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

March 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 Background 1
1.2 Overview 2
1.3 Scope 4
1.4 Terms andConventions 5
2. Registration and Naming Procedures 5
2.1 Registration Procedures 5
2.1.1 Automated Registration Procedures 6
2.1.2 IIMC Explicit Registration Procedures 7
2.1.2.1 Object Classes and Attributes Registration 7
2.1.2.2 Trap/Notification Registration 7
2.1.2.3 NAME BINDINGs Registration 8
2.1.2.4 Registration of ASN.1 Modules and IIMC Documents 9
2.2 Naming Procedures 9
2.2.1 Naming Attribute 9
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 to Internet 11
2.3.2 OID/Name Translation Internet to ISO 14
2.4 Inheritance for Object Classes 14
2.5 Reference Labels for Derived Entities 15
3. Internet to ISO/CCITT MIB Translation Procedures 16
3.1 Pre-translation Procedures 16
3.2 GDMO Translation Procedures 19
3.2.1 Translation of Groups 19
3.2.2 Translation of Table Objects 21
3.2.3 Translation of Table Entry Objects 22
3.2.4 Translation of Other OBJECT-TYPES 23
3.2.5 Translation of Notifications 26
3.2.6 Translation of Internet Attribute Types 30

3.3 Post-translation Procedures 31
3.3.1 Post-translation of BEHAVIOUR Cause 32
3.3.2 Deletion of Derived MIB Elements 32
3.3.3 Creation of NAME BINDING Templates 32
4. Common SNMP Derived Attribute Types 35
5. ASN.1 Definitions 42
6. Acknowledgments 46
References 47
Annex A 50

Revision History

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

Draft 1 - March 26, 1993
Current draft of this document (replaces Draft 0).

Major Changes Since Last Revision

1. Deleted proxy MIB (moved to [IIMCPROXY]).
2. Revised OID translation procedure.
3. Revised generic notification replaces previous notifications.
4. Updated to reflect SNMPv2 changes.
5. Added parsing capability to templates.
6. Removed NOTIFICATIONS clause from templates. All notifications shall be emitted by the objects in the Proxy MIB defined in [IIMCPROXY].
7. Revised naming hierarchy for MIBs.
8. Only single name bindings may now be allowed for objects.
9. Revised pre and post processing procedures.
10. Defined document automatic and explicit registration procedures, and registered IIMC docs.
11. Added temporary annex describing the editor's proposal for the naming hierarchy relative to agents and a proxy. This proposal is reflected throughout this document.
12. Corrected numerous errors.

Action Item Proposals Contained In This Document

#13 Registration of documents (proposed).
#17 Parsable behaviour clause (proposed).
#7 & #12 Generic Internet notification (proposed).
#14 & #15 Alternate naming hierarchy (proposed).

Outstanding Issues

1. What should the naming hierarchy be, especially for proxy? This document contains in Annex A a description of the editor's proposed naming hierarchy and its characteristics. The [IIMCPROXY] contains a description of an alternative naming hierarchy proposal. Both proposals should be considered by reviewers, and comments are solicited.

2. Should multiple name-bindings be accommodated for object classes derived from Internet MIBs?

1.Introduction

The past decade has witnessed the development of enterprise wide networks composed of a multi-vendor environment containing heterogeneous protocol and hardware suites. Organizations have become increasingly dependent on these enterprise networks for their daily operations. This dependence has focused attention on the need for operation, administration, maintenance, and provisioning (OAM&P) of the multi-vendor enterprise network on an end-to-end basis.

1.1 Background

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

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

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 technical team.

This work was initiated, in part, by NM Forum efforts to translate RFC 1214 for use with OMNIPoint 1 implementations. Through this effort, it became obvious 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"

[NMFMC92]. The documents included in the IIMC package are the next level of detailed specifications which implement several of the methodologies identified in the strategy.

1.2 Overview

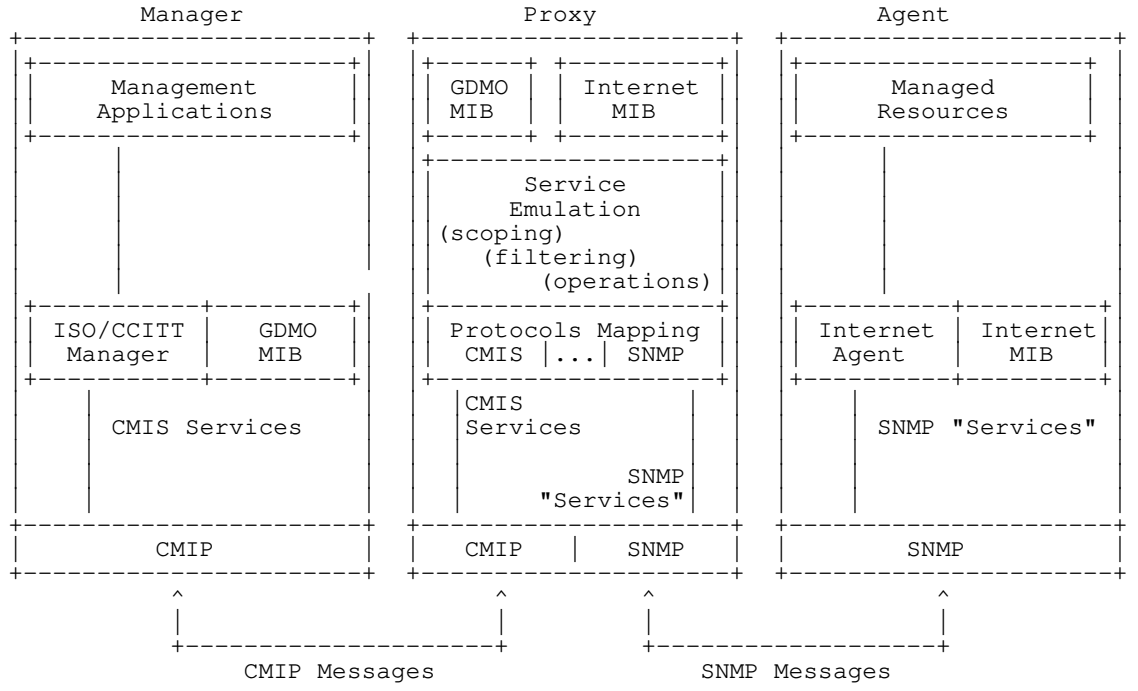
The response to the need for OAM&P of enterprise networks has been the development of network management standards within various networking communities - most notably the ISO/CCITT and Internet communities. However, coordination of standards activities between these two communities has not occurred. As a result, although they share a nearly common management model, differences in their management protocols and structures of management information (SMIs) have developed due to differing management philosophies.

The ISO/CCITT community has developed the Common Management Information Protocol (CMIP) [ISO9596-1], and related SMI documents [ISO10165-1,2,4]. The Internet community has developed the Simple Network Management Protocol (SNMP) [RFC1157], and its successor, SNMPv2 [SNMPv2PROT]. The Internet SMI is defined in [RFC1155] and [SNMPv2SMI]. Although functionally similar, the Internet and ISO/CCITT protocols and SMIs differ in terms of their complexity and specific operations.

The focus on the need for end-to-end enterprise management has indicated 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. One way to integrate management is by the development of "proxy" mechanisms which translate between functionally equivalent services, protocol and SMI differences to create this unified view.

A body of 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 integrated management model on the ISO/CCITT management model using CMIP and the ISO/CCITT SMI. These organizations are particularly interested in the development of proxies for devices that use the Internet management protocols and SMI. Their interest is primarily due to the widespread commercial implementation and use of such devices within their enterprises, especially devices that use the Internet TCP/IP protocol suite.

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



The proxy architecture 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 defined in [IIMCPROXY] uses these MIB definitions and rules to provide run-time translation of management information carried in service requests and responses.

The proxy architecture 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, SNMP and SNMPv2 are variants of the same protocol mapping process).

In addition, [IIMCOMIBTRANS] specifies translation procedures for converting ISO/CCITT GDMO MIBs into Internet MIBs. MIBs generated by this translation process cannot be utilized by the Proxy defined in [IIMCPROXY], although another kind of Proxy could be defined for this purpose in the future.

Finally, note that MIBs translated by procedures such as those defined by [IIMCIMIBTRANS] and [IIMCOMIBTRANS] may also be used without a proxy. For example, a translated MIB 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).

1.3 Scope

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 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.

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.

Editor's Note: [As of the date of this document, the SNMPv2 SMI and protocol are considered stable drafts. It is expected that the[SNMPv2PROT], [SNMPv2SMI], [SNMPv2MIB], [SNMPv2TC] and related documents will become RFCs and proposed Internet standards before the final publication of this document, and that changes will not significantly affect the registration, naming, and translation procedures described in this document.]

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

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

iimcIIMCIMIBTRANS OBJECT IDENTIFIER ::={iimcManagementDocMan 1}

Editor's Note: [The iimcManagementDocMan will be resolved before the final publication of this document.]

1.4 Terms and Conventions

Editor's Note: [To Be Provided.]

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 ISO/CCITT-Internet automated registration. The arc is named "iimcAutoTrans".

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

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 notifications and the IIMC proxy MIB.

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 5.

The following paragraphs describe 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 2.1 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) that is used to carry all Internet traps/notifications. This notification is to be emitted by the internetSystem managed object as defined in [IIMCMIB-II] and derived from the internet system group defined in RFC1213. However, each Internet defined trap/notification

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 [SNMPv2MIB].

For SNMPv1 trap PDUs, the snmpTrapOID is derived as stated in the SNMPv1/SNMPv2 Coexistence document [SNMPv2COEX] 4.1.2(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 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 3.2.5.

2.1.2.3 NAME BINDINGs Registration

As described in 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 name bindings 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 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)}

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. Documents derived from MIBs defined in Internet RFC are automatically registered under the iimcManagementModAutoarc 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 RFC are automatically 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.

Editor's Note: [Who is the registration authority for explicit registration of other documents?]

2.2 Naming Procedures

ISO/CCITT object 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, 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.

```
internetClassId ATTRIBUTE
          WITH ATTRIBUTE SYNTAX
                    IimcCommonDef.ObjectIdentifier;
```

```
            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.!;;
          MATCHES FOR     EQUALITY;
REGISTERED AS {iimcManagementNot 1};
```

The formats for naming attributes of table entries are 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.

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
}
```

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 2.1, and the structure of the naming attribute value was explained in 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) foripRouteNextHop 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.

However, 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 3.1, or because post processing procedures added new attributes.

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 [SNMPv2SMI].

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 83 217}, 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 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), either in the "cmipsnmpProxyAgent" RDN, or 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 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 3.2.5.

Given the definitions shown in 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 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 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)}.

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:

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

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

ii) the naming attribute value, and the internetClass OID defined in 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 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.1Externaltypereference or Externalvaluereference, 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

Externaltypereference. Default values must reference an Externalvaluereference.

(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 3.3.3). 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 <typek> 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". OBJECT-TYPES, if any, that are directly associated with the table become attributes of that table.

(f) Create an ASN.1 module for use in the GDMO template translations. 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 one of the Internet attribute types defined by the SNMPv1 or SNMPv2 SMI (e.g., Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32, Counter64, NsapAddress), or defined using an SNMPv2 TEXTUAL-CONVENTION macro, then do one of the following:

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

ii) If the value is an Externaltypereference: put the Externaltypereference syntax into the ASN.1 module.

g) If a DEFVAL clause is present, create an Externalvaluereference which has the type indicated by the OBJECT-TYPE SYNTAX clause and is assigned the value of the DEFVAL clause. The label for the Externalvaluereference shall be the attribute label preceded by "c_" (lower case letter "c"). Place the Externalvaluereference into the ASN.1 module created in f).

Note: automatic translation of some DEFVAL clauses into valid Externalvaluereferences may not be possible. Placeholders may be put into the ASN.1 module, e.g.,

reference label and DEFVAL clause value, for later manual translation during post processing. Also, repeated values may then be collapsed into a single reference.

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 Externalvaluereference.

Note: A syntax that is used more than once in the MIB to be translated shall be defined just once in the new ASN.1 module created in(f) and referenced repeatedly.

For convenience, an ASN.1 module of common definitions for Externaltypereferences of the basic ASN.1 types included in the SNMPv1 SMI and SNMPv2 SMI (e.g., INTEGER, OCTET STRING) is defined in 4.5. The Externaltypereferences in this module must either be imported into a local ASN.1 module within the document that defines the derived MIB, or appropriate equivalent references need to be declared within the local ASN.1 module.

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.

Note: other SNMPv2 SMI macro clauses contain textual or other information that may be inserted into the BEHAVIOUR clause of the an ISO/CCITT template, e.g., UNITS, REFERENCE. Provisions for including information in these macro clauses are not explicitly described in the translation procedures below, however the contents of these clauses should be included in the BEHAVIOUR clause.

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

```
[PARSE
[REFERENCE !!<text referencing internet document, group
     or object from which the ISO/CCITT object class was
     derived>!!;]
];
ENDPARSE ]
```

If used, the parsable 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	GET-REPLACE

<DEFVAL clause translation> - The value of the DEFVAL clause in the form of an Externalvaluereference, 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

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> - The text in the DESCRIPTION clause and any additional text deemed appropriate to defining the behaviour of the object class. To facilitate parsing of BEHAVIOUR clauses for classes derived from tables, the following parsable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[PARSE
[REFERENCE !!<text referencing the internet document,
     group or object from which the ISO/CCITT object
     class was derived>!!;]
];
ENDPARSE ]
```

If used, the parsable 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 DESCRIPTION clause text> - The text in the DESCRIPTION clause and any additional text deemed appropriate to defining the behaviour of the object class. To facilitate parsing of BEHAVIOUR clauses for classes derived from table entries, the following parsable structure is recommended (the [] indicate optionality, keywords must be in caps, keywords shall be excluded from the descriptive text):

```
[PARSE
[REFERENCE !!<text referencing internet document, group
or object from which the ISO/CCITT object class was
derived>!!;]
[MULTIPLEINSTANCES
     INDEX <attribute label, ..., attribute label>;
     [AUGMENTS <entry label that the object class
               augments>;]
     [CREATEDELETEATT <label of attribute used for
               creation and deletion of entries>;]
     [CREATEDELETEVALUE SNMPV2ROWSTATUS | <value of
     create/delete attribute indicating deletion>;]
ENDMULTIPLEINSTANCES]
ENDPARSE]
```

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

The SNMPV2ROWSTATUS keyword indicates that the definition of the attribute type rowStatus (see 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.

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 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 3.2.6 for translation procedures associated with the TEXTUAL CONVENTIONS macro.

Attributes of the defined SNMP types (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 for the conversion of these types into ISO/CCITT attribute types. In addition, 4 contains ISO/CCITT attribute type definitions derived from [SNMPv2TC].

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 Externaltypereference notation as GDMO

requires.

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

Note: This normally is a manual process; however matching rules that generally apply to attributes having a simple universal ASN.1 syntax are provided in the table below. These rules also to Externaltypereferences 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
SEQUENCE	EQUALITY

See 4.3 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 (Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32, Counter64, NsapAddress), and Internet attribute types defined using the SNMPv2 TEXTUAL CONVENTIONS macro, shall be treated as ISO/CCITT attribute types. Specific attributes are derived from these types.

The following table indicates the translations required for Internet attribute types as defined either in this document or Internet documents. ISO/CCITT attribute types corresponding to the following Internet attribute types are defined in 4.

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
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

<OBJECT-TYPE DESCRIPTION clause text> - The text in the DESCRIPTION clause and any additional text deemed appropriate to defining the behaviour of the attribute. To facilitate parsing of BEHAVIOUR clauses for

attributes derived from internet objects, the following parsable 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>;]
[UNITS <text indicating the units associated with the
     attribute>;]
[DISPLAY-HINT <as defined in SNMPv2TC>;
[DEFVAL <default value for attribute>;]
ENDPARSE]
```

If used, the parsable 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 [SNMPv2SMI] defines a NOTIFICATION macro and its mapping onto an SNMPv2 PDU. The document on SNMPv1/SNMPv2 Coexistence [SNMPv2COEX] 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].

```
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.!;;

```
 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};

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 1};


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 2};

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 3};
```

The referenced syntax is included in the IIMCCommonDef ASN.1 module in 5.

3.2.6 Translation of Internet Attribute Types

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

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

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

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

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

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> - Text from the DESCRIPTION clause and any additional text deemed appropriate to defining the behaviour of the attribute. To facilitate parsing of BEHAVIOUR clauses for attributes derived from internet objects, the following parsable 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>;]
[UNITS <text indicating display hint for values of
     attribute>;]
[DEFVAL <default value for attribute>;]
ENDPARSE]
```

If used, the parsable 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. However, sometimes the procedures are not so straightforward.

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 2.2, or the assignment of notifications to be emitted by the object class.

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.

Text descriptions of events that may trigger the emission of notifications should be included.

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 parsable 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 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;
              [CREATE [<create modifier>];]
              [DELETE [<delete-modifier>];]
```

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.

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.

<create modifier> - the "WITH-AUTOMATIC-INSTANCE-

NAMING" modifier may be part of the CREATE clause. The "WITH-REFERENCE-OBJECT" may also be used.

<delete-modifier> - The delete modifier for the DELETE clause shall be "DELETES-CONTAINED-OBJECTS" if the table entry contains an object class added as a result of an SNMPv2 OBJECT-TYPE AUGMENTS clause.

<name binding OID> - As defined in 2.1.3.

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

Name bindings for ISO/CCITT table and entry type object classes derived from Internet MIBs shall be automatically inferred from the Internet registration hierarchy. Thus 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 table object classes with which they are associated.

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

3.3.4 Post-processing of ASN.1 Module

Automatic translation of some DEFVAL clauses into valid Externalvaluereferences may not have been possible. Placeholders may have been put into the ASN.1 module, e.g., reference label and DEFVAL clause value, for later manual translation during post-processing.

It may be possible to collapse repeated values 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. Common SNMP Derived Attribute Types

Internet Attribute types defined by the SNMPv2 TEXTUAL-CONVENTION macro are translated into ISO/CCITT attribute types. Attributes of the defined SNMP types (e.g., Counter, IpAddress, Gauge, TimeTicks, Opaque, Counter32, Gauge32), which could also have been defined in a TEXTUAL-CONVENTION macro, are also considered to be Internet attribute types.

ISO/CCITT Attribute templates derived from these types shall contain the DERIVED FROM clause.

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.OctetString;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
          autonomousTypeBehaviour BEHAVIOUR
          DEFINED AS
          !Represents an independently extensible type
          identification value.  It may, for example,
              indicate a particular subtree with further MIB
              definitions, or define a particular type of
              protocol or hardware.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

counter32 ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.Counter32;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          counter32Behaviour BEHAVIOUR
          DEFINED AS
          !As defined for counter type in ISO/IEC 10165-2.
          Only the value range constraint (0..4294967295) is
          added.  This corresponds to the type defined in
          [SNMPv2SMI] by the same name!;;

counter64 ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.Counter64;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          counter64Behaviour BEHAVIOUR
          DEFINED AS
          !As defined for counter type in ISO/IEC 10165-2.
           Only the value range constraint
           (0..18446744073709551615) is added.
          This corresponds to the type defined in
[SNMPv2SMI]
```

```
          by the same name!;;;

dateAndTime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.DateAndTime;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          dateAndTimeBehaviour BEHAVIOUR
          DEFINED AS
          !DISPLAY-HINT "2d-1d-1d,1d:1d:1d.1d,1a1d:1d"
          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.

          This corresponds to the type defined in
          [SNMPv2SMI] by the same name!;;;

displayString ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcCommonDef.OctetString;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
          displayStringBehaviour BEHAVIOUR
          DEFINED AS
          !DISPLAY-HINT "255a"
     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.  This
     corresponds to the type defined in [SNMPv2TC] by
     the same name.!;;;

gauge32 ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.Gauge32;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          gauge32Behaviour BEHAVIOUR
```

```
          DEFINED AS
          !As defined for the integer gauge type in ISO/IEC
          10165-2.  Only the value range constraint
           (0..4294967295) is added.
          This corresponds to the type defined in
          [SNMPv2SMI] by the same name!;;;

instancePointer ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcCommonDef.ObjectIdentifier;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          instancePointerBehaviour BEHAVIOUR
          DEFINED AS
          !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.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

ipAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.OctetString;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
          ipAddressBehaviour BEHAVIOUR
          DEFINED AS
          !This attribute represents a 32-bit internet
          address.  It is represented as an octet string
          of length 4, in network Byte-order.
          This corresponds to the type defined in
          [SNMPv2SMI] by the same name!;;;

macAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.MacAddress;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
          macAddressBehaviour BEHAVIOUR
          DEFINED AS
          !DISPLAY-HINT "1x:"
          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.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

nsapAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.OctetString;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
```

```
          nsapAddressBehaviour BEHAVIOUR
          DEFINED AS
          !This attribute represents an ISO/CCITT network
          address.  It is represented as a variable
          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 of 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.  This corresponds to the type
          defined in [SNMPv2SMI] by the same name!;;;

opaque ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.OctetString;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          opaqueBehaviour BEHAVIOUR
          DEFINED AS
          !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.  This corresponds to the
          type defined in [SNMPv2SMI] by the same name.!;;;

physAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.OctetString;
     MATCHES FOR    EQUALITY, ORDERING, SUBSTRINGS;
     BEHAVIOUR
          physAddressBehaviour BEHAVIOUR
          DEFINED AS
          !DISPLAY-HINT "1x:"
          Represents media- or physical-level addresses.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

rowStatus ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IimcCommonDef.RowStatus;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          rowStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The RowStatus textual convention is used to
           manage the creation and deletion of conceptual
           rows, and is used as the value of the SYNTAX
           clause for the status column of a conceptual row
```

(as described in Section 7.7.1 of [2].)

The status column has six defined 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;

- `createAndGo', which is supplied by a management station wishing to create a new instance of a conceptual row and to have it available for use by the managed device;

- `createAndWait', which is supplied by a management station wishing to create a new instance of a conceptual row but not to have it available for use by the managed device; and,

- `destroy', which is supplied by a management station wishing to delete all of the instances associated with an existing conceptual row.

Whereas five of the six values (all except `notReady') may be specified in a management protocol set operation, only three values will be returned in response to a management protocol retrieval operation: `notReady', `notInService' or `active'. That is, when queried, an existing conceptual row has only three states: it is either available for use by the managed device (the status column has value `active'); it is not available for use by the managed device, though the agent has sufficient information to make it so (the status column has value `notInService'); or, it is not available for use by the managed device, (the status column has value `notReady').

To summarize the effect of having a conceptual row with a status column having a SYNTAX clause value of RowStatus,consider the following state diagram:

	STATE			
ACTION	A status column does not exist	B status col. is notReady	C status column is notInService	D statuscolumn isactive
set status column to createAndGo	noError ->D or inconsistent-Value	inconsist-entValue	inconsistent-Value	inconsistent-Value
set status column to createAndWait	noError see 1 or wrongValue	inconsist-entValue	inconsistent-Value	inconsistent-Value
set status column to active	inconsistent-Value	inconsist-entValue or see 2 ->C	noError ->D	noError ->D
set status column to notInService	inconsistent-Value	inconsist-entValue ->C	noError ->C	noError ->C or wrongValue
set status column to destroy	noError ->A	noError ->A	noError ->A	noError ->A
set any other column to some value	see 3 ->A	noError see 1	noError ->C	noError ->D

(1) go to B or C, depending on information available to the agent.

(2) if other variable bindings in the same PDU provide values for any missing but required columns, then noError is returned and state C is entered.

(3) at the discretion of the agent, either noError or inconsistentValue may be returned.

NOTE: other processing of the set request may prevent a noError response from being returned, e.g., wrongValue, noCreation, etc.

See [SNMPv2TC] for a description of the algorithm conceptual row creation and deletion.

This corresponds to the type defined in [SNMPv2TC] by the same name.!;;;

```
testAndIncrement ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
     MATCHES FOR    EQUALITY, ORDERING;
     IimcCommonDef.TestAndIncrement;
          BEHAVIOUR
          testAndIncrementBehaviour BEHAVIOUR
          DEFINED AS
          !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.
        This corresponds to the type defined in [SNMPv2TC]
        by the same name.!;;;


timeInterval ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IimcCommonDef.Integer32;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          timeIntervalBehaviour BEHAVIOUR
          DEFINED AS
          !A period of time, measured in units of 0.01
          seconds.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

timeStamp ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IimcCommonDef.TimeTicks;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          timeStampBehaviour BEHAVIOUR
          DEFINED AS
          !The value of MIB-II's sysUpTime object at which a
          specific occurrence happened.  The specific
          occurrence must be defined in the description of
          any object defined using this type.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

timeTicks ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcCommonDef.TimeTicks;
```

```
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          timeTicksBehaviour BEHAVIOUR
          DEFINED AS
          !This attribute 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. This corresponds to the type
          defined in [SNMPv2SMI] by the same name!;;;

truthValue ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.TruthValue;
     MATCHES FOR    EQUALITY;      BEHAVIOUR
          truthValueBehaviour BEHAVIOUR
          DEFINED AS
          !Represents a boolean value.
          This corresponds to the type defined in [SNMPv2TC]
          by the same name.!;;;

uInteger32 ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IimcCommonDef.UInteger32;
     MATCHES FOR    EQUALITY, ORDERING;
     BEHAVIOUR
          uInteger32Behaviour BEHAVIOUR
          DEFINED AS
          !As defined for the ASN.1 Builtin INTEGER type.
          Only the value range constraint (0..4294967295) is
          added.  This corresponds to the type
          defined in [SNMPv2SMI] by the same name!;;;
```

5. ASN.1 Definitions

```
IimcAssignedOIDs  {iimcManagementModMan 1}
DEFINITIONS ::= BEGIN
iimcAutoTrans OBJECT IDENTIFIER ::= {...}

iimcManagement OBJECT IDENTIFIER ::= {...}

iimcManagementNB  OBJECT IDENTIFIER ::= {iimcManagement 1}
     -- for ISO/CCITT derived NAME BINDINGS

iimcManagementModule OBJECT IDENTIFIER ::=
                                        {iimcManagement 2}
     -- for ISO/CCITT Translation ASN.1 Modules

iimcManagementModAuto OBJECT IDENTIFIER ::=
                                        {iimcManagement 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 1}
     -- 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
          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)}
          ObjectName, ObjectSyntax
          FROM RFC1155-SMI
          VarBind, VarBindList
          FROM RFC1098-SNMP
          Response-PDU
          FROM SNMPv2-PDU;
Integer ::= INTEGER
OctetString ::= OCTET STRING
ObjectIdentifier ::= OBJECT IDENTIFIER
Null ::= NULL
BitString ::= BIT STRING
Counter32 ::= INTEGER (0..4294967295)
Counter64 ::= INTEGER (0..18446744073709551615)
DateAndTime ::= OCTET STRING (SIZE (8 | 11))
Gauge32    ::= INTEGER (0..4294967295)
Integer32 ::=  INTEGER (2147483648..2147483647)
InternetAlarmInfo ::= SEQUENCE     {
     probableCause           ProbableCause,
     attributeIdList     [1] AttributeIdentifierList
                                          OPTIONAL,
     objectInstanceList  [2] ObjectInstanceList
                                     OPTIONAL,
     unknownVarBindList  [3] VarBindList
                                          OPTIONAL,
     internetTrapInfo    [4] InternetTrapInfo
                                          OPTIONAL,
     perceivedSeverity   [5] PerceivedSeverity
                                          OPTIONAL,
     notificationId      [6] NotificationIdentifier
                                          OPTIONAL,
     correlatedNot       [7] CorrelatedNotification
                                          OPTIONAL,
     transportDomain     [8] TransportDomain      OPTIONAL,
     transportAddress    [9] TransportAddress     OPTIONAL,
     accessControlInfo   [9] AccessControlInfo    OPTIONAL
                              }

AccessControlInfo   CHOICE    {
     communityString     [0] OCTET STRING,
     partyInfo           [1] SEQUENCE   {
                              srcParty  OBJECT IDENTIFIER,
                              dstparty OBJECT IDENTIFIER,
                              context   OBJECT IDENTIFIER
                                        }
                              }
InternetTrapInfo ::= SET OF {
                         SEQUENCE {
                              objectInstance ObjectInstance
                                          OPTIONAL,
                              COMPONENTS of Attribute}}
TimeTicks ::=  INTEGER (0..4294967295)
```

```
MacAddress ::= OCTET STRING (SIZE (6))
TransportDomain ::= OBJECT IDENTIFIER
TransportAddress ::= OCTET STRING

ObjectInstanceList  ::=  SET of ObjectInstance
TruthValue ::= INTEGER { true(1), false(2) }
TestAndIncrement ::= INTEGER (0..2147483647)
RowStatus ::=  INTEGER   {
                    -- the following two values are states:
                    -- these values may be read or written
                    active(1), notInService(2),

                    -- the following value is a state:
                    -- this value may be read but not written
                    notReady(3),

                    -- the following three values are
                    -- actions:  these values may be written,
                    -- but are never read
                    createAndGo(4),
                    createAndWait(5),
                    destroy(6)
                         }

UInteger32 ::= INTEGER (0..4294967295)
END
```

6. Acknowledgments

The following individuals have contributed to this effort.

Bob Aronoff - NIST
Jon Biggar - NetLabs
Mary Brady - NIST
April Chang - NetLabs
Jock Embry - Opening Technologies
Paul Golick - IBM
Pramod Kalyanas - University of Delaware
Lee LaBarre - The MITRE Corporation
David Liu - Northern Telecom, Inc
Owen Newnan - U S West Advanced Technologies
Steve Ng - MPR Teltech
Yasuhiro Ohara - NTT
George Pavlou - UCL
Lisa Phifer - Bellcore
Tom Rutt - AT&T
Mark Smith - Hewlett-Packard
Einar Stefferud - Network Management Associates, Inc.
Dean Voiss - NetLabs
Yoshi Yamashita - NKK Corporation

References

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

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

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

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

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

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

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

[ISO10165-4] ISO/IEC IS 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] RFC 1157, 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-basedinternets: MIB-II, March 1991.

[RFC1214] RFC1214, L. LaBarre - editor, OSI Internet Management: Management Information Base, April 1991.

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

[SNMPv2COEX] J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Coexistence between version 1 and version 2 of the Internet Network Management Framework, Internet-draft, December 1992.

[SNMPv2PROT] J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-draft, January 1992.

[SNMPv2SMI] 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), Internet-draft, December 1992.

[SNMPv2MIB] J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-draft, December 1992.

[SNMPv2TC] J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-draft, December 1992.

[SNMPv2ADMIN] J.R. Davin, J.M. Galvin, K.McCloghrie, Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-Draft, January 1993.

[SNMPv2SEC] J.M. Galvin, K. McCloghrie, J.R. Davin, Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-Draft, January 1993.

[SNMPv2TM] J.D. Case, K. McCloghrie, M.T. Rose, S.L. Waldbusser, Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-Draft, January 1993.

[SNMPv2PARTY] J.D. Case, K. McCloghrie, M.T. Rose, S.L. Waldbusser, Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-Draft, January 1993.

[IIMCSEC] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Security, Draft 1 March 26,1993.

[IIMCMIB-II] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB, Draft 1, March 26, 1993.

[IIMCPROXY] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Proxy, Draft 1, March, 1993 [to be distributed].

[IIMCOMIBTRANS] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of ISO/CCITT GDMO MIBs to Internet MIBs, Draft 1, March 26, 1993.

[NMFMC92] NM Forum and X/Open, ISO/CCITT and Internet Management: Coexistence and Interworking Strategy, October, 1992.

Annex A Editor's Naming Proposal

Naming is still an open issue, especially as regards naming within the proxy. The naming hierarchies below illustrate the editor's proposal for naming. The assumption is that a "native" CMIP agent that happened to be managing Internet resources would have one instantiation of the naming tree that is particular to that agent. A proxy implementation would have instantiated ISO system objects for each agent it is proxying to, and the MIB specific to the proxy.

Some characteristics of this naming hierarchy are discussed below.

```
system A {systemId = "A"}          system B {systemId = "B"}
        |                                  |
        |- internetSystem                  |- internetSystem
        |- tcp                             |- tcp
        |- partyTable                      |- partyTable
        |- contextTable                    |- contextTable
        |- aclTable                        |- aclTable
        |- viewTable                       |- viewTable
        |- etc.                            |- etc.
```

```
system Proxy {systemId = "proxy@x.y"}
        |
        |- cmipsnmpProxyTable
        |      |- cmipsnmpProxyAgent
        |- partyTable
        |- contextTable          (manager party MIB)
        |- aclTable
        |- viewTable
        |- EFD
        |- Log
        |- internetSystem *
        |- tcp *
        |- etc. *
```

Characteristics:

1. From perspective of ISO manager all agents are alike in their naming hierarchy, whether via proxy, or native CMIP implementation,

2. All objects under the system object for agents are "virtual" in that they are not instantiated in the proxy,

3. All objects under the system object for proxy are instantiated in the proxy, including those relative to protocol resources that may be managed in the proxy(*),

4. Proxy would instantiate all ISO system objects,

5. The party MIB objects in the proxy are equivalent to those found in an Internet Manager,

6. The party MIB in the proxy may minimally consist of the partyTable and its entries. The entries would consist of those specific to parties residing in the proxy, and copies of those parties stored in the internet agents and used for communication. The context acl, and views tables may not be required unless access control is applied to the MIB instantiated in the proxy, or unless the proxy applies access control on behalf of Internet agents, e.g., SNMPv1 agents.

7. Internet style access control may be applied to the MIB elements in the proxy, e.g., to the party MIB elements.

8. The internet style use of the "local" variables for context and view tables need not be used. All requests concerning objects under a system object particular to an agent are considered to be relative to the remote agent.

9. If the Proxy MIB, i.e., cmipsnmpProxyTable and cmipsnmpProxyAgent, have OIDs assigned using the Internet conventions, then it is possible to apply Internet style access control to the Proxy MIB elements as well. This may be desirable since then the manager would have to deal with only one style of access control. All of the party MIB tables may have to be implemented.

10. OSI access control may be applied at the proxy for the entire MIB, both for the MIB instantiated within the proxy, and those instantiated in the agents.

11. EFDs and Logs are instantiated in the proxy. Events translated from SNMP traps (for all agents) may be forwarded over single associations to each ISO Manager. This means that there does not have to be an EFD associated with each agent, although there may be. Events may be forwarded to specific local or remote logs depending on the filtering characteristics.

12. A proxy implementation could work in two modes: transparent, and non-transparent.

Transparent Mode: the manager establishes associations to the proxy on a per agent (or proxy) basis, just as if the proxy function was not there. The transparent

mode may be indicated by the fact that the systemId value used in the AETitle for the association contains the systemId value associated with an agent (or proxy). Once the mode has been established for the association, and the agent (or proxy), one particular tree among the forest of trees has been selected, i.e., access to all other trees may be prohibited over that association, and the local DN form could possibly be used.

Non-transparent Mode: the manager establishes a single association to the proxy, however individual CMIP requests/responses can apply to individual agents, as distinguished by their systemId value in the DN for the object. The transparent mode may be indicated by the fact that the systemId value used in the AETitle for the association contains the systemId value associated with the proxy. The global DN form would always be used in CMIP PDUs.

An alternative way of indicating the mode of operation may be by the use of different application contexts. The transparent mode could be the application context registered in SMO. The non-transparent mode could be a new application context.