree, and identify that sequence as <internetEntityId>,

> *Note: Remember, the first part of the ASN.1 BER encoded OID must be translated into two sub-identifiers.*

b) Determine the translated OID {<iimcTransOID>} as:

{<iimcTransOID>} = {<iimcAutoObjAndAttr> <internetEntityId>}

where <iimcAutoObjAndAttr> is the OID dedicated for ISO/CCITT-Internet automated registration of translated object classes and attributes.

This procedure preserves the unique identification of the entities within the Internet subtree, and entities identified by OIDs that are registered by other organizations.

2.1.1.2 Translated NAME BINDINGs Registration

As described in section 2.2.2 , the ISO/CCITT SMI requires that managed object instances be bound into a naming hierarchy, or tree, for purposes of naming. ISO/CCITT NAME BINDING templates are used to register the manner in which such instances may be bound. These name bindings shall be registered.

The Internet SMI does not include the concept of a naming tree and name binding. Thus, there exists no registered internet entity from which an OID for the ISO/CCITT NAME BINDING template can be derived. One solution is to use the object class OID to register name bindings under a special registration arc {iimcAutoNameBinding}. The following procedure shall be followed for registration of a single name binding for an object class to be inserted into the naming hierarchy, i.e., the subordinate object class.

- Assign each new name binding an OID, using the following rules. Start with the OID for the subordinate object class, derived using the procedures in section 2.1.1.1. Within the class OID, extract the <internetEntityId>(c) portion of the OID. Prepend {iimcAutoNameBinding} to the <internetEntityId>(c) so that the OID for the name binding is of the form:

  {iimcAutoNameBinding <internetEntityId>(c)}

> *WARNING: Object Identifiers for name bindings produced during the translation process are only provisionally assigned. The object identifiers are not registered until approved by the registration authority. In cases of conflict with existing registered name bindings that have different creation or deletion semantics, or behaviour characteristics, the registration authority will manually assign object identifiers to the name binding.*

Note: *In general multiple name bindings may be defined for an ISO/CCITT object class. However the automated registration procedures defined in this document only provide for a single name binding. This facilitates the proxy translation process, especially for received traps/notifications and informRequests.*

2.1.1.3 Translated ASN.1 Modules and Documents Registration

Translated MIBs may contain definitions that can be referenced by other documents, e.g., attribute types defined in this document. An identifier and registered OID shall be given to each such document so that management elements defined therein may be referenced. Documents that contain translated MIBs derived from RFCs, unless manually assigned, shall be automatically assigned short document identifiers, i.e., a document alias, of the form:

"iimcRFC" || <rfc number> [|| <rfc number>]*,

where <rfc number> is the internet number assigned to the RFC. If multiple RFCs are included in the translation, then the RFC numbers shall be concatenated in ascending sequence. Document aliases may be manually assigned by the registration authority.

Note: *Multiple RFCs are included in the same translated MIB in exceptional cases, not as a general rule.*

Documents derived from MIBs defined in Internet RFCs are registered under the {iimcAutoDocument} arc by concatenating the RFC number onto that arc. If multiple RFCs are included in the translation, then the RFC numbers shall be concatenated to iimcAutoDocument in ascending sequence. Explicit registration of other IIMC documents shall be under the iimcDocument arc.

Documents that define translated MIBs contain syntax in ASN.1 modules that can be referenced by other documents. An identifier and registered OID shall be given to each such ASN.1 module so that syntax defined therein may be referenced. ASN.1 modules in documents that contain translated MIBs derived from RFCs, unless manually assigned, shall be automatically assigned short module identifiers, i.e., a module alias, of the form:

"IIMCRFC" || <rfc number> [|| <rfc number>]* || "ASN1",

where <rfc number> is the internet number assigned to the RFC. If multiple RFCs are included in the translation, then the RFC numbers shall be concatenated in ascending sequence. Module aliases may be manually assigned by the registration authority.

Modules derived for MIBs defined in Internet RFCs are registered under the {iimcAutoModule} arc by concatenating the RFC number onto that arc. If multiple RFCs are included in the translation, then the RFC numbers shall be concatenated to iimcAutoModule in ascending sequence. Explicit registration of other IIMC ASN.1 modules shall be under the iimcModule arc.

Translated MIBs defined according to the procedures described in this document shall be documented in the following format.

- The OID of the document shall be specified first as

  <document alias> OBJECT IDENTIFIER ::= {<document OID>}

where <document alias> is the short document identifier derived as described previously, e.g., iimcRFC12131354, and <document OID> is the OID assigned to the document using the procedures described in this clause, e.g., {iimcAutoDocument 1213 1354}.

- The order in which definitions shall appear is:
    - managed object classes,
    - attribute types,
    - attributes, and
    - ASN.1 Modules.

All text that is not part of the GDMO template, or ASN.1 definitions shall be preceded by "--" to indicate that they are comments. This includes clause headers.

2.1.1.4 Naming Attribute Registration

As described in 2.2.1, the conversion of Internet MIBs to ISO/CCITT MIBs requires that naming attributes be defined and registered for each ISO/CCITT object class derived from Internet management entities. Naming attributes shall be registered as

{iimcAutoName <internetEntityId>(c)},

where <internetEntityId>(c) is the OID assigned to the Internet entity from which the associated ISO/CCITT object class is derived.

2.1.2 IIMC Explicit Registration Procedures

Automated registration procedures alone are not sufficient to support the translation process. ISO/CCITT management entities other than translated objects, attributes, modules, and documents need to be explicitly registered. These entities include:

- common ASN.1 modules that may be referenced for inclusion in other ASN.1 modules of other documents,

- IIMC documents to enable MIB elements and attribute types defined in these documents to be referenced within other documents, and

- IIMC defined management elements, such as the generic notification defined in this document, the IIMC Proxy MIB defined in [23], the IIMC ACL MIB defined in [24], and the omibtransSpt MIB defined in [25].

This document allocates an arc in the registration hierarchy for explicitly registering IIMC management entities. The arc is named "iimcManual". This document assigns sub-arcs under the "iimcManual" arc in the ASN.1 module in section 4.2.

2.1.2.1 Explicit ASN.1 Modules and Documents Registration

Explicit registration of IIMC documents shall be under the iimcDocument arc, as specified in section 4.2. For example, this document iimcIIMCIMIBTRANS is registered as {iimcDocument 1}.

Explicit registration of IIMC ASN.1 modules shall be under the iimcModule arc, as specified in section 4.2. For example, the IimcAssignedOIDs module specified in section 4.2 of this document is registered as {iimcModule 1}.

2.1.2.2 Explicit Trap/Notification Registration

Internet traps/notifications and informRequests are not considered by the Internet SMI to be associated with any one object or group. The ISO/CCITT SMI, however, requires that a notification be emitted by a specific object instance. Therefore, determining which ISO/CCITT managed object class should emit specific internet traps/notifications is problematic.

This document defines a generic IIMC notification, internetAlarm (see 3.2.4 and 4.1.4) that is used to carry all Internet traps/notifications and informRequests. This notification is to be emitted according to the conditions described in section 3.2.4 either by the {iimcRFC12131354}:internetSystem managed object defined in [22] and derived from the internet system group defined in [14], or by the {iimcIIMCProxy}:cmipsnmpProxyAgent managed object defined in [23]. However, each Internet defined trap/notification and informRequest must be registered such that they may be differentiated within the IIMC generic notification.

Internet traps/notifications are registered using the OID corresponding to the value of the Internet snmpTrapOID object defined in [20].

For SNMPv1 trap PDUs, the snmpTrapOID is derived as stated in the SNMPv1/SNMPv2 Coexistence document [21] 3.1.2(2). That definition is repeated below:

> "... if the value of the generic-trap field is 'enterpriseSpecific' then the value used is the concatenation of the enterprise field from the trap PDU with two additional sub-identifiers, '0', and the value of the specific-trap field."

For notifications, informRequests, defined according to the SNMPv2 SMI, the registered OID is the value of the snmpTrapOID.

The registered OID for the Internet trap/notification is then used as the value for the probableCause field of the IIMC generic notification, internetAlarm, as defined in section 3.2.4 and 4.1.4.

2.2 NAMING PROCEDURES

ISO/CCITT objects are identified by specifying read-only attributes within the object class as naming attributes. The naming attribute is used to form the relative distinguished name of the object instance. The sequence of relative distinguished names that trace the path in the naming hierarchy from the root to the object forms a distinguished name that uniquely identifies the object instance.

2.2.1 Naming Attributes

The conversion of Internet MIBs into ISO/CCITT GDMO MIBs requires that naming attributes be defined and registered for each ISO/CCITT object class derived from Internet management entities.

This paper specifies conventions for naming attributes: their labeling, registration, and value definition, to facilitate automated generation of naming attributes for object classes derived from Internet MIBs.

The convention for assigning labels to naming attributes shall be to append "Id" to the label of the object class with which they are associated.

The convention for assigning registration OIDs to naming attributes is described in 2.1.1.4.

The convention for assigning syntaxes to naming attributes for instance identification shall be as follows.

The naming attribute value syntax shall be either an ASN.1 NULL syntax or an ASN.1 SEQUENCE identifying the Internet INDEX objects used to name the object class and their syntax.

Managed objects for which only a single instance resides in the managed system, e.g., {iimcRFC12131354}:tcp, shall use the ASN.1 NULL for their syntax. Managed object classes that may have multiple instances, e.g., table entries, shall use as their syntax an ASN.1 SEQUENCE that contains syntaxes of the internet objects identified in the INDEX (or AUGMENTS) clause of the Internet Macro from which the object class is derived, as defined in [11] or [17]. The values of the indexing attributes shall be transferred using the ASN.1 syntax specified for the Internet objects.

Syntax specifications for single instance naming attributes shall be of the form:

```
        <naming attribute label> || "IdValue" ::= NULL.
```

Syntax specifications for multiple instance naming attributes shall be of the form:

```
        <naming attribute label> || "IdValue" ::= SEQUENCE     {
                <index label>  <tag>  <index syntax>
              [, <index label> <tag>   <index syntax>]*i }
```

where the following definitions apply.

<naming attribute label> — The label of the naming attribute which is derived by appending "Id" to the label of the object class for which the naming attribute is being defined.

<index label> — The label of an Internet object which is listed in the INDEX clause of the Internet entity from which the ISO/CCITT object class is derived.

<index syntax> — The syntax associated with the Internet object having the associated <index label>.

The values of INDEX variables used in the naming attribute structure shall NOT follow the IMPLIED semantics defined in SNMPv2.

<tag>— For multiple instance object classes, the index variables shall be listed, and tagged, within the SEQUENCE in the order of their appearance in the INDEX clause of the associated OBJECT-TYPE macro from which the ISO/CCITT object class is derived. Each ASN.1 tag has the syntax "[i]", where i = 1 to the number of index variables.

2.2.2 ISO/CCITT-Internet Naming Tree

The ISO/CCITT SMI requires that managed object instances (conventionally just called managed objects) be bound into a naming hierarchy, or tree, for purposes of naming. This hierarchy is often called the containment hierarchy. The binding must specify for each managed object class: the object class which is superior to it in the containment hierarchy; and a naming attribute in the object class that is used to distinguish instances of the object class at a given level in the hierarchy. The name binding is specified after the object class has been defined.

2.2.3 Distinguished Names

The distinguished name (DN) of a managed object consists of a sequence of relative distinguished names (RDN), one for each managed object on the naming path from the root to the managed object. Each relative distinguished name contains exactly one attribute, the "naming" attribute for the corresponding class, as specified by a NAME BINDING template. This DN is used as the CMIP ManagedObjectInstance or BaseObjectInstance parameter for identifying managed objects.

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

```
{
{"Rec. X.721 | ISO/IEC 10165-2 : 1992":systemTitle     =     "troi.mitre.org"}
{{iimcRFC12131354}:ipId                                =     NULL          }
{{iimcRFC12131354}:ipRouteEntryId                      =     SEQUENCE      {
                                                       ipRouteDest {129.83.2.17}
                                                       }
}
```

with appropriate ASN.1 tags, etc., included.

Note: *The beginning of the above example distinguished name is implementation dependent. For example, the naming attribute for the system object could have been chosen to be the systemId attribute instead of the systemTitle attribute, and the system object could have been bound to object classes other than root.*

2.3 OID TRANSLATION

The procedures required to translate between Internet registered OIDs and names, and ISO/CCITT registered attribute and class OIDs are described in this section.

2.3.1 OID/Name Translation ISO/CCITT to Internet

The general procedure for deriving ISO/CCITT registered OIDs from Internet OIDs was explained in section 2.1, and the structure of the naming attribute value was explained in section 2.2. This paragraph explains how the information used in an ISO/CCITT case OID, attribute OID, and the naming attribute value may be used to create the identifier for naming Internet objects.

The following definitions apply: ((c) and (a) refer to class and attribute, respectively)

From 2.1,

{classOID} ::= {iimcAutoObjAndAttr <internetEntityId>(c)}

and

{attributeOID} ::= {iimcAutoObjAndAttr <internetEntityId>(a)}

For example, examine the {iimcRFC12131354}:ipRouteEntry object class OID:

{iimcRFC12131354}:ipRouteEntry ::= {iimcAutoObjAndAttr 1 3 6 1 2 1 4 21 1}
where <internetEntityId>(c) ::= 1 3 6 1 2 1 4 21 1,

and an attribute that belongs ipRouteEntry, {iimcRFC12131354}:ipRouteNextHop:

{iimcRFC12131354}:ipRouteNextHop ::= {iimcAutoObjAndAttr 1 3 6 1 2 1 4 21 1 7}
where <internetEntityId>(a) ::= 1 3 6 1 2 1 4 21 1 7.

Note that the attribute <internetEntityId>(a) for ipRouteNextHop is equal to <internetEntityId>(c) for its associated object class, ipRouteEntry, with the sub-identifier (7) appended to it. Most of the time the relationship:

<internetEntityId>(a) ::= <internetEntityId>(c) <sub-identifier>

is true for translated MIB attributes. This property is useful for determining the object class and object instance with which an attribute may be associated during run-time translation of Internet object instances contained in SNMP response PDUs and traps/notifications.

Note: When attributes that were not a part of the original Internet group, or table entry, are included in a translated object class, then this relationship is not valid. For example, derived attributes assigned to an object class because their inclusion was indicated by an SNMPv2 AUGMENTS clause, as discussed in section 3.1.

From 2.2, the ISO/CCITT naming attribute value contains either an ASN.1 NULL value, or a list of index variables in an ASN.1 SEQUENCE with their values represented in their appropriate syntax (where the "( )" indicates "contents of"):

(naming attribute) ::= <internet instanceId>

where the <internet instanceId> is an ASN.1 NULL, or a SEQUENCE identifying the INDEX attributes used to name the object class, and their syntax. Managed objects for which only a single instance resides in the managed system, e.g., {iimcRFC12131354}:tcp and {iimcRFC12131354}:ip, shall use the ASN.1 NULL for their value. Managed object classes that may have multiple instances, e.g., table entries, shall use as their value an ASN.1 SEQUENCE that contains values of the attributes derived from internet objects identified in the INDEX (or AUGMENTS) clause of the Internet Macro from which the object class is derived, as defined in [11] or [17].

For example, the ISO/CCITT naming attribute value for the instance of {iimcRFC12131354}:ipRouteEntry specific to IP address "129.83.2.17" contains the ipDestination index variable as an IP address represented in an ASN.1 OCTET STRING of size 4 as specified in [RFC1242].

The Internet uses the following convention to uniquely identify an Internet object instance:

{internet object name}::= {<internetEntityId>(a) <internet instanceId>}

For example, the internet object name for ipRouteNextHop corresponding to IP address 129 83 2 17 is {1 3 6 1 2 1 4 21 1 7 129 83 2 17}, where <internetEntityId>(a) ::= 1 3 6 1 2 1 4 21 1 7, <internetInstanceId> ::=129 83 2 17.

Therefore, given the contents of the naming attribute for the ISO/CCITT object instance being accessed, the <internet instanceId> is extracted. Given the attributeOID for the attribute being operated upon, the <internetEntityId>(a) is extracted. The {internet object name} is then formed from the results, taking into account appropriate conversions from their ASN.1 representation.

For example, assume that a CMIS request is issued specifying a distinguished name for an {iimcRFC12131354}:ipRouteEntry managed object as illustrated in section 2.2.3; and an attribute for ipRouteEntry, say {iimcRFC12131354}:ipRouteNextHop. The ipRouteNextHop attribute has been assigned the identifier {iimcAutoObjAndAttr 1 3 6 1 2 1 4 21 1 7} in the MIB defined in [22]. Therefore, the ipRouteNextHop attribute identifier would first have to be translated into the corresponding Internet identifier {1 3 6 1 2 1 4 21 1 7} by stripping off the iimcAutoObjAndAttr portion of the OID. Then, from the RDN value for the ipRouteEntry extract the value for the <internet instanceId>, i.e., the value for the ipRouteDest index "129 83 2 17". Finally the Internet identification for this piece of management information would be constructed according to [14] as {ipRouteNextHop 129 83 2 17}, or equivalently, {1 3 6 1 2 1 4 21 1 7 129 83 2 17}. The agent with which this information resides is identified (see 2.2.3), by the RDN for the system managed object naming attribute, e.g., the "systemTitle", as "troi.mitre.org."

2.3.2 OID/Name Translation Internet to ISO/CCITT

Internet to ISO/CCITT OID/name translation is only necessary when used during run-time proxy translation. At run-time internet identifiers are provided as internet object names in SNMP responses and traps/notifications. The internet object names are of the form described in section 2.3.1. Although actual translation is required only during run-time in proxy implementations, the translation properties, and information that may be obtained, must be understood in order to properly define the structure of the IIMC generic notification, internetAlarm, defined in section 3.2.4.

Given the definitions shown in section 2.3.1, and knowledge of the IIMC naming hierarchy (name bindings), the ISO/CCITT {classOID},{attributeOID}, and distinguished name can be derived from Internet names and the Internet agent address.

- The iimcAutoObjAndAttr OID is known.
- Using knowledge of the internet name structure as described in section 2.3.1, and knowledge of valid <internetEntityId>(a) values known to the proxy, the <internetEntityId>(a) and <internet instanceId> may be extracted from the internet name.

Note: The extraction process is not possible if the valid <internetEntityId>(a) value is not known to the proxy. The translation process cannot be performed.

- The ISO/CCITT attribute OID is formed as:

{iimcAutoObjAndAttr <internetEntityId>(a)}

- The ISO/CCITT class OID may be determined in one of two ways:

i) assume that the <internetEntityId>(a) contains the object class OID, <internetEntityId>(c), with which the attribute may be associated, as discussed in section 2.3.1. Then stripping off the final component of the OID will yield the <internetEntityId>(c). The object class OID is then formed as {iimcAutoObjAndAttr <internetEntityId>(c)}. However, see the note in section 2.3.1.

ii) use a safer approach, and determine the class OID by looking up the ISO/CCITT object class OID to which the attribute identified as {iimcAutoObjAndAttr <internetEntityId>(a)} belongs.

- The naming attribute OID may be derived by extracting the <internetEntityId(c)> from the class OID derived previously, and prepending it with the OID for {iimcAutoName} as described in 2.1.1.4.

- The managed object instance value, the object's DN, may be determined as follows:

a) the value of the naming attribute associated with the object class may be formed as:

<internet instanceId>.

b) the naming attribute value, and the naming attribute OID defined in section 2.2.1, are used to form the final RDN for the object's DN. The sequence of other RDNs for the DN are determined from knowledge of the naming hierarchy defined for proxy [23], i.e., the IIMC proxy name bindings, and the Internet agent's address.

Note: *If the Internet agent's address cannot be determined, then it may not be possible to associate a notification with a specific agent. This may be a problem if multiple Internet agents are associated with the same network address.*

2.4 INHERITANCE FOR OBJECT CLASSES

The "Rec. X.721 | ISO/IEC 10165-2 : 1992":top class defined by [8] is the ultimate superior in the ISO/CCITT inheritance hierarchy. The class "top" contains attributes which are inherited by all managed object classes that are defined using the ISO/CCITT SMI and GDMO templates.

Not all attributes of "top" need to be instantiated in any single managed object. All objects shall instantiate the mandatory "objectClass", and "nameBindings" attributes. If conditional packages may apply, an object shall instantiate the "packages" attribute.

2.5 REFERENCE LABELS FOR DERIVED ENTITIES

The labels used to reference Internet entities (groups, objects, traps) shall be used to reference the corresponding templates for the derived ISO/CCITT entity (object class, attribute, notification).

3. INTERNET TO ISO/CCITT MIB TRANSLATION PROCEDURES

The procedures for translating Internet SMI MIBs into ISO/CCITT SMI MIBs consist of pre-translation procedures, GDMO template translation procedures, and post-translation procedures. Many of the procedures are subject to automation; some are not. Comments are provided concerning possible aids to automation; however, it is not the intent of this document to provide fully automated translation algorithms.

3.1 PRE-TRANSLATION PROCEDURES

Pre-translation procedures are outlined below. The rationale for steps (a)—(d) is that the ISO/CCITT object classes and associated attributes must be identified. The rationale for steps (e)—(f) is that all ASN.1 syntax references in templates must be an ASN.1 External type reference or External value reference, i.e., must reference a label that appears on the left side of an ASN.1 construct within some ASN.1 module that appears in the document that defines the derived MIB. Internet MIBs often reference basic ASN.1 constructs, such as INTEGER and OCTET STRING, which must be converted into an External type reference. Default values must reference an External value reference.

(a) Identify Internet groups and OBJECT-TYPEs associated with each group. For SNMPv2 defined MIBs, the OBJECT-GROUP macro includes this information. For SNMPv1 defined MIBs, the group may be identified manually and then the members of the group identified by the fact that their OIDs contain the group object identifier. For SNMPv1 defined MIBs, procedure (c) must be followed.

(b) Identify conceptual table OBJECT-TYPEs, conceptual table entry (row) OBJECT-TYPEs associated with each table, and columnar OBJECT-TYPEs associated with each conceptual table entry.

For SNMPv2 defined MIBs, the MAX-ACCESS clause of the conceptual table and entry OBJECT-TYPES macro will have a value of 'not-accessible', and the convention often used is to include the word "Table" or "Entry" in the macro label. Once a conceptual table has been identified, the corresponding conceptual entry OBJECT-TYPE macro can be identified via its registered object identifier through the convention of appending a 1 to the table object identifier. Alternatively, the conceptual table's SYNTAX clause may be examined to determine the label of the corresponding conceptual entry Macro. SNMPv1 defined MIBs are not so consistent in their use of "not-accessible"; however, the other conventions apply.

For SNMPv2 defined MIBs, tables may be defined with table entries that augment (SNMPv2 AUGMENT clause) entries for an existing table. The table entry object class for the augmented table will be bound to the table entry object class that corresponds to the reference in the AUGMENTS clause.

(c) For each group, the OBJECT-TYPEs not identified in procedure (b), and not having an ACCESS or MAX-ACCESS clause value of "not-accessible", are identified for translation into attributes of an ISO/CCITT object class associated with the group. The OBJECT-TYPEs that have an ACCESS or MAX-ACCESS clause of 'not-accessible', i.e. conceptual table and conceptual table entry objects, are not translated.

Note: Some groups may not have any OBJECT-TYPEs to translate into attributes.

(d) For each conceptual table entry OBJECT-TYPE, the set (set1) of columnar OBJECT-TYPEs associated with the table entry are identified for translation into ISO/CCITT attributes of an ISO/CCITT object class associated with the entry. Another set (set2) of OBJECT-TYPES identified in the INDEX clause of the conceptual table entry OBJECT-TYPE are also identified for inclusion in the class. If the AUGMENTS clause is present, then the INDEX clause of the conceptual table entry OBJECT-TYPE pointed to by the AUGMENTS clause identifies the elements of (set2). The union of these two sets constitutes the set of ISO/CCITT attributes associated with the table entry object class. All OBJECT-TYPEs are translated, excluding those that have an ACCESS or MAX-ACCESS clause of 'not-accessible'.

Note: The set of columnar OBJECT-TYPES associated with a table entry can be identified by the SYNTAX clause for the OBJECT-TYPE for the conceptual table entry. The SYNTAX clause is of the form:

SEQUENCE OF <type1,..., typeN>

where <typeN> includes the label of an OBJECT-TYPE included in the conceptual table entry.

(e) Create an ASN.1 module for use in the GDMO template translations. Create an IMPORTS clause for the module and include in it the syntax to be imported from other modules. This may be done by including the parameters for the IMPORTS clause encountered in the Internet module. (An alternative is to import the syntax for attribute types defined in this document from the IimcCommonDef module. However, not all of the syntax may be needed, and some necessary syntax may be omitted for attribute types defined in other MIBs.)

When any Internet TEXTUAL-CONVENTIONS macros that may be present in the Internet module are translated according to the procedures of 3.2.6, the resulting ASN.1 syntax shall be included in the new ASN.1 module. TEXTUAL-CONVENTIONS macros shall be translated first so that these ASN.1 constructs may be used during the translation of OBJECT-TYPE macros.

For each OBJECT-TYPE that is to be translated into an ISO/CCITT attribute, check the value of the SYNTAX clause, and if it is not a syntax included in the IMPORTS clause of the new ASN.1 module, or defined using an SNMPv2 TEXTUAL-CONVENTION macro, then do one of the following:

i) If the value is not an External type reference: create an External type reference for the value in the SYNTAX clause and put it into the ASN.1 module. The label for the External type reference shall be the attribute label with the first letter capitalized.

ii) If the value is an External type reference put the External type reference syntax into the ASN.1 module.

(f) If a DEFVAL clause is present, create an External value reference which has the type indicated by the OBJECT-TYPE SYNTAX clause and is assigned the value of the DEFVAL clause. The label for the External value reference shall be the attribute label preceded by "c-" (lower case letter "c"). Place the External value reference into the ASN.1 module created in e). For example, the following would be a valid value references (assuming StorageType was declared in, or imported to, the same ASN.1 module):

```
c-contextStorageType  StorageType     ::= nonVolatile
```

```
        c-xyz               INTEGER     ::= 100.
```

(g) If the ASN.1 module for the Internet MIB definition contains ASN.1 value assignments, then the syntax for those value assignments pertinent to the translation shall either be placed in the ASN.1 module created in (e) or imported into the module using an External value reference.

Note: *It is recommended that a syntax that is used more than once in the MIB to be translated be defined just once in the new ASN.1 module created in (e) and referenced repeatedly. Examples of such commonly referenced types are INTEGER, OCTET STRING, and OBJECT IDENTIFIER.*

3.2 GDMO TRANSLATION PROCEDURES

Readers of this document are assumed to have familiarity with the GDMO templates and their format. The templates structures presented here should be considered definitive for MIBs defined according to the translation procedures defined in this document.

The GDMO templates in this paragraph contain elements indicated by "< x >", where "x" is to be interpreted and the result substituted into the template.

The "||" symbol indicates concatenation of elements.

Adjacent elements that are not concatenated shall be separated by at least one space.

The "[ ]" symbols surrounding a construct indicate that it maybe present or absent in any particular instance of the template.

The "[ ]*" symbols surrounding a construct indicate that it may be present or absent in any particular instance of the template an undefined number of times.

An "|" symbol is used to indicate that a choice must be made between alternate constructs.

The "!" symbol is used to delimit text strings in the templates. Other delimiters may be used, as specified by the GDMO. The delimiter may not be used in the text string unless it is "doubled", e.g.,"!!".

Elements that are defined in one template are not repeated for other templates unless its interpretation has changed.

The Internet SNMPv2 SMI also includes macros for specifying compliance with the MIBs. These macros are similar to ISO/CCITT managed object conformance statements (MOCS), and are not addressed here.

3.2.1 Translation of Groups

Internet groups may be translated to ISO/CCITT managed object classes by filling in the following GDMO template:

```
<group label> MANAGED OBJECT CLASS
        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
        CHARACTERIZED BY
        <group label> || "Pkg" PACKAGE
                [BEHAVIOUR
                <group label> || "PkgBehaviour" BEHAVIOUR
                DEFINED AS !<optional textual definition>!;;]
        ATTRIBUTES
                <group label> || "Id" GET
                ["," <OBJECT-TYPE label n>
                [DEFAULT VALUE <DEFVAL clause translation>]
                [PERMITTED VALUES <permitted attribute syntax sub-type>]
                <OBJECT-TYPE label n ACCESS clause translation>]*;;;
REGISTERED AS { iimcAutoObjAndAttr <internetEntityId>(c) };
```

where the following definitions apply.

<group label> — The label associated with the Internet group.

<optional textual definition> — Any textual description that is applicable to the resource modeled by this group, and the resource's management behaviour. Text in the SNMPv2 DESCRIPTION clause of the OBJECT-GROUP macro may be used. To facilitate parsing of BEHAVIOUR clauses for classes derived from groups, the following scannable structure shall be used (the [] indicate optional constructs, keywords must be in caps, keywords shall be excluded from the descriptive text, keywords shall appear in the order illustrated below):

```
        [BEGINPARSE
        [REFERENCE !!
                <text referencing internet document, group or object from which the ISO/CCITT
                object class was derived>!!;]
        [DESCRIPTION !!
                <applicable textual description, or text of Internet macro DESCRIPTION clause,
                if present>!!;]
        ENDPARSE]
```

The scannable structure shall be the first text in the BEHAVIOUR clause.

<OBJECT-TYPE label n> — The label of an Internet OBJECT-TYPE which is included in the group and does not represent a conceptual table, a conceptual table entry, or an OBJECT-TYPE included in a conceptual table entry. These become attributes of the object class.

<permitted attribute syntax sub-type> — The constraints on the attribute values expressed as an ASN.1 subtype that is valid for the attribute's syntax. This clause is present only when constraints are specified by the source MIB ASN.1 syntax, but are not specified in the syntax associated with the corresponding translated attribute.

<OBJECT-TYPE label n ACCESS clause translation> — The mapping of the ACCESS (or SNMPv2 MAX-ACCESS) clause value as defined below (multi-valued attributes are not permitted in the Internet SMI):

OBJECT-TYPE ACCESS Clause Value	ISO/CCITT
read-only	GET
read-write	GET-REPLACE
write-only	REPLACE
read-create	not applicable
not-accessible	not translated

Table 1. Mapping the Group ACCESS clause.

<DEFVAL clause translation> — The value of the DEFVAL clause in the form of an External value reference, i.e., <module-name>.<value-name>, where the module-name is the name of an ASN.1 module within the document in which this object class is defined, and the value-name is the label assigned to the ASN.1 value definition contained in the DEFVAL clause. Where it is necessary to refer to the label of a definition contained in another document, this may be achieved by means of a local ASN.1 module which makes use of the ASN.1 IMPORTS mechanism to import the appropriate value definition.

3.2.2 Translation of Table Entry Objects

Internet conceptual table entry objects may be translated to ISO/CCITT managed object classes by filling in the following GDMO template:

```
<OBJECT-TYPE label> MANAGED OBJECT CLASS
        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
        CHARACTERIZED BY <OBJECT-TYPE label> || "Pkg" PACKAGE
                BEHAVIOUR
                <OBJECT-TYPE label> || "PkgBehaviour" BEHAVIOUR
                DEFINED AS
                !<OBJECT-TYPE DESCRIPTION clause text>!;;
        ATTRIBUTES
                <OBJECT-TYPE label> || "Id" GET
                ["," <OBJECT-TYPE label n>
                [DEFAULT VALUE <DEFVAL clause translation>]
                [PERMITTED VALUES <permitted attribute syntax sub-type>]
                <OBJECT-TYPE label n ACCESS clause translation>]*;;;
REGISTERED AS {iimcAutoObjAndAttr <internetEntityId>(c) };
```

where the following definitions apply.

<OBJECT-TYPE label> — The label of an Internet OBJECT-TYPE which represents a conceptual table entry.

<OBJECT-TYPE label n> — The label of an Internet OBJECT-TYPE which represents an OBJECT-TYPE included in a conceptual table entry. These become attributes of the table entry managed object.

<OBJECT-TYPE label n ACCESS clause translation> — The mapping of the ACCESS (or SNMPv2 MAX-ACCESS) clause value as defined below (multi-valued attributes are not permitted in the Internet SMI):

OBJECT-TYPE ACCESS Clause Value	**ISO/CCITT**
read-only	GET
read-write*	GET-REPLACE
write-only	REPLACE
read-create*	GET-REPLACE
not-accessible	not-translated

Table 2. Mapping the Table Entry Object ACCESS clause.

* Some attributes that were derived from OBJECT-TYPEs with a read-create or read-write access clause will be changed to GET during post-translation processing of INDEX type attributes. See section 3.3.3.

<OBJECT-TYPE DESCRIPTION clause text> — To facilitate parsing of BEHAVIOUR clauses for classes derived from table entries, the following scannable structure shall be used (the [] indicate optional constructs, keywords must be in caps, keywords shall be excluded from the descriptive text, keywords shall appear in the order illustrated below):

```
[BEGINPARSE
[REFERENCE !!
        <text referencing internet document, group or object from which the ISO/CCITT
        object class was derived>!!;]
[DESCRIPTION !!
        <text of Internet macro DESCRIPTION clause, if present>!!;]
INDEX
        [IMPLIED] <Internet ASN.1 module reference> || "." ||<index object label>
        [,[IMPLIED] <Internet ASN.1 module reference> || "." || <index object label>]*;
[AUGMENTS <entry label that the object class augments>;]
ENDPARSE]
```

where the following definitions apply.

IMPLIED — The key word as encountered in the INDEX clause of SNMPv2 MIBs. The SNMPv2 semantics of IMPLIED are to be applied to the index variable that it precedes when translating from OSI names into Internet names. The values of INDEX variables, when used in the naming attribute structure, shall NOT follow the IMPLIED semantics defined in SNMPv2.

<Internet ASN.1 module reference> — The label for the ASN.1 module that contains the Internet MIB definitions,

<index object label> — The Internet label assigned to the object that appears in the Internet macro INDEX clause for a table entry.

The scannable structure shall be the first text in the BEHAVIOUR clause.

3.2.3 Translation of Other OBJECT-TYPES

Other Internet OBJECT-TYPEs are translated into ISO/CCITT attributes by filling in the following GDMO template:

```
<OBJECT-TYPE label> ATTRIBUTE
        [WITH ATTRIBUTE SYNTAX
                <module identification>|| "." || <SYNTAX clause translation 1>;]
        | [DERIVED FROM <SYNTAX clause translation 2>;]
        [MATCHES FOR <SYNTAX clause type matching rules>;]
        [BEHAVIOUR
                <OBJECT-TYPE label> || "Behaviour" BEHAVIOUR
                DEFINED AS
                !<OBJECT-TYPE DESCRIPTION clause text>!;;]
REGISTERED AS {iimcAutoObjAndAttr <internetEntityId>(a)};
```

where the following definition applies.

<module identification> — The label of the ASN.1 module that contains the ASN.1 syntax being referenced. The module must appear in the document that defines the translated MIB.

ISO/CCITT have the concept of a generic "attribute type", which is a defined type that can be used in the definition of specific attributes. Attribute types have a defined syntax, generic semantics, and matching rules. For example, counter and gauge are attribute types.

The SNMPv2 SMI has a similar concept embodied in the TEXTUAL-CONVENTIONS macro, which allows the definition of "Internet attribute types" with associated syntax and semantics. See section 3.2.6 for translation procedures associated with the TEXTUAL CONVENTIONS macro.

Attributes of the defined SNMP types (e.g., Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32, Counter64, NsapAddress) shall inherit the semantics associated with the type — not just the syntax. As such, they could have been defined as Internet attribute types using a TEXTUAL CONVENTIONS macro. See 4.1.4 for the conversion of these types into ISO/CCITT attribute types. In addition, 4.1.4 contains ISO/CCITT attribute type definitions derived from [18].

Attribute templates derived from OBJECT-TYPE macros that specify these Internet attribute types in their SYNTAX clause shall contain the DERIVED FROM clause. Attribute templates derived from other ASN.1 types shall contain the WITH SYNTAX clause.

<SYNTAX clause translation 1> — Translation of the SYNTAX clause into a valid type reference name using the ASN.1 External type reference notation as GDMO requires.

<SYNTAX clause type matching rules> — The matching rules for use in CMIS filtering operations.

These rules also apply to External type references that reference the syntax type. These matching rules may be applied by automated mechanisms and then examined in the post-translation phase.

Syntax Type	Matching Rules
INTEGER, ENUMERATED	EQUALITY, ORDERING
OCTET STRING	EQUALITY, ORDERING, SUBSTRINGS
BIT STRING	EQUALITY
OBJECT IDENTIFIER*	EQUALITY, ORDERING
NULL	EQUALITY

Table 3. Mapping to matching rules.

* ORDERING, when applied to OBJECT IDENTIFIER, refers to the lexicographical ordering of the OIDs as used in [12] [19].

See 4.1.2 for the matching rules that are inherited from some ISO/CCITT attribute types derived from Internet attribute types.

<SYNTAX clause translation 2> — Attributes of the defined SNMP/SNMPv2 types (e.g., Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32, Counter64, NsapAddress), and Internet attribute types defined using the SNMPv2 TEXTUAL CONVENTIONS macro, shall be treated as ISO/CCITT attribute types. Specific attributes are derived from these types.

The following table indicates the translations required for Internet attribute types as defined either in this document or Internet documents. ISO/CCITT attribute types corresponding to the following Internet attribute types are defined in 4.1.2.

Syntax Type	Substituted Value
AutonomousType	{iimcIIMCIMIBTRANS} :autonomousType
Counter	{iimcIIMCIMIBTRANS} :counter32
Counter32	{iimcIIMCIMIBTRANS} :counter32
Counter64	{iimcIIMCIMIBTRANS} :counter64
DisplayString	{iimcIIMCIMIBTRANS} :displayString
Gauge	{iimcIIMCIMIBTRANS} :gauge32
Gauge32	{iimcIIMCIMIBTRANS} :gauge32
InstancePointer	{iimcIIMCIMIBTRANS} :instancePointer
IpAddress	{iimcIIMCIMIBTRANS} :ipAddress
MacAddress	{iimcIIMCIMIBTRANS} :macAddress
NetworkAddress	{iimcIIMCIMIBTRANS} :ipAddress
NsapAddress	{iimcIIMCIMIBTRANS} :nsapAddress
Opaque	{iimcIIMCIMIBTRANS} :opaque
PhysAddress	{iimcIIMCIMIBTRANS} :physAddress
RowStatus	{iimcIIMCIMIBTRANS} :rowStatus
TestAndIncrement	{iimcIIMCIMIBTRANS} :testAndIncrement
TimeInterval	{iimcIIMCIMIBTRANS} :timeInterval
TimeStamp	{iimcIIMCIMIBTRANS} :timeStamp
TimeTicks	{iimcIIMCIMIBTRANS} :timeTicks
TruthValue	{iimcIIMCIMIBTRANS} :truthValue
UInteger32	{iimcIIMCIMIBTRANS} :uInteger32

Table 4. Common attribute types.

<OBJECT-TYPE DESCRIPTION clause text> — To facilitate parsing of BEHAVIOUR clauses for attributes derived from Internet objects, the following scannable structure shall be used (the [] indicate optional constructs, keywords must be in caps, keywords shall be excluded from the descriptive text, keywords shall appear in the order illustrated below):

```
[BEGINPARSE
[REFERENCE !!
        <text referencing internet document, object from which the ISO/CCITT attribute
        was derived>!!;]
[DESCRIPTION !!
        <text of Internet macro DESCRIPTION clause, if present>!!;]
[UNITS !!
        <text of Internet macro UNITS clause, if present indicating the units associated
        with the attribute>!!;]
[DEFVAL
        <value in the Internet macro DEFVAL clause, if present, indicating the default
        value for the attribute>;]
ENDPARSE]
```

The scannable structure shall be the first text in the BEHAVIOUR clause.

3.2.4 Translation of Notifications

The Concise MIB Definitions [13] for SNMPv1 does not contain a macro for representing traps since, in SNMPv1, traps were considered part of the protocol and not part of the MIB. A subsequent attempt was made to correct this problem in [15] by defining a TRAP-TYPE macro. The SNMPv2 SMI [17] defines a NOTIFICATION macro and its mapping onto an SNMPv2 PDU. The document on SNMPv1/SNMPv2 Coexistence [21] defines a mapping between SNMPv1 trap PDUs and SNMPv2 notifications. Therefore, by inference, there exists a mapping of SNMP trap PDUs into an SNMPv2 NOTIFICATION macro.

The ISO/CCITT SMI models notifications as being emitted by specific managed objects. As a consequence, notifications must be assigned to appropriate object classes at the time the object class is defined. New notifications cannot be added to an object class without changing the class's registration.

The Internet SMI has no explicit concept of traps being associated with an object. Consequently, determining the IIMC derived managed object which is the source of a notification is not always possible. Therefore, this document defines a generic notification into which all Internet traps/notifications may be mapped.

Internet Traps/Notifications may contain information related to multiple internet objects. Consequently, the generic notification may contain variables not affiliated with the same derived ISO/CCITT object class. This document requires that variables be placed into the generic notification even if they are not attributes of the object class from which the notification is emitted.

The generic notification, "internetAlarm", shall be emitted by the internetSystem managed object as defined in [22] and derived from the internet system group defined in [14]. The notification shall be sent in the unconfirmed mode in the context that an Internet trap/notification would be sent, and in the confirmed mode in the context that an SNMPv2 InformRequest PDU would be sent.

When generated within a proxy, the events that shall trigger the notification to be emitted are the receipt of an Internet trap/notification, or an SNMPv2 InformRequest PDU.

In accordance with [7] the decision whether to send a notification in the confirmed or unconfirmed mode is a matter for the agent to determine based on the policies associated with the manager.

The SNMPv2 InformRequest PDU shall cause the notification to be sent in the confirmed mode, with the response containing no reply information, i.e., the CMIS service shall omit the event reply parameter.

All SNMP traps/notifications shall cause the generic notification to be sent in the unconfirmed mode.

In the case of proxy, an Internet trap/notification and SNMPv2 InformRequest PDU for which the agent address cannot be determined by the proxy shall cause the generic notification to be emitted by a different object than the internetSystem object, as defined in [23].

The internetAlarm notification is defined in section 4.1.4.

3.2.5 Log Record for Internet Alarm

If internetAlarm notifications and event reports are to be logged, then a corresponding log record object class must be defined. The internetAlarmRecord managed object class is defined in section 4.1.1.

3.2.6 Translation of Internet Attribute Types

ISO/CCITT has the concept of a generic "attribute type", which is a defined type that can be used in the definition of specific attributes. Attribute types have a defined syntax, generic semantics, and matching rules. For example, counter and gauge are attribute types.

The SNMPv2 SMI has a similar concept embodied in the TEXTUAL-CONVENTION macro, which allows the definition of "Internet attribute types" with associated syntax and semantics.

Attributes of the defined SNMP types (e.g., Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32, Counter64, NsapAddress) should inherit the semantics associated with the type — not just the syntax. As such, they could have been defined as Internet attribute types using a TEXTUAL-CONVENTION macro.

ISO/CCITT attribute types are defined using the ATTRIBUTE template, without the REGISTERED AS clause.

```
<Internet attribute type label> ATTRIBUTE
        [[WITH ATTRIBUTE SYNTAX
                <module identification>|| "." ||<SYNTAX clause translation 1>;]
        | [DERIVED FROM <SYNTAX clause translation 2>;]]
        [MATCHES FOR <SYNTAX clause type matching rules>;]
        [BEHAVIOUR
        <Internet attribute type label>  || "Behaviour"
                BEHAVIOUR
                DEFINED AS
                !<OBJECT-TYPE DESCRIPTION clause text>!;;]
```

where the following definitions apply.

<Internet attribute type label> — The label associated with the TEXTUAL-CONVENTION macro, or with the generic type that could have been defined using that macro.

<OBJECT-TYPE DESCRIPTION clause text> — To facilitate parsing of BEHAVIOUR clauses for attributes derived from internet objects, the following scannable structure shall be used (the [] indicate optional constructs, keywords must be in caps, keywords shall be excluded from the descriptive text, keywords shall appear in the order illustrated below):

```
        [BEGINPARSE
        [REFERENCE !!
                <text referencing internet document, object from which the ISO/CCITT attribute
                was derived>!!;]
        [DESCRIPTION !!
                <text of Internet macro DESCRIPTION clause, if present>!!;]
        [UNITS !!
                <text of Internet macro UNITS clause, if present indicating the units associated
                with the attribute.>!!;]
        [DISPLAY-HINT !!
                <hints on how to display integer or octet string attributes as defined in [18]. This
                may be the text of the Internet macro DISPLAY-HINT clause.>!!;
        ENDPARSE]
```

The scannable structure shall be the first text in the BEHAVIOUR clause.

Attribute templates derived from OBJECT-TYPE macros that specify Internet attribute types in their SYNTAX clause shall specify the corresponding ISO/CCITT attribute types in their DERIVED FROM clause.

In many cases, an SNMP SMI MIB will define a new ASN.1 type which is repeatedly referenced by a large number of OBJECT-TYPE macros. In this case, it would be useful to define a new generic attribute type once and then use DERIVED FROM wherever the type is used. Furthermore, if the new ASN.1 type is actually a refinement of one of the defined SNMP types (for example, a refinement of DisplayString), it is desirable that the behaviour associated with the defined SNMP type gets carried over into the translated MIB. To accomplish this, such cases could use the DERIVED FROM clause when defining new generic attribute types. For example, the ASN.1 syntax:

```
        DateAndTime ::= DisplayString (SIZE (14))
        -- comments provide additional semantics
```

could be represented as a new generic attribute type:

```
dateAndTime ATTRIBUTE
        DERIVED FROM {iimcIIMCIMIBTRANS}:displayString;
        BEHAVIOUR dateAndTimeBehaviour BEHAVIOUR
        DEFINED AS !<text from comments>!;;
```

Constraints on the attribute value range, e.g., (SIZE(14)) from the example, may either be included in the syntax definition, if one is specified, or textually in the descriptive text. This avoids the problem of always having to specify a PERMITTED VALUES clause in the class template to constrain the values, or value range, of an attribute derived from a generic attribute type. The latter use of PERMITTED VALUES is recommended only for special cases where additional special constraints are to be applied.

3.3 POST-TRANSLATION PROCEDURES

Post-translation procedures generally include manual checking of the BEHAVIOUR clause text for proper behaviour definitions, the addition of information concerning variables for creation and deletion in the case of NAME BINDING templates, and the creation of name bindings for placing the derived ISO/CCITT objects into the containment hierarchy. Also, ATTRIBUTE templates must be created for the naming attributes assigned to the derived ISO/CCITT object classes.

Post-translation of the property-label is required for attributes derived from Internet objects used in conceptual table entry creation and deletion.

Post-translation may also be required for the ASN.1 module created during the translation process.

3.3.1 Post-translation of BEHAVIOUR Cause

The SNMP and SNMPv2 text descriptions often contain SNMP/SNMPv2 protocol, or SMI, relevant information that is inappropriate for an ISO/CCITT object class or attribute; such text shall be removed or properly modified.

For BEHAVIOUR clauses in NAME BINDING templates, the behaviour of the object relative to creation and deletion, and any constraints that pertain, shall be explained, especially if the action causes an effect on the resource, e.g., deletion of a transport connection object may cause the transport connection to be terminated.

The scannable structures within the BEHAVIOUR shall be checked for completeness and missing fields filled in.

3.3.2 Creation of NAME BINDING Templates

The ISO/CCITT name bindings for object classes to be bound into the naming hierarchy described in section 2.2.2 are created by filling in the GDMO template defined below.

```
<subordinate-superior MOC labels>||"NB" NAME BINDING
        SUBORDINATE OBJECT CLASS
                <object class label> AND SUBCLASSES;
        NAMED BY SUPERIOR OBJECT CLASS
                <superior object class label> AND SUBCLASSES;
        WITH ATTRIBUTE
                <object class label> || "Id";
        BEHAVIOUR
        <subordinate-superior MOC labels> || "Behaviour"
                BEHAVIOUR
                DEFINED AS !<Behaviour text>!;;
        [CREATE WITH-AUTOMATIC-INSTANCE-NAMING, WITH-REFERENCE-OBJECT;]
        [DELETE DELETES-CONTAINED-OBJECTS;]
REGISTERED AS { <name binding OID>};
```

<subordinate-superior MOC labels> — The combination of the subordinate and superior managed object class reference labels separated by a hyphen. An example of the resulting label is: ip-systemNB.

<superior object class label> — The reference label of the superior object class in the naming hierarchy.

Table entry object classes, derived from conceptual table entries, have the object class derived from the group in which they are contained as their superior. One way to determine the group is to use the structure of the OID for the table entry object class, i.e., it contains the internet specific portion of the OID for the group. However, table entry object classes derived from OBJECT-TYPES that contain an AUGMENTS clause have the table entry object class derived from the OBJECT-TYPE referenced in the AUGMENTS clause as their superior.

<Behaviour text> — If required, the behaviour clause shall describe any conditions, beyond the superior versus subordinate relationship, concerning the creation or deletion of the object relative to the existence of other objects, or attribute values in other objects.

To facilitate parsing of BEHAVIOUR clauses for name bindings, the following scannable structure shall be used (the [] indicate optional constructs, keywords must be in caps, keywords shall be excluded from the descriptive text, keywords shall appear in the order illustrated below):

```
        [BEGINPARSE
        [REFERENCE !!
                <text referencing internet document, group or object from which the ISO/CCITT
                object class was derived>!!;]
        [DESCRIPTION !!
                <text describing object create/delete behaviour>!!;]
        [INDEX
                [IMPLIED] <Internet ASN.1 module reference> || "." ||<index object label>
                [,[IMPLIED] <Internet ASN.1 module reference> || "." || <index object label>]*;]
        [AUGMENTS
                <entry label that the object class augments>;]
        [DELETEATT
                <label of Internet OBJECT-TYPE used for deletion of conceptual table entry
                object>;
        [DELETEVALUE
                SNMPV2ROWSTATUS |
                <valid ASN.1 value of DELETEATT object indicating deletion>;]
        ENDPARSE]
```

where the following definitions apply.

IMPLIED — The key word as encountered in the INDEX clause of SNMPv2 MIBs. The SNMPv2 semantics of IMPLIED are to be applied to the index variable that it precedes when translating from OSI names into Internet names. The values of INDEX variables, when used in the naming attribute structure, shall NOT follow the IMPLIED semantics defined in SNMPv2.

<Internet ASN.1 module reference> — The label for the ASN.1 module that contains the Internet MIB definitions.

<index object label> — The Internet label assigned to the object that appears in the Internet macro INDEX clause for a table entry.

The SNMPV2ROWSTATUS keyword indicates that the definition of the attribute type rowStatus (see 4.1.2), designed for use in SNMP row creation and deletion, is the DELETEATT attribute type. The semantics and syntax of rowStatus are appropriate for a proxy to use for row creation and deletion.

The DELETEATT and DELETEVALUE constructs shall appear in the scannable behaviour if the name binding specifies that the object class may be created or deleted, i.e., the CREATE and DELETE clauses appear in the name binding.

The scannable structure shall be the first text in the BEHAVIOUR clause.

For example, the INDEX clause for the {iimcRFC1447}:aclEntry in the Party MIB translation is translated as:

```
        INDEX   SNMPv2-Party-MIB.aclSubject,
                SNMPv2-Party-MIB.aclTarget,
                SNMPv2-Party-MIB.aclResources;
```

The INDEX clause for the {iimcRFC1447}:contextEntry in the Party MIB translation is translated as:

```
        INDEX   IMPLIED SNMPv2-Party-MIB.contextIdentity;
```

The INDEX clause for the {iimcRFC1447}:viewEntry in the Party MIB translation is translated as:

```
        INDEX   SNMPv2-Party-MIB.viewIndex,
                IMPLIED SNMPv2-Party-MIB.viewSubtree;
```

The Internet SMI only allows the possibility of conceptual table entries being created and deleted. Many table entries are automatically created and deleted as a result of normal resource operation, and are not appropriate for creation and deletion by management means. However, dynamic creation and deletion of such objects by management may still be desired, e.g., for interface cards that may be dynamically added or removed. Another example is to allow the deletion of transport connections by management, thereby causing the transport connection to be terminated.

For SNMPv2 defined MIBs, if the table entry contains an OBJECT-TYPE that has a SYNTAX clause value of "RowStatus" and a MAX-ACCESS clause value of "read-create", then the name binding for the table entry shall contain the CREATE and DELETE clauses, and appropriate scannable clauses in the BEHAVIOUR clause, as specified in the template defined in this clause.

For SNMPv1 defined MIBs, the use of an object for the purposes of creation and deletion is inconsistent. Usually, the text definition for the table entry may need to be consulted to determine if creation and deletion are allowed, and to determine the columnar object with an ACCESS clause of "read-write" and an associated value which indicates deletion. If a columnar object is defined in the entry that is used for creation and deletion, then the name binding for the table entry shall contain the CREATE and DELETE clauses, and appropriate scannable clauses in the BEHAVIOUR clause, as specified in the template defined in this clause.

If a columnar object is not defined in the entry for use in creation and deletion, then the name binding for the table entry shall not contain the CREATE and DELETE clauses, and associated scannable clauses in the BEHAVIOUR clause, as specified in the template defined in this clause.

Name bindings definitions that do not adhere to the criteria stated above for inclusion or exclusion of CREATE and DELETE clauses for object classes derived from SNMPv1 and SNMPv2 entities, or for which behaviour clauses contain different semantics than are defined for the Internet equivalent entity, shall be noted in the System Conformance Statement (SCS) in Annex A.

<name binding OID> — As defined in section 2.1.1.2.

The conventions for name binding registration shall be as defined below.

Object classes derived from Internet groups shall be bound to the ISO/CCITT system object class defined in [8]. Object classes derived from Internet conceptual table entries shall be bound to the object classes derived from the group with which they are associated. Object classes derived from Internet conceptual table entries which augment other Internet conceptual table entries shall be bound to the table entry object class that they augment.

The structure of the naming tree is illustrated below.

Note: The system object class may be bound to objects other than root.

```
"CCITT Rec. X.660 | ISO/IEC 98344-1 : 1992": root
        |
        |-- "Rec. X.721 | ISO/IEC 10165-2 : 1992" : system
                |
                |-- group derived object class
                |
                |-- group derived object class
                |               |-- table entry
                |                       |-- augmentation of table entry
                |
                |-- group derived object class
                |
                | . . .
```

The naming tree for the Internet MIB-II derived object classes [22] is illustrated below.

```
 "CCITT Rec. X.660 | ISO/IEC 9834-1 : 1992":root
        |
        |-- "Rec. X.721 | ISO/IEC 10165-2 : 1992":system
                 |
                 |-- {iimcRFC12131354}:internetSystem
                 |
                 |-- {iimcRFC12131354}:at --- {iimcRFC12131354}:atEntry
                 |
                 |-- {iimcRFC12131354}:egp --- {iimcRFC12131354}:egpNeighEntry
                 |
                 |-- {iimcRFC12131354}:icmp
                 |
                 |-- {iimcRFC12131354}:interfaces --- {iimcRFC12131354}:ifEntry
                 |
                 |-- {iimcRFC12131354}:ip
                 |     |--- {iimcRFC12131354}:ipRouteEntry
                 |     |--- {iimcRFC12131354}:ipAddrEntry
                 |     |--- {iimcRFC12131354}:ipNetToMediaEntry
                 |     |--- {iimcRFC12131354}:ipForwardEntry
                 |
                 |-- {iimcRFC12131354}:snmp
                 |
                 |-- {iimcRFC12131354}:tcp --- {iimcRFC12131354}:tcpConnEntry
                 |
                 |-- {iimcRFC12131354}:udp --- {iimcRFC12131354}:udpEntry
```

3.3.3 Attribute Property-label Changes

OSI naming attributes are constrained to be GET only since the name of the object cannot change during its lifetime. Since the name is derived from the values of the Internet objects used for indexing conceptual table entries, the attributes derived from those indexing objects shall not be modified after the table entry object has been created.

For managed object classes that have been derived from Internet conceptual rows, ensure that the property-label of the attributes derived from the Internet objects used for naming have the GET property-label. The attributes may be identified by the INDEX construct of the scannable notation used in the behaviour clause associated with the template.

3.3.4 Post-processing of ASN.1 Module

The syntax specific to each of the naming attributes must be inserted into the ASN.1 module. Insert into the ASN.1 module the syntax and type references appropriate for the all naming attributes, as derived according to the procedures in 2.2.1, for referencing by templates for naming attributes. The syntax shall be of the form specified in 2.2.1. The label associated with the syntax shall be of the form <class label> || "IdValue".

It may be possible to collapse repeated ASN.1 type references into a single common type reference, if desired. Several commonly-used ASN.1 type references (e.g., Integer64k, Integer128, ObjectIdentifier) are defined by the IimcCommonDef ASN.1 module specified in 4.2, and may be IMPORTed into translated MIB ASN.1 modules when appropriate. However, care must be taken to ensure that any new common type reference labels are appropriately reflected in templates.

3.3.5 Templates for Naming Attributes

Naming attributes shall be defined for each object class as described in 2.2.1, and included as part of the object class templates in 3.2. The templates shall be of the form:

```
<class label> || "Id" ATTRIBUTE
        WITH ATTRIBUTE SYNTAX
        <module identification>|| "." || <class label> || "IdValue";
        MATCHES FOR EQUALITY;;
        BEHAVIOUR
                <class label> || "Id"  || "Behaviour" BEHAVIOUR
                DEFINED AS
                !The naming attribute for object class <class label>!;;
REGISTERED AS {iimcAutoName <internetEntityId>(c)};
```

where the following definitions apply.

<module identification> — The label of the ASN.1 module that contains the ASN.1 syntax being referenced. The module must appear in the document that defines the translated MIB. It must contain the ASN.1 syntax appropriate for the naming attribute, as specified in 2.2.1, and associated with the label <class label> || "IdValue".

<class label> — The label of the class for which the naming attribute is being defined.

3.3.6 Generation of MOCS

A MOCS proforma shall be generated according to the format specified by [10]. All information shall be derived directly from the GDMO MIB except for the "set by create" column in attribute tables. The "set by create" column is "m" for all attributes which are GET- REPLACE [ADD-REMOVE]. The "set by create" column is "x" for all attributes which are GET only.

4. IIMCIMIBTRANS MIB

The GDMO templates and ASN.1 modules are included here in one section to facilitate automated processing. Comments and subsection headers are included in the form of ASN.1 comments, i.e., preceded by "--".

This document (IIMCIMIBTRANS) is allocated the following registration identifier for purposes of referencing material contained herein.

```
iimcIIMCIMIBTRANS OBJECT IDENTIFIER ::={ iimcDocument 1 }
```

-- 4.1 IMIBTRANS MIB GDMO TEMPLATES

-- 4.1.1 IMIBTRANS Managed Object Classes

```
internetAlarmRecord MANAGED OBJECT CLASS
    DERIVED FROM
    "Rec. X.721 | ISO/IEC 10165-2 : 1992":logRecord;
    CHARACTERIZED BY
    internetAlarmRecordPkg PACKAGE
        BEHAVIOUR
        internetAlarmRecordBehaviour BEHAVIOUR
            DEFINED AS !This managed object is used to represent logged
            information that resulted from internetAlarm notifications or event
            reports.!;;
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":probableCause GET;;;

    CONDITIONAL PACKAGES

    attributeIdentifierListPkgPACKAGE
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeIdentifierList GET;
    REGISTERED AS {iimcPackage 1};
    PRESENT IF
        !The attributeIdentifierList parameter is present in the internetAlarm notification
        or event report corresponding to the instance of the internet alarm record.!,

    objectInstanceListPkg PACKAGE
        ATTRIBUTES
        objectInstanceList GET;
    REGISTERED AS {iimcPackage 2};
    PRESENT IF
        !The objectInstanceList parameter is present in the internetAlarm notification or
        event report corresponding to the instance of the internet alarm record.!,

    perceivedSeverityPkg PACKAGE
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":perceivedSeverity GET;
    REGISTERED AS {iimcPackage 3};
        PRESENT IF !The perceivedSeverity parameter is present in the internetAlarm
        notification or event report corresponding to the instance of the internet alarm
        record.!,

    notificationIdPkg PACKAGE
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":notificationIdentifier GET;
    REGISTERED AS {iimcPackage 4};
    PRESENT IF
        !The notificationId parameter is present in the internetAlarm notification or event
        report corresponding to the instance of the internet alarm record.!,

    correlatedNotPkg PACKAGE
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":correlatedNotifications GET;
    REGISTERED AS {iimcPackage 5};
    PRESENT IF
        !The correlatedNot parameter is present in the internetAlarm notification or event
        report corresponding to the instance of the internet alarm record.!,

    transportDomainPkg PACKAGE
        ATTRIBUTES
        transportDomain GET;
    REGISTERED AS {iimcPackage 6};
    PRESENT IF
        !The transportDomain parameter is present in the internetAlarm notification or
        event report corresponding to the instance of the internet alarm record.!,

    transportAddressPkg PACKAGE
        ATTRIBUTES
        transportAddress GET;
    REGISTERED AS {iimcPackage 7};
    PRESENT IF
        !The transportAddress parameter is present in the internetAlarm notification or
        event report corresponding to the instance of the internet alarm record.!,

    accessControlInfoPkg PACKAGE
        ATTRIBUTES
        accessControlInfo GET;
    REGISTERED AS {iimcPackage 8};
    PRESENT IF
        !The accessControlInfo parameter is present in the internetAlarm notification or
        event report corresponding to the instance of the internet alarm record.!,

    additionalInformationPkg PACKAGE
        ATTRIBUTES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":additionalInformation GET;
    REGISTERED AS {iimcPackage 9};
    PRESENT IF
        !The additionalInformation parameter is present in the internetAlarm notification
        or event report corresponding to the instance of the internet alarm record.!;
REGISTERED AS {iimcObjectClass 1};
```

-- 4.1.2 IMIBTRANS Attribute Types

-- The following ISO/CCITT attribute types, listed in alphabetical order, are derived from Internet
-- attribute types to facilitate Internet MIB translation. Other attributes may be derived from these
-- types.

```
autonomousType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.AutonomousType;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        autonomousTypeBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!Represents an independently extensible type identification value. It
            may, for example, indicate a particular sub-tree with further MIB
            definitions, or define a particular type of protocol or hardware.!!;
        ENDPARSE!;;;

counter32 ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.Counter32;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        counter32Behaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [17] by the same name.!!;
        ENDPARSE!;;;

counter64 ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.Counter64;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        counter64Behaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [17] by the same name.!!;
        ENDPARSE!;;;

dateAndTime ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.DateAndTime;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        dateAndTimeBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!A date-time specification.
            field   octets contents             range
            -----   ------ --------             -----
            1       1-2    year                 0..65536
            2       3      month                1..12
            3       4      day                  1..31
            4       5      hour                 0..23
            5       6      minutes              0..59
            6       7      seconds              0..60 (use 60 for leap-second)
            7       8      deci-seconds         0..9
            8       9      direction from UT    "+"/ "-"
            9       10     hours from UT        0..11
            10      11     minutes from UT      0..59

            For example, Tuesday May 26, 1992 at 1:30:15 PM EDT would be
            displayed as: 1992-5-26,13:30:15.0,-4:0

            Note that if only local time is known, then timezone information (fields 8-
            10) is not present.!!;
        DISPLAY-HINT !!2d-1d-1d,1d:1d:1d.1d,1a1d:1d!!;
        ENDPARSE!;;;

displayString ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.DisplayString;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR
        displayStringBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!Represents textual information taken from the NVT ASCII character
            set, as defined in pages 4, 10-11 of RFC 854. Any object defined using
            this syntax shall not exceed 255 characters in length.!!;
        DISPLAY-HINT !!255a!!;
        ENDPARSE!;;;

gauge32 ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.Gauge32;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        gauge32Behaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [17] by the same name.!!;
        ENDPARSE!;;;

instancePointer ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.InstancePointer;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        instancePointerBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!A pointer to a specific instance of a conceptual row of a MIB table in the
            managed device. By convention, it is the name of the particular instance
            of the first columnar object in the conceptual row.!!;
        ENDPARSE!;;;

ipAddress ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.IpAddress;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR
        ipAddressBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!This attribute represents a 32-bit internet address. It is represented as
            an octet string of length 4, in network Byte-order.!!;
        ENDPARSE!;;;

macAddress ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.MacAddress;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR
        macAddressBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!Represents an 802 MAC address represented in the `canonical' order
            defined by IEEE 802.1a, i.e., as if it were transmitted least significant bit
            first, even though 802.5 (in contrast to other 802.x protocols) requires
            MAC addresses to be transmitted most significant bit first.!!;
        DISPLAY-HINT !!1x:!!;
        ENDPARSE!;;;

nsapAddress ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.NsapAddress;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR
        nsapAddressBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [17] by the same name.!!;
        DESCRIPTION
            !!This attribute represents an ISO/CCITT network address. It is length
            octet string. The first octet of the string contains a binary value, in the
            range of 0..20, and indicates the length in octets of the NSAP. Following
            the first octet, is the NSAP expressed in concrete binary notation,
            starting with the most significant octet. A zero-length NSAP is used as a
            "special" address, meaning "the default NSAP" (analogous to the IP
            address 0.0.0.0). Such an NSAP is encoded as a single octet containing
            the value 0. All other NSAPS are encoded in at least 4 octets.!!;
        ENDPARSE!;;;

opaque ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.Opaque;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        opaqueBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [17] by the same name.!!;
        DESCRIPTION
            !!This attribute represents arbitrary ASN.1 syntax. A value is encoded
            using the Basic Encoding Rules [3] into a string of octets. This, in turn, is
            encoded as an OCTET STRING, in effect "double-wrapping" the original
            ASN.1 value.!!;
        ENDPARSE!;;;

physAddress ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.PhysAddress;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR
        physAddressBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!Represents media- or physical-level addresses.!!;
        DISPLAY-HINT !!1x:!!;
        ENDPARSE!;;;

rowStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.RowStatus;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        rowStatusBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!The RowStatus attribute is used by SNMP to manage the creation and
            deletion of conceptual rows, and is used as the value of the SYNTAX
            clause for the conceptual row status column.

            Creation and deletion of object classes derived from conceptual rows
            shall only be via the CMIS M-CREATE and M-DELETE services.

            The rowStatus attribute has two valid values:
```

```
            •   `active', which indicates that the conceptual row is available for use
                by the managed device;

            •   `notInService', which indicates that the conceptual row exists in the
                agent, but is unavailable for use by the managed device.

            When used in conjunction with a CMIS M-CREATE request, the
            rowStatus attribute shall indicate whether the entry shall be created in
            the "active" state, or the "notInService" state.

            When retrieved, the rowStatus attribute shall only return one of the two
            values described above.

            When used in a SET operation, the rowStatus attribute shall only
            assume one of the two values described above.!!;
        ENDPARSE!;;;

testAndIncr ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.TestAndIncr;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        testAndIncrBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
            !!This corresponds to the type defined in [18] by the same name.!!;
        DESCRIPTION
            !!Represents integer-valued information used for atomic operations.
            When the management protocol is used to specify that an object
            instance having this syntax is to be modified, the new value supplied via
            the management protocol must precisely match the value presently held
            by the instance. If not, the management protocol set operation fails with
            an error of `inconsistentValue'. Otherwise, if the current value is the
            maximum value of 2^31-1 (2147483647 decimal), then the value held by
            the instance is wrapped to zero; otherwise, the value held by the
            instance is incremented by one. (Note that regardless of whether the
            management protocol set operation succeeds, the previous value held
            by the instance is returned.)

            The value of the ACCESS clause for objects having this syntax is either
            `read-write' or `read-create'. When an instance of a columnar object
            having this syntax is created, any value may be supplied via the
            management protocol.!!;
        ENDPARSE!;;;

timeInterval ATTRIBUTE
    WITH ATTRIBUTE SYNTAX IimcCommonDef.TimeInterval;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        timeIntervalBehaviour BEHAVIOUR
        DEFINED AS
        !BEGINPARSE
        REFERENCE
```

```
                !!This corresponds to the type defined in [18] by the same name.!!;
            DESCRIPTION
                !!A period of time, measured in units of 0.01 seconds.!!;
            ENDPARSE!;;;

timeStamp ATTRIBUTE
        DERIVED FROM timeTicks;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
            timeStampBehaviour BEHAVIOUR
            DEFINED AS
            !BEGINPARSE
            REFERENCE
                !!This corresponds to the type defined in [18] by the same name.!!;
            DESCRIPTION
                !!The value of MIB-II's sysUpTime object at which specific occurrence
                happened. The specific occurrence must be defined in the description of
                any object defined using this type.!!;
            ENDPARSE!;;;

timeTicks ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.TimeTicks;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
            timeTicksBehaviour BEHAVIOUR
            DEFINED AS
            !BEGINPARSE
            REFERENCE
                !!This corresponds to the type defined in [18] by the same name.!!;
            DESCRIPTION
                !!This attribute type represents a non-negative integer which represents
                the time, modulo 2->32 (4294967296 decimal), in hundredths of a
                second between two epochs. When attributes are defined which use this
                attribute type, the description of the object identifies both of the
                reference epochs.!!;
            ENDPARSE!;;;

truthValue ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.TruthValue;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            truthValueBehaviour BEHAVIOUR
            DEFINED AS
            !BEGINPARSE
            REFERENCE
                !!This corresponds to the type defined in [18] by the same name.!!;
            DESCRIPTION !!Represents a boolean value.!!;
            ENDPARSE!;;;

uInteger32 ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.UInteger32;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
            uInteger32Behaviour BEHAVIOUR
```

```
                DEFINED AS
                !BEGINPARSE
                REFERENCE
                        !!This corresponds to the type defined in [18] by the same name.!!;
                DESCRIPTION
                        !!As defined for the ASN.1 Built-in INTEGER type. Only the value range
                        constraint (0..4294967295) is added.!!;
                ENDPARSE!;;;
```

-- 4.1.3 IMIBTRANS Attributes

```
accessControlInfo ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.AccessControlInfo;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                accessControlInfoBehaviour BEHAVIOUR
                DEFINED AS
                !This attribute is used for filtering on the access control information associated with
                internetAlarms.!;;
REGISTERED AS {iimcAttribute 1};

internetTrapInfo ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.InternetTrapInfo;
        BEHAVIOUR
                internetTrapInfoBehaviour BEHAVIOUR
                DEFINED AS
                !This attribute is used for filtering on the internetTrapInfo field associated with
                internetAlarms.!;;
REGISTERED AS {iimcAttribute 2};

objectInstanceList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.ObjectInstanceList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
                objectInstanceListBehaviour BEHAVIOUR
                DEFINED AS
                !This attribute is used for filtering on the object instances associated with
                internetAlarms.!;;
REGISTERED AS {iimcAttribute 3};

transportAddress ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.TransportAddress;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR
                transportAddressBehaviour BEHAVIOUR
                DEFINED AS
                !The transport service address by which the party receives network management
                traffic, formatted according to the corresponding value of transportDomain. For
                snmpUDPDomain, transportAddress is formatted as a 4-octet IP Address
                concatenated with a 2-octet UDP port number.!;;
REGISTERED AS {iimcAttribute 4};
```

```
transportDomain ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.TransportDomain;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                transportDomainBehaviour BEHAVIOUR
                DEFINED AS
                !Indicates the kind of transport service by which the party receives network
                management traffic. An example of a transport domain is 'snmpUDPDomain' (SNMP
                over UDP).!;;
REGISTERED AS {iimcAttribute 5};

unknownVarBindList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX IimcCommonDef.UnknownVarBindList;
        BEHAVIOUR
                unknownVarBindListBehaviour BEHAVIOUR
                DEFINED AS
                !This attribute is used for filtering on the unknownVarBindList field associated with
                internetAlarms.!;;
REGISTERED AS {iimcAttribute 6};
```

-- 4.1.4 IMIBTRANS Notifications

```
internetAlarm NOTIFICATION
                BEHAVIOUR
                internetAlarmBehaviour BEHAVIOUR
                DEFINED AS
                !This is a generic notification for translated Internet SNMPv1 traps and SNMPv2
                notifications.

                The attributeIdentifierList, and objectInstanceList fields contain information which
                may be duplicated in other fields. They are included to facilitate filtering of
                notifications on the basis of contained attributes and the object instances to
                which the notification may pertain.

                The probableCause field shall contain the snmpTrapOID as derived in clause
                2.1.2. This uniquely distinguishes SNMP traps and may be used for filtering.
                Only the "globalValue", i.e., OID, form of the probableCause syntax shall be
                used.

                The attributeIdentifierList field shall contain the attribute identifiers for the
                attributes derived from the varBind components of the SNMP variable-
                bindings list. This field is optional.

                The objectInstanceList field shall contain the object instances associated with the
                attributes derived from the varBind components of the SNMP variable-bindings
                list. This field is optional.

                The internetTrapInfo field shall contain the attributes and their values, and
                optionally their associated object instances, as derived from the varBind
                components of the SNMP variable-bindings list. This field is optional.

                The unknownVarBindList shall consist of the sequence of varBinds contained in
                the variable-bindings list for which translation was not possible, i.e., the attribute
```

OID and object instance information could not be determined. This field is optional.

The perceivedSeverity, notificationIdentifier, and correlatedNotification field semantics are as defined in [6], and the syntax is as defined in [8]. These fields are optional.

The transportDomain field shall contain the OID for the transport protocol associated with the Internet agent that sent the alarm. This field is optional.

The transportAddress field shall contain the transport layer address that the Internet agent used to send the SNMP trap/notification. The format of the field shall be in accordance with the transportDomain format. This field is optional.

The accessControlInfo field shall contain the access control information associated with the trap/notification, i.e., either the community string or the party information. This field is optional.

The additionalInformation field shall contain any additional information that may be associated with the notification.!;;

```
        WITH INFORMATION SYNTAX IimcCommonDef.InternetAlarmInfo
        AND ATTRIBUTE IDS
        probableCause           "Rec. X.721 | ISO/IEC 10165-2 : 1992":probableCause,
        attributeIdList         "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeIdentifierList,
        objectInstanceList      objectInstanceList,
        unknownVarBindList      unknownVarBindList,
        internetTrapInfo internetTrapInfo,
        perceivedSeverity       "Rec. X.721 | ISO/IEC 10165-2 : 1992":perceivedSeverity,
        notificationId          "Rec. X.721 | ISO/IEC 10165-2 : 1992":notificationIdentifier,
        correlatedNot           "Rec. X.721 | ISO/IEC 10165-2 : 1992":correlatedNotifications,
        transportDomain         transportDomain,
        transportAddress        transportAddress,
        accessControlInfo       accessControlInfo,
        additionalInformation   "Rec. X.721 | ISO/IEC 10165-2 : 1992":additionalInformation;
REGISTERED AS {iimcNotification 1};
```

-- 4.2 IMIBTRANS ASN.1 MODULES

```
IimcAssignedOIDs
{iso(1) member-body(2) 124 forum(360501) iimcManual(15) iimcModule(0) 1}
DEFINITIONS ::= BEGIN




-- The following object identifiers are assigned for IIMC use by the NM Forum --

iimcAutoTrans   OBJECT IDENTIFIER ::= { iso(1) member-body(2) 124 forum(360501) 14 }
iimcManual      OBJECT IDENTIFIER ::= { iso(1) member-body(2) 124 forum(360501) 15 }

-- The following object identifiers are assigned for registering MIB elements
-- generated using the translation procedures defined by this document. For example,
-- these arcs are used by the translated MIB-II [22] and translated Party MIB [24].
-- Refer to section 2.1.1 of this document for instructions on use of these arcs.

iimcAutoModule          OBJECT IDENTIFIER ::= { iimcAutoTrans 0 }
iimcAutoObjAndAttr              OBJECT IDENTIFIER ::= { iimcAutoTrans 1 }
iimcAutoNameBinding             OBJECT IDENTIFIER ::= { iimcAutoTrans 4 }
iimcAutoDocument                OBJECT IDENTIFIER ::= { iimcAutoTrans 13 }
iimcAutoName                    OBJECT IDENTIFIER ::= { iimcAutoTrans 15 }

-- The following object identifiers are assigned for manual registration of MIB elements
-- specified by IIMC documents. These arcs are used to register IIMC documents and the
-- common MIB elements defined by this document, the Proxy MIB defined by [23], the
-- ACL MIB defined by [24], and the omibtransSpt MIB defined by [25].
-- Refer to section 2.1.2 of this document for description of the use of these arcs.

iimcModule              OBJECT IDENTIFIER ::= { iimcManual 0 }
iimcAttribute           OBJECT IDENTIFIER ::= { iimcManual 1 }
iimcObjectClass         OBJECT IDENTIFIER ::= { iimcManual 3 }
iimcNameBinding         OBJECT IDENTIFIER ::= { iimcManual 4 }
iimcNotification        OBJECT IDENTIFIER ::= { iimcManual 5 }
iimcParameter           OBJECT IDENTIFIER ::= { iimcManual 9 }
iimcPackage             OBJECT IDENTIFIER ::= { iimcManual 10 }
iimcDocument            OBJECT IDENTIFIER ::= { iimcManual 13 }
iimcExtension           OBJECT IDENTIFIER ::= { iimcManual 14 }

-- The following object identifiers are assigned to register IIMC documents.

iimcIIMCIMIBTRANS       OBJECT IDENTIFIER ::= {iimcDocument 1}
iimcIIMCProxy           OBJECT IDENTIFIER ::= {iimcDocument 3}
iimcIIMCACLMIB          OBJECT IDENTIFIER ::= {iimcDocument 4}
iimcIIMCOMIBTRANS       OBJECT IDENTIFIER ::= {iimcDocument 5}

END
```

```
IimcCommonDef
{iso(1) member-body(2) 124 forum(360501) iimcManual(15) iimcModule(0) 2}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
        AdditionalInformation, ProbableCause,
        PerceivedSeverity, NotificationIdentifier,
        CorrelatedNotifications, AttributeIdentifierList
                FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3)
                part2(2) asn1Module(2) 1}
        Attribute, ObjectInstance
                FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) version(1) protocol(3)}
        Counter32, Counter64, NsapAddress, IpAddress,
        UInteger32, Gauge32, Opaque, TimeTicks, Integer32
                FROM SNMPv2-SMI
        TEXTUAL-CONVENTION, DateAndTime, DisplayString,
        PhysAddress, MacAddress, TruthValue, TestAndIncr,
        AutonomousType, InstancePointer, TimeStamp,
        TimeInterval
                FROM SNMPv2-TC;

AccessControlInfo ::= CHOICE  {
        communityString         [0] OCTET STRING,
        partyInfo               [1] SEQUENCE{
                                        srcParty OBJECT IDENTIFIER,
                                        dstparty OBJECT IDENTIFIER,
                                        context OBJECT IDENTIFIER
                                        }
                                }

Integer ::= INTEGER

Integer128 ::= INTEGER (0..127)

Integer64k ::= INTEGER (0..65535)

InternetAlarmInfo ::= SEQUENCE      {
        probableCause             ProbableCause,
        attributeIdList         [1] AttributeIdentifierList OPTIONAL,
        objectInstanceList      [2] ObjectInstanceList OPTIONAL,
        unknownVarBindList      UnknownVarBindList OPTIONAL,
        internetTrapInfo        [5] InternetTrapInfo OPTIONAL,
        perceivedSeverity       [6] PerceivedSeverity OPTIONAL,
        notificationId          [7] NotificationIdentifier OPTIONAL,
        correlatedNot           [8] CorrelatedNotifications OPTIONAL,
        transportDomain         [9] TransportDomain OPTIONAL,
        transportAddress        [10] TransportAddress OPTIONAL,
        accessControlInfo       [11] AccessControlInfo OPTIONAL,
        additionalInformation   [12] AdditionalInformation OPTIONAL
        }

InternetTrapInfo ::= SET OF SEQUENCE {
        objectInstance ObjectInstance OPTIONAL,
        COMPONENTS OF Attribute}
```

```
ObjectIdentifier ::= OBJECT IDENTIFIER

ObjectInstanceList ::= SET OF ObjectInstance

OctetString ::= OCTET STRING

RowStatus ::=   INTEGER          {
        -- the following two values are states:
                active(1),
                notInService(2)
        }

TransportAddress ::= OCTET STRING

TransportDomain ::= OBJECT IDENTIFIER

UnknownVarBindList ::= CHOICE {
                v1               [3] RFC1157-SNMP.VarBindList,
                v2               [4] SNMPv2-PDU.VarBindList
                                 }

END
```

5. COMPLIANCE AND CONFORMANCE

5.1 COMPLIANCE

An Internet MIB translated into ISO/CCITT GDMO format in compliance with this specification shall:

(a) be registered in accordance with the procedures specified by section 2.1,

(b) comply with the naming approach specified by section 2.2,

(c) contain object identifiers derived in accordance with the procedures specified by section 2.3,

(d) comply with the inheritance hierarchy specified by section 2.4,

(e) contain GDMO templates labeled in accordance with the convention specified by section 2.5,

(f) comply with the pre-translation procedures specified by section 3.1,

(g) contain GDMO templates derived in accordance with the templates and procedures specified by section 3.2, and

(h) comply with post-translation procedures specified by (at minimum) sections 3.3.1-3.3.3 and 3.3.5-3.3.6. Deviations permitted by clause 3.3.2 shall be documented in a System Conformance Statement as shown in Annex A.

5.2 CONFORMANCE

An implementation claiming conformance to a translated ISO/CCITT GDMO MIB shall conform to the all of the requirements stated in the corresponding MOCS proforma (generated according to the procedure specified by section 3.3.6).

An implementation claiming conformance to any IMIBTRANS GDMO templates or ASN.1 types specified in section 4 shall conform to the requirements stated in the corresponding MOCS proforma templates specified by Annex A.

ANNEX A (NORMATIVE): MANAGED OBJECT CONFORMANCE STATEMENTS (MOCS)

A.1 INTERNET ALARM RECORD [{IIMCIIMCIMIBTRANS}]

Table A.1.1 internetAlarmRecord Managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class t which conformance is claimed? (Y/N)
1	internetAlarmRecord	{1 2 124 360501 15 3 1}		

Table A.1.2 internetAlarmRecord PACKAGES

Index	Package template label	Value of object identifier for package	Constraints and values	Status	Support	Additional information
1	accessControlInfoPkg	{1 2 124 360501 15 10 8}	"The accessControlInfo parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c1		
2	additionalInformationPkg	{1 2 124 360501 15 10 9}	"The additionalInformation parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c2		
3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphicPackage	{2 9 3 2 4 17}	"if an object supports allomorphism"	o		
4	attributeIdentifierListPkg	{1 2 124 360501 15 10 1}	"The attributeIdentifierList parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c3		
5	correlatedNotPkg	{1 2 124 360501 15 10 5}	"The correlatedNot parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c4		
6	internetAlarmRecordPkg		Mandatory	m		
7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": logRecordPackage		Mandatory	m		
8	notificationIdPkg	{1 2 124 360501 15 10 4}	"The notificationId parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c5		
9	objectInstanceListPkg	{1 2 124 360501 15 10 2}	"The objectInstanceList parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c6		
10	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packagesPackage	{2 9 3 2 4 16}	"any registered package, other than this package has been instantiated"	c7		
11	perceivedSeverityPkg	{1 2 124 360501 15 10 3}	"The perceivedSeverity parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c8		
12	"Rec. X.721 | ISO/IEC 10165-2 : 1992": topPackage		Mandatory	m		
13	transportAddressPkg	{1 2 124 360501 15 10 7}	"The transportAddress parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c9		
14	transportDomainPkg	{1 2 124 360501 15 10 6}	"The transportDomain parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record."	c10		

Table A.1.3 internetAlarmRecord Attribute support

				Set by create		Get		Replace		Add		Remove		Set to default		
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	accessControlInfo	{1 2 124 360501 15 1 1}	CHOICE	x		c1		-		-		-		-		
2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": additionalInformation	{2 9 3 2 7 6}	SET OF SEQUENCE	x		c2		-		-		-		-		
3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{2 9 3 2 7 50}	SET OF ObjectClass	-		o		-		-		-		-		
4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeIdentifierList	{2 9 3 2 7 8}	SET OF AttributeId	x		c3		-		-		-		-		
5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": correlatedNotifications	{2 9 3 2 7 12}	SET OF SEQUENCE	x		c4		-		-		-		-		
6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": logRecordId	{2 9 3 2 7 3}	CHOICE	m		m		-		-		-		-		
7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": loggingTime	{2 9 3 2 7 59}	GeneralizedTime	m		m		-		-		-		-		
8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{2 9 3 2 7 63}	OBJECT IDENTIFIER	o		m		-		-		-		-		
9	"Rec. X.721 | ISO/IEC 10165-2 : 1992": notificationIdentifier	{2 9 3 2 7 16}	INTEGER	x		c5		-		-		-		-		
10	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{2 9 3 2 7 65}	ObjectClass	-		m		-		-		-		-		
11	objectInstanceList	{1 2 124 360501 15 1 3}	SET OF ObjectInstance	x		c6		-		-		-		-		
12	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{2 9 3 2 7 66}	SET OF OBJECT IDENTIFIER	o		c7		-		-		-		-		
13	"Rec. X.721 | ISO/IEC 10165-2 : 1992": perceivedSeverity	{2 9 3 2 7 17}	ENUMERATED	x		c8		-		-		-		-		
14	"Rec. X.721 | ISO/IEC 10165-2 : 1992": probableCause	{2 9 3 2 7 18}	CHOICE	m		m		-		-		-		-		
15	transportAddress	{1 2 124 360501 15 1 4}	OCTET STRING	x		c9		-		-		-		-		
16	transportDomain	{1 2 124 360501 15 1 5}	OBJECT IDENTIFIER	x		c10		-		-		-		-		

Table A.1.4 internetAlarmRecord Attribute group support

There are no attribute groups specified for this managed object class.

Table A.1.5 internetAlarmRecord Action support

There are no actions specified for this managed object class.

Table A.1.6 internetAlarmRecord Notification support

There are no notifications specified for this managed object class.

Table A.1.7 internetAlarmRecord Parameter support

There are no parameters specified for this managed object class.

Table A.1.8 internetAlarmRecord Conditions

Condition number	Condition	Reference
c1	IF "The accessControlInfo parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/1
c2	IF "The additionalInformation parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/2
c3	IF "The attributeIdentifierList parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/4
c4	IF "The correlatedNot parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/5
c5	IF "The notificationId parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/8
c6	IF "The objectInstanceList parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/9
c7	IF "any registered package, other than this package has been instantiated" THEN m ELSE -	A.1.2/10
c8	IF "The perceivedSeverity parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/11
c9	IF "The transportAddress parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/13
c10	IF "The transportDomain parameter is present in the internetAlarm notification or event report corresponding to the instance of the internet alarm record." THEN m ELSE -	A.1.2/14

Table A.1.9 internetAlarmRecord Name Binding support

Index	Name binding template label	Value of object identifier for name binding	Contraints and values	Status	Support	Additional information	Subindex	Operation	Constraints and values	Status	Support	Additional information
1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": logRecord-log	{2 9 3 2 6 3}	Superior class: "Rec. X.721 | ISO/IEC 10165-2 : 1992": Superior class: log AND SUBCLASSES	o			1.1	Create support		-		
							1.2	Create with reference object		-		
							1.3	Create with automatic instance naming		-		
							1.4	Delete support		c:m		
							1.5	Delete only if no contained objects		c:m		
							1.6	Delete contained objects		c:x		

ANNEX B: GLOSSARY

ASN.1	Abstract Syntax Notation One
CCITT	Consultative Committee on Telephony and Telegraphy
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
DN	Distinguished Name
GDMO	Guidelines for the Definition of Managed Objects
GNMP	Government Network Management Profile
IIMC	ISO/CCITT and Internet Management Coexistence
ISO	International Standards Organization
MIB	Management Information Base
MOCS	Managed Object Conformance Statement
NMF	Network Management Forum
OID	Object Identifier
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RDN	Relative Distinguished Name
RFC	Request For Comments
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SNMPv1	Simple Network Management Protocol Version 1
SNMPv2	Simple Network Management Protocol Version 2
TCP/IP	Transmission Control Protocol/Internetwork Protocol

ANNEX C: REFERENCES

1) CCITT Recommendation X.700, Management Framework Definition for Open Systems Interconnection (OSI).

   ISO/IEC 7498-4: 1989, Information Processing Systems — Open Systems Interconnection - Basic Reference Model Part 4 — Management Framework.

2) ISO/IEC 8824: Information Technology — Open System Interconnection — Specification of Abstract Syntax Notation One (ASN.1),1990.

3) CCITT Recommendation X.209 (1988), Specification of basic encoding rules for abstract syntax notation one (ASN.1).

   ISO/IEC 8825: 1990, Information Technology — Open System Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

4) CCITT Recommendation X.710, (1991), Common Management Information Service Definition for CCITT Applications.

   ISO/IEC 9595: 1991, Information Technology — Open System Interconnection — Common Management Information Service Definition.

5) CCITT Recommendation X.711 | ISO/IEC 9596-1: 1991, Information Technology — Open Systems Interconnection — Common Management Information Protocol — Part 1: Specification.

6) CCITT Recommendation X.733 (1992) | ISO/IEC 10164-4: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 4: Alarm Reporting Function.

7) CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 1: Management Information Model.

8) CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 2: Definition of Management Information.

9) CCITT Recommendation X.721 (1992) | ISO/IEC 10165-4: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 4: Guidelines for the Definition of Managed Objects.

10) CCITT Recommendation X.723 (1993) | ISO/IEC 10165-6: 1993, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 6: Requirements and Guidelines for Implementation Conformance Statement Proformas associated with OSI Management.

11) RFC1155, M. Rose and K. McCloghrie, Structure and Identification of Management Information for TCP/IP based internets, May 1990.

12) RFC1157, J.D. Case, M.S. Fedor, M.L. Schoffstall,C. Davin, Simple Network Management Protocol (SNMP), May 1990.

13) RFC1212, M. Rose, K. McCloghrie — Editors, Concise MIB Definitions, March 1991.

14) RFC1213, K. McCloghrie and M. Rose — Editors, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991.

15) RFC1215, M. Rose — Editor, A convention for Defining Traps for use with the SNMP, March 1991.

16) RFC1441, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Introduction to version 2 of the Internet-standard Network Management Framework, April 1993.

17) RFC1442, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

18) RFC1443, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

19) RFC1448, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

20) RFC1450, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

21) RFC1452, J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Coexistence between version 1 and version 2 of the Internet Network Management Framework, April 1993.

22) Network Management Forum: Forum 029, Translation of Internet MIB-II (RFC 1213) to ISO/CCITT GDMO MIB, Issue 1.0, October 1993.

23) Network Management Forum: Forum 028, ISO/CCITT to Internet Management Proxy, Issue 1.0, 1993.

24) Network Management Forum: Forum 027, ISO/CCITT to Internet Management Security, Issue 1.0, October 1993.

25) Network Management Forum: Forum 030, Translation of ISO/CCITT GDMO MIBs to Internet MIBs, Issue 1.0, October 1993.

26) NM Forum and X/Open, ISO/CCITT and Internet Management: Coexistence and Interworking Strategy, Issue 1.0, October, 1992.

27) Federal Information Processing Standards Publication 179 — Government Network Management Profile v1.0, December 1992.