ISO/CCITT and Internet Management Coexistence (IIMC): Translation of Internet MIBs to ISO/CCITT GDMO MIBs (IIMCIMIBTRANS)

Draft 2
May 26, 1993

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 nic.ddn.mil, nnsc.nsf.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

Status of this Memo i
Abstract ii
Table of Contents ii
Revision History iv
1.Introduction 1
1.1 Problem Statement 1
1.2 Overview of IIMC 1
1.3 MIB Translation Procedures 2
1.4 Native Management Model 3
1.5 Proxy Management Model 4
1.6 Scope of this Document 5
1.7 Terms and Conventions 5
2. Registration and Naming Procedures 6
2.1 Registration Procedures 6
2.1.1 Automated Registration Procedures 6
2.1.2 IIMC Explicit Registration Procedures 7
2.1.2.1 Object Classes and Attributes Registration 8
2.1.2.2 Trap/Notification Registration 8
2.1.2.3 NAME BINDINGs Registration 9
2.1.2.4 Registration of ASN.1 Modules and IIMC Documents .9
2.2 Naming Procedures 10
2.2.1 Naming Attribute 10
2.2.2 ISO/CCITT-Internet Naming Tree 10
2.2.3 Distinguished Names 11
2.3 OID Translation 11
2.3.1 OID/Name Translation
ISO/CCITT to Internet 11
2.3.2 OID/Name Translation
Internet to ISO/CCITT 13
2.4 Inheritance for Object Classes 15
2.5 Reference Labels for Derived Entities 15
3. Internet to ISO/CCITT MIB Translation Procedures 15
3.1 Pre-translation Procedures 15
3.2 GDMO Translation Procedures 18
3.2.1 Translation of Groups 19
3.2.2 Translation of Table Objects 20
3.2.3 Translation of Table Entry Objects 21

3.2.4 Translation of Other OBJECT-TYPES 23
3.2.5 Translation of Notifications 26
3.2.6 Log Record for Internet Alarm. 27
3.2.7 Translation of Internet Attribute Types 27
3.3 Post-translation Procedures 29
3.3.1 Post-translation of BEHAVIOUR Cause 29
3.3.2 Deletion of Derived MIB Elements 30
3.3.3 Creation of NAME BINDING Templates 30
3.3.4 Attribute Property-label Changes 34
3.3.4.1 Create/Delete Attributes 34
3.3.4.2 Naming (INDEX) Attributes 34
4. IIMCIMIBTRANS MIB 35
4.1 IMIBTRANS MIB GDMO Templates 35
-- 4.1.1 IMIBTRANS Managed Object Classes 35
-- 4.1.2 IMIBTRANS Attributes 36
-- 4.1.3 IMIBTRANS Notifications 38
-- 4.1.4 IMIBTRANS Attribute Types 39
4.2 IMIBTRANS ASN.1 Modules 46
5. Acknowledgments 50
References 51
Appendix A (Normative)
Managed Object Conformance Statements (MOCS) 53
Appendix B (Informative)
Issues in Conceptual Table Translation 54

Revision History

Draft 0 - October 9, 1992
Initial draft of this document.

Draft 1 - March 26, 1993
Previous draft of this document (replaced Draft 0).

Draft 2 - May 26, 1993
Current draft of this document (replaces Draft 1).

Major Changes Since Last Revision

1. Additional information added to the generic notification. Also changed the unknownVarBindList syntax to a choice between SNMPv1 and SNMPv2 syntax.
2. Updated references to reflect SNMPv2 changes.
3. All notifications are now emitted by the internetSystem object (if source agent known) or cmipsnmpProxyAgent (if source agent unknown).
4. Clarified naming hierarchy for translated MIBs.
5. Only single name bindings are now automatically assigned for translated objects.
6. Extended pre and post processing procedures.
7. Clarified that automatic and explicit registration procedures apply to entities registered under the IIMC with the NM Forum as registration authority.
8. Added a temporary Appendix B to describe issues related to translation of conceptual table objects.
9. Added a log record class related to the internetAlarm notification.
10. Changed post processing procedures to disallow creation and deletion of table entries by CMIS set attribute; now allowed only via CMIS create and delete operations.
11. Added post-translation checking to ensure that attributes used for naming table entries (derived from Internet indexing objects) may not be modified by CMIS operations except through the create/delete process. Their translated property-label must be GET.
12. Now allow multiple name bindings for the ISO system managed object (e.g., to root or to some other object class).
13. Changed ASN.1 syntax to use Internet syntax for attributes, including the syntax for counters and gauges.
14. Changed create and delete modifiers for name bindings to improve consistent translation of name bindings.
15. Put place holder for MOCS in normative Appendix A.
16. Added BEHAVIOUR clause to the name binding template and specified scannable behaviour for all BEHAVIOURs.
17. Removed CREATEDELETEATT and CREATEDELETEVALUE from the scannable behaviour for table entry objects and put them into the scannable behaviour for name bindings. All information relative to creation and deletion via CMIS or

SNMP is in the scannable behaviour for name binding.
18. Added scannable behaviour to attribute types.
19. Changed allowed operations and values for the rowStatus attribute type, and deleted most of text description.

Outstanding Issues

1. Should we keep the current translation of Internet "conceptual" tables into ISO object classes (which we call table objects)? Is the convenience they provide for scoping sufficient to warrant their inclusion in the translated MIB? Appendix B provides a description of this issue; comments are solicited during review.

2. Can most current ASN.1 compilers handle ASN.1 macros (e.g., Internet MIB macros)? The current ASN.1 modules IMPORT some Internet MIB macros for the syntax that they contain (e.g., the TEXTUAL-CONVENTION macro).

3. What should internetAlarmRecords used for logging internetAlarms contain? Comments are solicited on this new object class defined in this 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) [ISO9596-1], and related SMI documents [ISO10165-1,2,4].

- The Internet community developed the Simple Network Management Protocol (SNMP) [RFC1157], and its successor, SNMPv2 [RFC1448]. The Internet SMI is defined in [RFC1155] and [RFC1442].

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 components 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), 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

This document is part of a package of ISO/CCITT and Internet Management Coexistence (IIMC) drafts. Documents included in this package are:

[IIMCIMIBTRANS]	Translation of Internet MIBs to ISO/CCITT GDMO MIBs
[IIMCOMIBTRANS]	Translation of ISO/CCITT GDMO MIBs to Internet MIBs
[IIMCMIB-II]	Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB
[IIMCPROXY]	ISO/CCITT to Internet Management Proxy
[IIMCSEC]	ISO/CCITT to Internet Management Security

These documents together comprise a package aimed at integrating ISO/CCITT-based and Internet-based management systems. These documents represent coexistence and interworking efforts underway within the IIMC working group, chartered under the auspices of the Network Management Forum Architecture Integration ISO/Internet (AIII) technical team.

The IIMC intends to 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" [NMFTR107]. The documents included in the IIMC package are the next level of detailed specifications which implement several of the methodologies identified in the strategy.

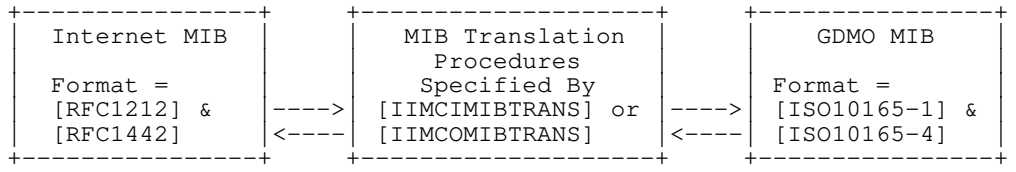
1.3 MIB Translation Procedures

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

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

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

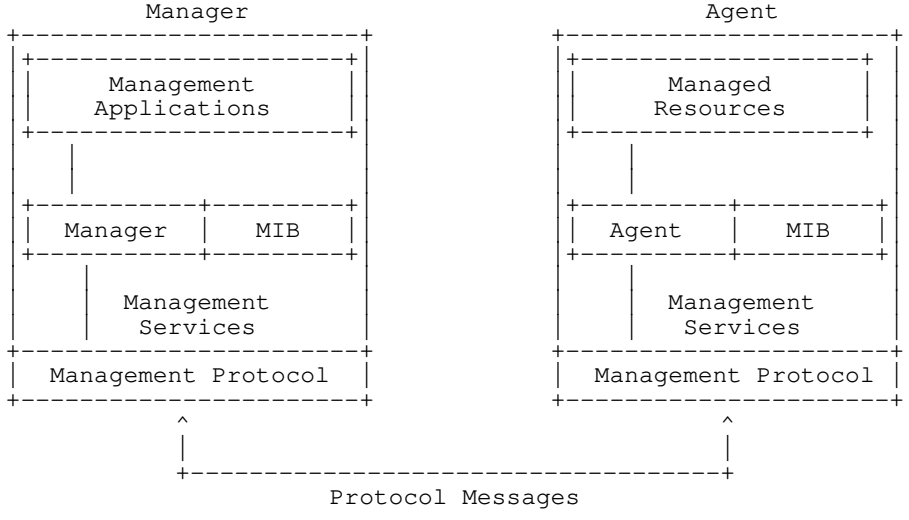
The IIMC approach is to specify direct translation procedures which yield a pair of functionally-equivalent MIBs, as shown in the following figure.



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

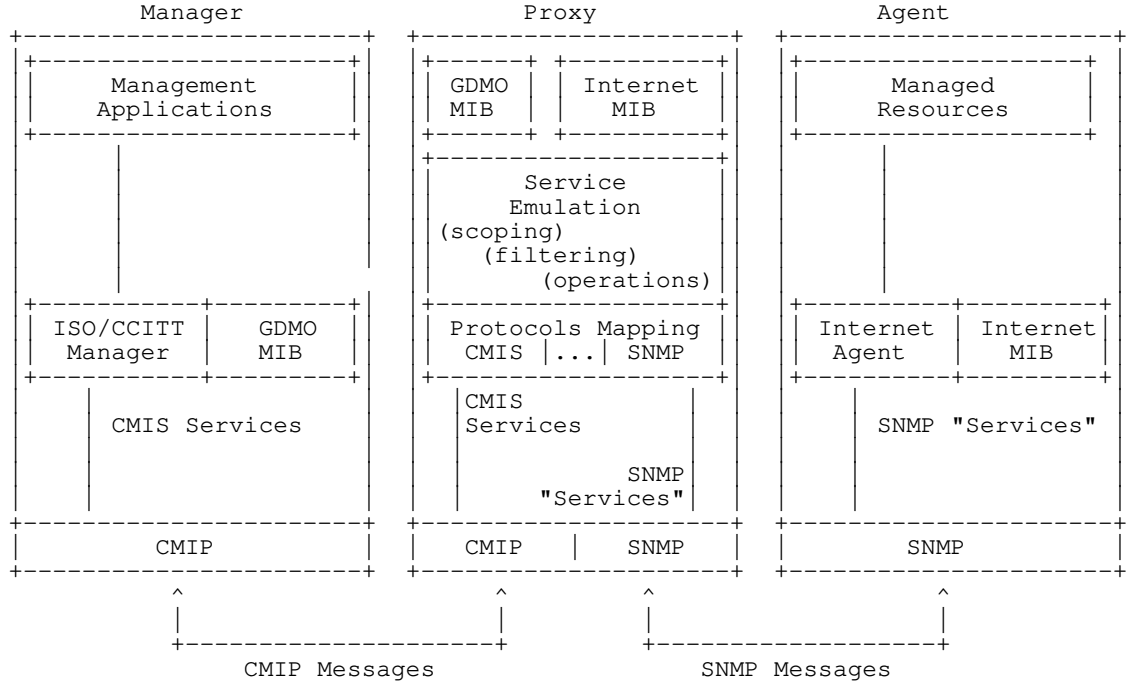
The basic model for ISO/CCITT and Internet management is illustrated in the following diagram.



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 [RFC1213] specified by [IIMCMIB-II]. 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 [IIMCPROXY]. A similar approach could also be taken to specify an Internet to ISO/CCITT proxy, although no such IIMC document is currently specified.



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 CMIS 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 [ISO10165-4]. 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

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.

Registration is accomplished by the following procedure:

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 encoded OID must be translated into two sub-identifiers.

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

```
{<iimcTransOID>}    =    {<iimcAutoTrans>
                          <internetEntityId>}
```

where <iimcAutoTrans> is the OID dedicated for ISO/CCITT-Internet automated registration procedures.

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

Internet defined groups and objects must be registered as ISO/CCITT object classes and attributes; and Internet traps must be registered for inclusion in as ISO/CCITT notifications. 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".

```
iimcAutoTrans OBJECT IDENTIFIER ::= {...TBD...}
```

Editor's Note: [This TBD value to be provided prior to publication.]

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, and Internet traps, need to be explicitly registered. These entities include:

- name bindings for object classes,
- ASN.1 modules that may be referenced for inclusion in other ASN.1 modules of other documents, - documents to enable MIB elements and attribute types defined in one document to be referenced within other documents,
- IIMC defined management elements, such as the generic notification defined in this document and the IIMC proxy MIB defined in [IIMCPROXY].

This document allocates an arc in the registration hierarchy for explicitly registering IIMC management entities. The arc is named "iimcManagement".

This document assigns sub-arcs under the "iimcManagement" arc in the ASN.1 module 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.

2.1.2.1 Object Classes and Attributes Registration

Follow the procedure described in section 2.1.1 for OIDs associated with Internet groups, conceptual tables, conceptual table entries, and objects.

2.1.2.2 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 3.2.5 and 4.1.3) that is used to carry all Internet traps/notifications and informRequests. This notification is to be emitted according to the conditions described in section 3.2.5 either by the internetSystem managed object defined in [IIMCMIB-II] and derived from the internet system group defined in [RFC1213], or by the cmipsnmpProxyAgent managed object defined in [IIMCPROXY]. 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 [RFC1450].

For SNMPv1 trap PDUs, the snmpTrapOID is derived as stated in the SNMPv1/SNMPv2 Coexistence document [RFC1452] 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 3.2.5 and 4.1.3.

2.1.2.3 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 {iimcManagementNB}. The following procedure is recommended 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. Within the class OID, extract the <internetEntityId>(c) portion of the OID. Prepend iimcManagementNB to the <internetEntityId>(c) so that the OID for the name binding is of the form:

  {iimcManagementNB <internetEntityId>(c)}

Note: in general multiple name bindings may be defined for an OSI 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.

2.1.2.4 Registration of ASN.1 Modules and IIMC Documents

ASN.1 modules defined in IIMC documents are registered under the {iimcManagementModule} arc. Modules derived for MIBs defined in Internet RFC are registered under the iimcManagementModAuto 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 iimcManagementModAuto in ascending sequence. Explicit registration of other ASN.1 modules within the IIMC sub-tree shall be under the iimcManagementModMan arc.

IIMC documents are registered under the {iimcManagementDoc} arc. Documents derived from MIBs defined in Internet RFCs are registered under the iimcManagementDocAuto 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 iimcManagementDocAuto in ascending sequence. Explicit registration of other documents within the IIMC sub-tree shall be under the iimcManagementDocMan arc.

The registration authority for the iimcManagementModule and iimcManagementDoc arcs shall be the Network Management Forum.

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 Attribute

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 a generic naming attribute, internetClassId, and the conventions for its value definition, to facilitate automated generation of naming attributes for object classes derived from Internet MIBs. This generic naming attribute is applicable to all ISO/CCITT object classes derived from Internet defined MIBs.

The format for the values of the internetClassId naming attribute is compatible with instance identification conventions used by the Internet, thereby facilitating the automated conversion of Internet MIBs into ISO/CCITT SMI format and the name mapping required for proxy management.

The internetClassId is defined in section 4.1.2.

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:

```
{
{systemId = {troi.mitre.org}
                                -- ISO/CCITT system
{internetClassId = {iimcAutoTrans 1 3 6 1 2 1 4 0}}
                                -- ip
{internetClassId = {iimcAutoTrans 1 3 6 1 2 1 4 21 0}}
                                -- ipRouteTable
{internetClassId =
          {iimcAutoTrans 1 3 6 1 2 1 4 21 1 129 83 217}}
                         -- ipRouteEntry
}
```

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 systemTitle attribute instead of the systemId 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 class 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}     ::= {iimcAutoTrans
```

```
                        <internetEntityId>(c)}
```

and

```
{attributeOID}  ::= {iimcAutoTrans
                        <internetEntityId>(a)}
```

For example, examine the ipRouteEntry object class OID:

```
ipRouteEntry ::= {iimcAutoTrans 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, ipRouteNextHop:

```
ipRouteNextHop ::=
                {iimcAutoTrans 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, table, 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 the OID formed as, (the "( )" indicates "contents of"):

```
(naming attribute)  ::=  {iimcAutoTrans
                           <internetEntityId>(c)
                           <internet instanceId>}
```

where the <internet instanceId>, the OID created uniquely for each Internet object instance, is "0" for object classes that may only have a single instance. The <internet instanceId> for object classes that may have multiple instances is an OID fragment derived from the values 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 [RFC1155] or [RFC1442].

For example, the ISO/CCITT naming attribute value for the instance of ipRouteEntry specific to IP address 129 83 2 17 is {iimcAutoTrans 1 3 6 1 2 1 4 21 1 129 83 2 17}, where <internetEntityId>(c) ::= 1 3 6 1 2 1 4 21 1, and <internetinstanceId> ::= 129 83 2 17.

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.

For example, assume that a CMIS request is issued specifying a distinguished name for an ipRouteEntry managed object as illustrated in section 2.2.3; and an attribute for ipRouteEntry, say ipRouteNextHop. The ipRouteNextHop attribute has been assigned the identifier {iimcAutoTrans 1 3 6 1 2 1 4 21 1 7} in the MIB defined in [IIMCMIB-II]. 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 iimcAutoTrans portion of the OID. Then, from the RDN value for the ipRouteEntry extract the <internet instanceId> {129 83 217}. Finally the Internet identification for this piece of management information would be constructed according to [RFC1213] 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 "systemId", 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 3.2.5.

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 iimcAutoTrans 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:

> {iimcAutoTrans <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 {iimcAutoTrans <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 {iimcAutoTrans <internetEntityId>(a)} belongs.

- 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:

{iimcAutoTrans <internetEntityId>(c) <internet instanceId>}

b) the naming attribute value, and the internetClass 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 [IIMCPROXY], 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 response or 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 "top" class defined by [ISO10165-2] 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) - (e) is that the ISO/CCITT object classes and associated attributes must be identified. The rationale for steps (f) - (g) 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.

Note: 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.

Note: For SNMPv2 defined MIBs, tables may be defined with table entries that augment (SNMPv2 AUGMENT clause) entries for an existing table. The table object classes derived from such tables will be deleted from the ISO/CCITT MIB during post-translation (see 3.3.2). The table entry object class for the deleted 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' are not translated.

(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, including 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) For each conceptual table, an ISO/CCITT object class is created that contains the generic naming attribute "internetClassId".

(f) 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 3.2.7, the resulting ASN.1 syntax shall be included in the new ASN.1 module. TEXTUAL-CONVENTIONS macros should 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.

g) 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 f).

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.
```

h) 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 (f) 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 (f) 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 presented here should be considered exemplar, and not definitive.

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
           {iimcManagementDoc 1}:internetClassId   GET
           ["," <OBJECT-TYPE label n>
           <OBJECT-TYPE label n ACCESS clause translation>
           [DEFAULT VALUE  <DEFVAL clause translation>]]*;;;
REGISTERED AS  { iimcAutoTrans <internetEntityId>(c) };
```

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 is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[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 ]
```

If used, 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.

<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

<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 Objects

Editor's Note: [The translation of conceptual table objects is an issue. See Appendix B for a discussion of this issue.]

Internet conceptual table 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
          {iimcManagementDoc 1}:internetClassId   GET;;;
REGISTERED AS  { iimcAutoTrans <internetEntityId>(c) };
```

The following definitions apply:

<OBJECT-TYPE label> - The label associated with the OBJECT-TYPE macro.

<OBJECT-TYPE DESCRIPTION clause text> - To facilitate parsing of BEHAVIOUR clauses for classes derived from tables, the following scannable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

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

If used, the scannable structure shall be the first text in the BEHAVIOUR clause.

3.2.3 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
          {iimcManagementDoc 1}:internetClassId   GET
          ["," <OBJECT-TYPE label n>
          <OBJECT-TYPE label n ACCESS clause translation>
          [DEFAULT VALUE  <DEFVAL clause translation>]]*;;;
REGISTERED AS {iimcAutoTrans <internetEntityId>(c) };
```

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	GET

* 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 create/delete attributes and INDEX type attributes. See section 3.3.4.

<OBJECT-TYPE DESCRIPTION clause text> - To facilitate parsing of BEHAVIOUR clauses for classes derived from table entries, the following scannable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[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 <attribute label, ..., attribute label>;
[AUGMENTS <entry label that the object class
               augments>;]
ENDPARSE]
```

If used, the scannable structure shall be the first text in the BEHAVIOUR clause.

Note: Table object classes that contain table entry object classes which were derived from OBJECT-TYPES with an AUGMENTS clause shall be deleted in the post-translation phase according to section 3.3.2.

3.2.4 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 {iimcAutoTrans <internetEntityId>(a)};
```

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.

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 3.2.7 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 [RFC1443].

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.

Note: 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	EQUALITY, ORDERING
OCTET STRING	EQUALITY, ORDERING, SUBSTRINGS
BIT STRING	EQUALITY
OBJECT IDENTIFIER	EQUALITY, ORDERING
NULL	EQUALITY

See section 4.1.4 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 section 4.1.4

Syntax Type	**Substituted Value**
AutonomousType	{iimcManagementDoc 1} :autonomousType
Counter	{iimcManagementDoc 1} :counter32
Counter32	{iimcManagementDoc 1} :counter32
Counter64	{iimcManagementDoc 1} :counter64
DisplayString	{iimcManagementDoc 1} :displayString
Gauge	{iimcManagementDoc 1} :gauge32
Gauge32	{iimcManagementDoc 1} :gauge32
InstancePointer	{iimcManagementDoc 1} :instancePointer
IpAddress	{iimcManagementDoc 1} :ipAddress
MacAddress	{iimcManagementDoc 1} :macAddress
NetworkAddress	{iimcManagementDoc 1} :ipAddress
NsapAddress	{iimcManagementDoc 1} :nsapAddress
Opaque	{iimcManagementDoc 1} :opaque
PhysAddress	{iimcManagementDoc 1} :physAddress
RowStatus	{iimcManagementDoc 1} :rowStatus
TestAndIncrement	{iimcManagementDoc 1} :testAndIncrement
TimeInterval	{iimcManagementDoc 1} :timeInterval
TimeStamp	{iimcManagementDoc 1} :timeStamp
TimeTicks	{iimcManagementDoc 1} :timeTicks
TruthValue	{iimcManagementDoc 1} :truthValue
UInteger32	{iimcManagementDoc 1} :uInteger32

<OBJECT-TYPE DESCRIPTION clause text> - To facilitate parsing of BEHAVIOUR clauses for attributes derived from Internet objects, the following scannable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[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]
```

If used, the scannable structure shall be the first text in the BEHAVIOUR clause.

3.2.5 Translation of Notifications

The Concise MIB Definitions [RFC1212] 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 [RFC1215] by defining a TRAP-TYPE macro. The SNMPv2 SMI [RFC1442] defines a NOTIFICATION macro and its mapping onto an SNMPv2 PDU. The document on SNMPv1/SNMPv2 Coexistence [RFC1452] 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 [IIMCMIB-II] and derived from the internet system group defined in [RFC1213]. 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 [ISO10165-1] 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 [IIMCPROXY].

The internetAlarm notification is defined in section 4.1.3.

3.2.6 Log Record for Internet Alarm.

If internetAlarms 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.

3.2.7 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>!;;]
```

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 is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[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
          [RFC1443]. This may be the text of the
          Internet macro DISPLAY-HINT clause.>!!;
ENDPARSE]
```

If used, 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.

Note: 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 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 attributes. For example, the ASN.1 syntax:

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

could be represented as a new generic attribute:

```
dateAndTime ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:displayString;
     BEHAVIOUR dateAndTimeBehaviour BEHAVIOUR
     DEFINED AS !<text from comments>!;;
```

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.

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

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

Post-translation procedures may result in deletion of some ISO/CCITT MIB elements derived from the procedures described in section 2.2.

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 should 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, should 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 clause should be checked for completeness and missing fields filled in.

3.3.2 Deletion of Derived MIB Elements

Tables which contain entries that augment the entries of another table shall be deleted from the derived MIB. The corresponding entries shall be bound to the table entries that they augment.

The reason for this is that the ISO/CCITT SMI prohibits adding attributes to an object class. The solution used in this document is to make a table entry object class that augments another table entry the direct subordinate of the table entry object class being augmented. The table is no longer needed.

3.3.3 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
                    {iimcManagementDoc 1}:internetClassId;
     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 object classes, derived from conceptual tables, have the object class derived from the group in which they were defined as their superior. One way to determine the group is to use the structure of the OID for the table object, i.e., it contains the internet specific portion of the OID for the group. However, if the table object class contains entries that were derived from Internet OBJECT-TYPES that contained an AUGMENTS clause, then the table is deleted from the MIB.

Table entry object classes, derived from conceptual table entries, have the corresponding table object class as their superior. One way to determine the table 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 table. 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> - To facilitate parsing of BEHAVIOUR clauses for name bindings, the following scannable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

[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 <attribute label, ..., attribute label>;]
[AUGMENTS <entry label that the object class
              augments>;]
[CREATEDELETEATT <label of Internet OBJECT-TYPE used for
              creation and deletion of conceptual table
              entry object>;]
[CREATEDELETEVALUE <SNMPV2ROWSTATUS> |
          <valid ASN.1 value of create/delete object
          indicating deletion>;]
ENDPARSE]
```

The SNMPV2ROWSTATUS keyword indicates that the definition of the attribute type rowStatus (see 4.1.4), designed for use in SNMP row creation and deletion, is the CREATEDELETEATT attribute type. The semantics and syntax of rowStatus are appropriate for a proxy to use for row creation and deletion.

If used, the scannable structure shall be the first text in the BEHAVIOUR clause.

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 table entry may be created and deleted.

For SNMPv1 defined MIBs, the use of "RowStatus" 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 and associated value which indicates deletion.

<name binding OID> - As defined in section 2.1.3.

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

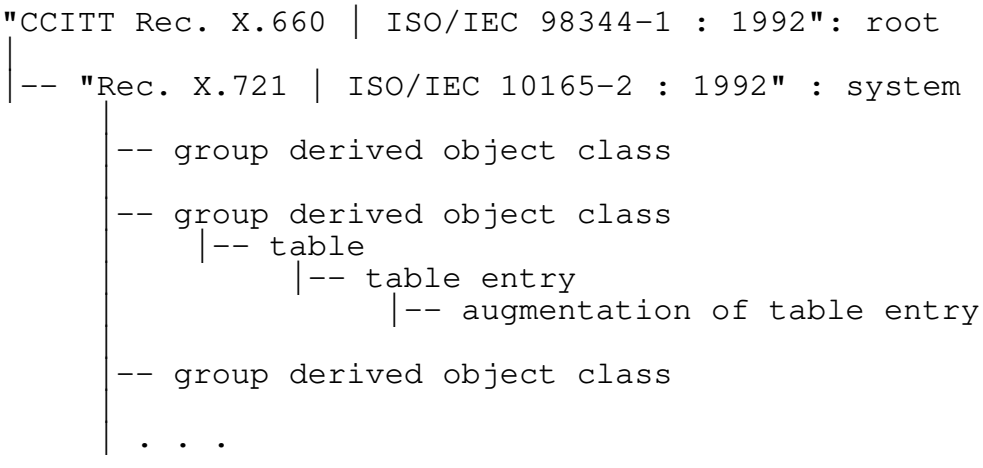
Object classes derived from Internet groups shall be bound to the ISO/CCITT system object class defined in [ISO10165-2]. Object classes derived from Internet conceptual table objects shall be bound to the object class derived from the group with

which it is associated. Object classes derived from Internet conceptual table entries shall be bound to the conceptual table object classes 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.

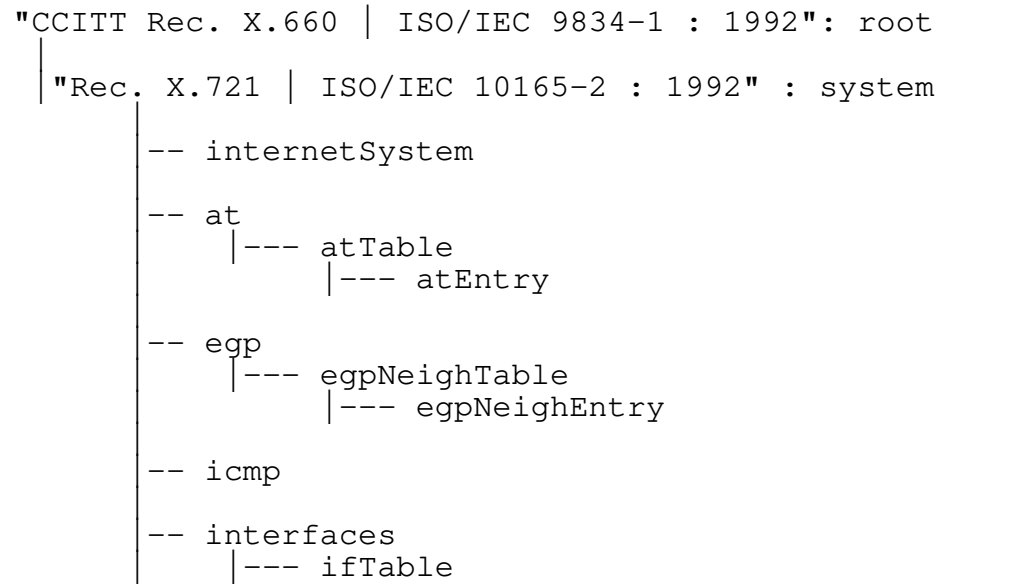
Editor's Note: [The inclusion of object classes derived from conceptual tables is a major issue; see Appendix B for a discussion of this issue. If such object classes are removed, then the conceptual table entries will be bound directly to the group with which they are associated.]

The structure of the naming tree is illustrated below.

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



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



```
|          |--- ifEntry
|
|-- ip
|     |--- ipRouteTable
|     |     |--- ipRouteEntry
|     |
|     |--- ipAddrTable
|     |     |--- ipAddrEntry
|     |
|     |--- ipNetToMediaTable
|     |     |--- ipNetToMediaEntry
|     |--- ipForwardTable
|           |--- ipForwardEntry
|
|-- snmp
|
|-- tcp
|     |--- tcpConnTable
|           |--- tcpConnEntry
|
|-- udp
      |--- udpTable
            |--- udpEntry
```

3.3.4 Attribute Property-label Changes

Attributes that are used for creation/deletion and for naming must not be modified via CMIS operations.

3.3.4.1 Create/Delete Attributes

Table entry objects are allowed to be created or deleted only via the CMIS CREATE and DELETE services. Therefore, the attributes derived from the Internet objects used to manipulate creation and deletion of conceptual rows, e.g., the rowStatus type, are to be treated as read only attributes. This prevents creation and deletion of conceptual rows via direct manipulation of the attribute values.

For managed object classes that have been derived from Internet conceptual rows, change the property-label of the attribute derived from the Internet object used for creation and deletion to GET. If the scannable notation conventions were used for the behaviour clause of the name binding template associated with the object class, then the attribute may be identified by the CREATEDELETEATT construct.

3.3.4.2 Naming (INDEX) Attributes

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 may 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. If the scannable notation conventions were used for the behaviour clause of the template associated with the object class, then the attributes may be identified by the INDEX construct.

3.3.5 Post-processing of ASN.1 Module

It may be possible to collapse repeated ASN.1 references into a single reference, if desired. However, care must be taken to ensure that any new reference labels are appropriately reflected in the templates that reference the old labels.

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 ::={iimcManagementDocMan 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":eventLogRecord;
     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
```

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

          transportDomainPkg  PACKAGE
               ATTRIBUTES
               transportDomain                         GET;;
               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;;
               PRESENT IF !The
               transportAddress parameter is present
               in the internetAlarm notification or event
               report corresponding to the instance of the
               internet alarm record.!;
REGISTERED AS {iimcManagementMOC 1};
```

-- 4.1.2 IMIBTRANS Attributes

```
internetClassId ATTRIBUTE
          WITH ATTRIBUTE SYNTAX
```

```
                    IimcCommonDef.ObjectIdentifier;
          MATCHES FOR EQUALITY;
          BEHAVIOUR
               internetClassIdBehaviour BEHAVIOUR
          DEFINED AS !This is a generic naming attribute
          intended to be used for naming all object
          classes derived from Internet MIB translation.

          For ISO/CCITT object classes that may have only a
          single instance per managed system, the value
          is the OID for the object class concatenated with
          the sub-identifier "0".  Object classes derived
          from the Internet TCP, UDP, and IP groups are
          examples of such object classes.

          For object classes that may have multiple
          instances per managed system, such as table
          entries, the value is the concatenation of the
          ISO/CCITT object class OID and the Internet object
          instance identifier (an OID) of the form
          specified for the table entry instance
          identification in the original Internet MIB
          definition.

          The Internet object instance identification is the
          concatenation of the values of the Internet
          OBJECT-TYPE(s) identified in the conceptual table
          entry OBJECT-TYPE INDEX clause.  If an SNMPv2
          AUGMENTS clause is present, the instance
          identification is the concatenation of the values
          of the OBJECT-TYPE(s) identified in the INDEX
          clause of the conceptual table entry referenced in
          the value of the AUGMENTS clause.!;;
REGISTERED AS {iimcManagementAtt 1};

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 {iimcManagementAtt 2};

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 {iimcManagementAtt 3};

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 'rfc1351Domain'
          (SNMP over UDP).!;;
REGISTERED AS {iimcManagementAtt 4};
```

-- 4.1.3 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 [ISO10164-4], and the syntax is as defined in [ISO10165-2]. 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 of the Internet agent that issued 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 may 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,
attributeIdentifierList
"Rec. X.721 | ISO/IEC 10165-2 : 1992":
                    attributeIdentifierList,
objectInstanceList      objectInstanceList,
perceivedSeverity
"Rec. X.721 | ISO/IEC 10165-2 : 1992":
                    perceivedSeverity,
notificationIdentifier
"Rec. X.721 | ISO/IEC 10165-2 : 1992":
                    notificationIdentifier,
correlatedNotification
"Rec. X.721 | ISO/IEC 10165-2 : 1992":
                    correlatedNotification,
transportDomain         transportDomain,
```

```
          transportAddress         transportAddress;
REGISTERED AS {iimcManagementNot 1};
```

-- 4.1.4 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, SUBSTRINGS;    BEHAVIOUR
          autonomousTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1443] 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 [RFC1442] 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 [RFC1442] 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 [RFC1443] by the same name.!!;
          DISPLAY-HINT !!2d-1d-1d,1d:1d:1d.1d,1a1d:1d!!;
          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.!!;
          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 [RFC1443] by the same name.!!;
          DISPLAY-HINT !!255a!!;
          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
                    may not exceed 255 characters in
                    length.!!;
          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
                    [RFC1442] 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 [RFC1443] 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 [RFC1443] 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 [RFC1443] by the same name.!!;
          DISPLAY-HINT !!1x:!!;
          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.!!;
          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 [RFC1442] 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 [RFC1442] by the same name.!!;
          DESCRIPTION !!This attribute represents arbitrary
                    ASN.1 syntax.  A value is encoded using
                    the Basic Encoding Rules [ISO8825] 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 [RFC1443] by the same name.!!;
          DISPLAY-HINT !!1x:!!;
          DESCRIPTION !!Represents media- or physical-level
                    addresses.!!;
          ENDPARSE!;;;

rowStatus ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IimcCommonDef.RowStatus;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          rowStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1443] 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.

          The rowStatus attribute shall be defined as a
          read-only (GET) attribute for IIMC defined MIBs.
          Creation and deletion of object classes derived
          from conceptual rows shall only be via the CMIS
          CREATE and DELETE services. The rowStatus
          attribute has three 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
          (see NOTE below);

          - `notReady', which indicates that the
          conceptual row exists in the agent, but is
          missing information necessary in order to be
          available for use by the managed device.!!;
          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 [RFC1443] 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 [RFC1443] by the same name.!!;
          DESCRIPTION !!A period of time, measured in units
                    of 0.01 seconds.!!;
          ENDPARSE!;;;

timeStamp ATTRIBUTE
     DERIVED FROM
               {iimcManagementDoc 1} :timeTicks;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          timeStampBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1443] 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 [RFC1443] 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 [RFC1443] 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 [RFC1443] by the same name.!!;
          DESCRIPTION
          !!As defined for the ASN.1 Builtin INTEGER type.
          Only the value range constraint (0..4294967295) is
          added.!!;
          ENDPARSE!;;;
```

4.2 IMIBTRANS ASN.1 Modules

```
IimcAssignedOIDs  {iimcManagementModMan 1}
DEFINITIONS ::= BEGIN
```

```
-- Editor's Note: [The following TBD values will be assigned
-- prior to publication.]

iimcAutoTrans OBJECT IDENTIFIER ::= {...TBD...}
iimcManagement OBJECT IDENTIFIER ::= {...TBD...}

iimcManagementNB  OBJECT IDENTIFIER ::= {iimcManagement 1}
     -- for IIMC derived NAME BINDINGS

iimcManagementModule OBJECT IDENTIFIER ::=
                                    {iimcManagementModule 1}
     -- for IIMC Translation ASN.1 Modules

iimcManagementModAuto OBJECT IDENTIFIER ::=
                                    {iimcManagementModule 2}
     -- for automatically registering IIMC ASN.1 modules by
     -- RFC number corresponding to the Internet MIB
     -- translated.

iimcManagementModMan OBJECT IDENTIFIER ::=
                                    {iimcManagement 2}
     -- for explicit registration of ASN.1 Modules

iimcManagementDoc    OBJECT IDENTIFIER ::=
                                    {iimcManagement 3}
     -- for registering IIMC documents

iimcManagementDocAuto OBJECT IDENTIFIER ::=
                                    {iimcManagementDoc 1}
     -- for automatically registering IIMC documents by RFC
     -- number corresponding to the Internet MIB translated.

iimcManagementDocMan OBJECT IDENTIFIER ::=
                                    {iimcManagementDoc 2}
     -- for explicitly registering IIMC documents

iimcIIMCIMIBTRANS OBJECT IDENTIFIER ::=
                                    {iimcManagementDocMan 1}
     -- for registering this document, IIMCIMIBTRANS

iimcIIMCProxy OBJECT IDENTIFIER ::=
                                    {iimcManagementDocMan 3}
     -- for registering document IIMCProxy

iimcIIMCSEC OBJECT IDENTIFIER ::=
                                    {iimcManagementDocMan 4}
     -- for registering document IIMCSEC

iimcIIMCOMIBTRANS OBJECT IDENTIFIER ::=
                                    {iimcManagementDocMan 5}
     -- for registering document IIMCOMIBTRANS

iimcManagementProxy  OBJECT IDENTIFIER ::= {iimcManagement 4}
```

```
     -- for ISO/CCITT-internet proxy

iimcManagementNot  OBJECT IDENTIFIER ::= {iimcManagement 5}
     -- for IIMC defined notifications

iimcManagementMOC  OBJECT IDENTIFIER ::= {iimcManagement 6}
     -- for IIMC defined object classes

iimcManagementAtt  OBJECT IDENTIFIER ::= {iimcManagement 7}
     -- for IIMC defined attributes
END


IimcCommonDef {iimcManagementModMan 2}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
          AdditionalInformation, ProbableCause,
          PerceivedSeverity, NotificationIdentifier,
          CorrelatedNotifications,
          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
                                        }
                             }

InternetAlarmInfo ::= SEQUENCE     {
     probableCause           ProbableCause,
     attributeIdList     [1] AttributeIdentifierList
                                          OPTIONAL,
     objectInstanceList  [2] ObjectInstanceList
                                     OPTIONAL,
     unknownVarBindList  CHOICE {
                         [3] RFC1157-SNMP.VarBindList,
                         [4] SNMPv2-PDU.VarBindList
                                }                 OPTIONAL,
     internetTrapInfo    [5] InternetTrapInfo
                                          OPTIONAL,
```

```
     perceivedSeverity   [6] PerceivedSeverity
                                                 OPTIONAL,
     notificationId      [7] NotificationIdentifier
                                                 OPTIONAL,
     correlatedNot       [8] CorrelatedNotification
                                                 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

RowStatus ::=  INTEGER   {
                    -- the following three values are
                    -- states:
                    -- these values may be read.
                    active(1),
                    notInService(2),
                    notReady(3)
                         }

TransportAddress ::= OCTET STRING

TransportDomain ::= OBJECT IDENTIFIER

END
```

5. Acknowledgments

The following individuals have contributed to this effort.

Bob Aronoff - NIST
Jon Biggar - NetLabs
Mary Brady - NIST
April Chang - NetLabs
Ken Chapman - Stratus Computer Inc.
Alice Chen - Boeing
Christopher Crowell - Cabletron Systems
Jock Embry - Opening Technologies
Ian Emsley - Bull S.A
Paul Golick - IBM
Ulrich Gremmelmaier - University of Stuttgart
Pramod Kalyanasundaram - University of Delaware
Ken Hunter - Hewlett-Packard
Lee LaBarre - The MITRE Corporation
David Liu - Northern Telecom
Jim MacLeod - U S West
Reece Markowsky - OSIWare
Subrata Mazumdar - IBM
Keith McCloghrie - Hughes LAN Systems
Owen Newnan - U S West
Steve Ng - MPR Teltech
Yasuhiro Ohara - NTT
Jong-Tae Park - KyungPook National University
George Pavlou - University College of London
Lisa Phifer - Bellcore
Jim Reilly - Technical Rsch Ctr of Finland
Tom Rutt - AT&T
Adarsh Sethi - University of Delaware
Raj Sirsikar - University of Delaware
Baltej Singh - OSIWare
Mark Smith - Hewlett-Packard
Einar Stefferud - Network Management Associates
Mark Sylor - Digital
Hector Trevino - Bellcore
Huy Truong - Tandem
Al Vincent - U S West
Dean Voiss - NetLabs
David Waitzman - BBN
Graham Wisdom - Timeplex
Yoshi Yamashita - NKK Corporation

References

[ISO7498-4] ISO/IEC IS 7498-4, Information Processing Systems - Open Systems Interconnection -Basic Reference Model Part 4 - Management Framework, 1989.

[ISO8824] ISO/IEC 8824: Information Technology - Open System Interconnection - Specification of Abstract Syntax Notation One (ASN.1),1990.

[ISO8825] ISO/IEC 8825: Information Technology - Open System Interconnection-Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1),1990.

[ISO9595] ISO/IEC 9595, Information Technology - Open System Interconnection - Common Management Information Service Definition, 1991.

[ISO9596-1] ISO/IEC 9596-1, Information Technology - Open Systems Interconnection - Common Management Information Protocol - Part 1: Specification, 1991.

[ISO10164-4] ISO/IEC 10164-4: Information Technology - Open Systems Interconnection - Systems Management - Part 4: Alarm Reporting Function, 1991.

[ISO10165-1] ISO/IEC 10165-1: Information Technology - Open Systems Interconnection - Structure of Management Information - Part 1: Management Information Model, 1991.

[ISO10165-2] ISO/IEC 10165-2: Information Technology - Open Systems Interconnection - Structure of Management Information - Part 2: Definition of Management Information, 1992.

[ISO10165-4] ISO/IEC 10165-4: Information Technology - Open Systems Interconnection - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects, 1991.

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

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

[RFC1212] RFC1212, M. Rose, K. McCloghrie - Editors, Concise MIB Definitions, March 1991.

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

[RFC1215] RFC1215, M. Rose - Editor, A convention for Defining Traps for use with the SNMP, March 1991.

[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.

[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.

[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.

[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.

[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.

[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.

[IIMCMIB-II] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB, Draft 2, May 1993.

[IIMCPROXY] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Proxy, Draft 2, May 1993.

[IIMCSEC] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Security, Draft 2, May 1993.

[IIMCOMIBTRANS] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of ISO/CCITT GDMO MIBs to Internet MIBs, Draft 2, May 1993.

[NMFTR107] NM Forum and X/Open, ISO/CCITT and Internet Management: Coexistence and Interworking Strategy, Issue 1.0, October, 1992.

Appendix A (Normative) Managed Object Conformance Statements (MOCS)

Editor's Note: [This section will be filled in prior to publication. When completed, this section will contain a tabular representation of the managed object classes, attributes, notifications, and name bindings defined in this document. The format of these proforma tables will be as defined by ISO/IEC 10165-6.]

Appendix B (Informative) Issues in Conceptual Table Translation

The issue is whether to keep the current translation of Internet "conceptual" tables into ISO object classes (which we call table objects), or to abandon such translations.

The Internet has the concept of "conceptual" tables and "conceptual" entries (or rows). However, conceptual tables are never instantiated as a whole, i.e., treated as a single entity, whereas conceptual entries are treated as a whole for instantiation (creation, deletion) and naming.

Conceptual tables as ISO object classes are not necessary since a conceptual table is never treated as a single entity, in the Internet context, e.g., no translated attributes or notifications are associated with the tables.

However, the use conceptual tables as containers for table entry objects may provide some convenience. The real issue is whether the convenience provided is sufficient to warrant their inclusion in the translation process.

Arguments Against Table Objects

- They are not necessary to faithfully translate Internet MIBs.

- They provide no information.

- They increase the length of distinguished names.

- Their use for scoping can be provided by filtering on the object class, e.g., the table entry object class.

- Since they don't exist in the Internet context, proxy devices must use special handling to treat them as "pseudo-objects."

- Current OSI defined MIBs do not use the concept of table objects - the availability of filtering is assumed to be present with scoping as is called for in the multiple object selection functional unit of CMIP.

- Management personnel do not like to see them cluttering up their consoles, especially if they are used to current SNMP based managers.

Arguments For Table Objects

- By increasing the granularity of scoping, they minimize the need for filters. For example, suppose we want to retrieve the ipNetToMediaEntry objects using the ip object as the base managed object with a scoping level of one (no table objects). Without filtering, the table entry objects for ipRouteEntry, ipAddrEntry, and ipNetToMediaEntry would be retrieved. With

filtering on the object class, only the ipNetToMediaEntry objects would be retrieved - but at the expense of filtering.

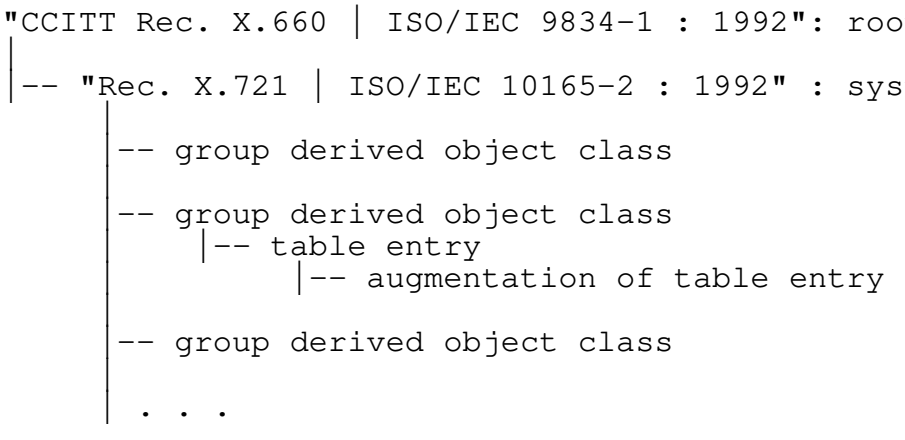
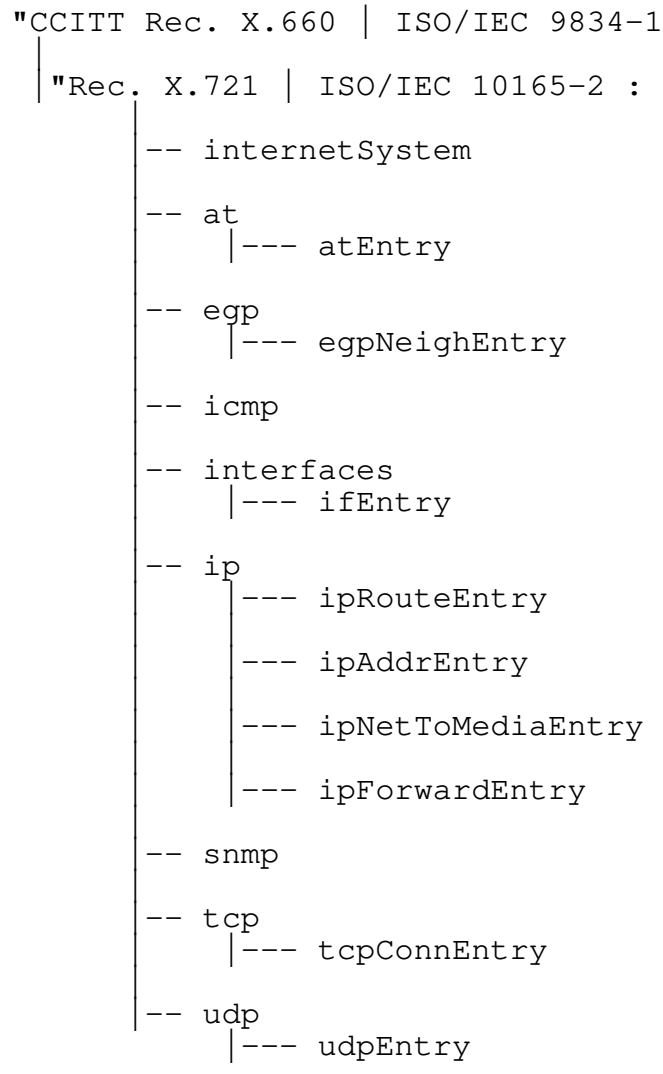
- For efficiency in proxies, filters would have to be examined before retrieval of remote objects to determine which object classes are actual candidates for the operation. In the example above, if the filter was not examined to eliminate non candidate object classes before retrieval of the scoped objects then much unnecessary data would have to be retrieved from the remote SNMP agents. Filtering in proxies may have to be performed before and after the retrieval process for efficiency.

- Relatively efficient native and proxy implementations could be built without the filtering capabilities.

Impact of Removing Table Objects on Containment Trees

Should table object classes be removed, then the conceptual table entries would be bound directly to the group with which they are associated. The structure of the resulting containment tree is illustrated below. Compare this tree with the one illustrated in section 3.3.3.

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



The revised containment tree which would result for the Internet MIB-II derived object classes [IIMCMIB-II] is illustrated below.