ISO/CCITT and Internet Management Coexistence (IIMC):

Translation of Internet Party MIB (RFC1353)

to

ISO/CCITT GDMO MIB

(IIMCPARTY)

9 October, 1992

Lee LaBarre

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

Status of this Memo

This memo provides information to the network and systems management community. This memo is intended as a contribution to ongoing work in the area of multi-protocol management coexistence and interworking. This memo is part of a package of ISO/CCITT and Internet Management Coexistence (IIMC) drafts; see also [IICMIMIBTRANS] [IIMCOMIBTRANS] [IIMCPARTY] [IIMCPROXY].

{Editor's Note: This memo is incomplete and requires thorough review in terms of MIB use, content, initial values, and adaptation for use with SNMP community strings. Comments are solicited.}

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.

Distribution of this memo is unlimited. Comments on this memo should be sent to iimc@thumper.bellcore.com by November 20, 1992.

Abstract

This memo is intended to facilitate the multi-protocol management coexistance and interworking for networks that are managed using the OSI Common Management Information Protocol (CMIP) and networks that are managed using the Simple Network Management Protocol (SNMP). This RFC contains the OSI definition and registration of the IIMC SNMP Parties MIB as derived from the Internet SNMP Parties MIB (RFC1353) according to the procedures defined in "Translation of Internet MIBs for CMIP/SNMP and SMP Proxy" [IIMCMIBTRANS].

Table of Contents

Status of this Memo i
Abstract ii
Table of Contents ii
1. Introduction 1
1.1 Background 1
1.2 Overview 2
1.3 Purpose and Scope 4
1.4 Terms and Conventions 4
2. Object Class Definitions 4
3. Attribute Definitions 10
4. Notifications 18
5. The Containment Hierarchy 18
6. ASN.1 Definitions 22
7. Use of Party MIB 25
7.1 Initial Values for Proxy/Agent Secure Communications 25
7.2 Authentication and Access Control 25
7.3 Integrity and Confidentiality 25
8. Acknowledgements 26
References 26

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 focussed 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 memo is part of a package of ISO/CCITT and Internet Management Coexistence (IIMC) drafts. Other memos included in this package are:

- Translation of Internet MIBs to ISO/CCITT GDMO MIBs (LaBarre) [IIMCIMIBTRANS]

- Translation of ISO/CCITT GDMO MIBs to Internet MIBs (Newnan) [IIMCOMIBTRANS]

- Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB (LaBarre) [IIMCMIB-II]

- ISO/CCITT to Internet Management Proxy (Chang) [IIMCPROXY]

These memos together comprise a package aimed at integrating ISO/CCITT-based and Internet-based management systems. These memos are offered as input to coexistence and interworking efforts underway throughout the industry,including organizations such as:

- IETF OSI Internet Management (OIM),
- Network Management Forum Technology Convergence Team,
- X/Open Systems Management (SysMan),
- OIW Network Management Special Interest Group (NMSIG), and
- OSF Management Special Interest Group (MANSIG).

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 memos included in the IIMC package are intended as detailed specifications which implement several of the methodologies identified in this strategy.

1.2 Overview

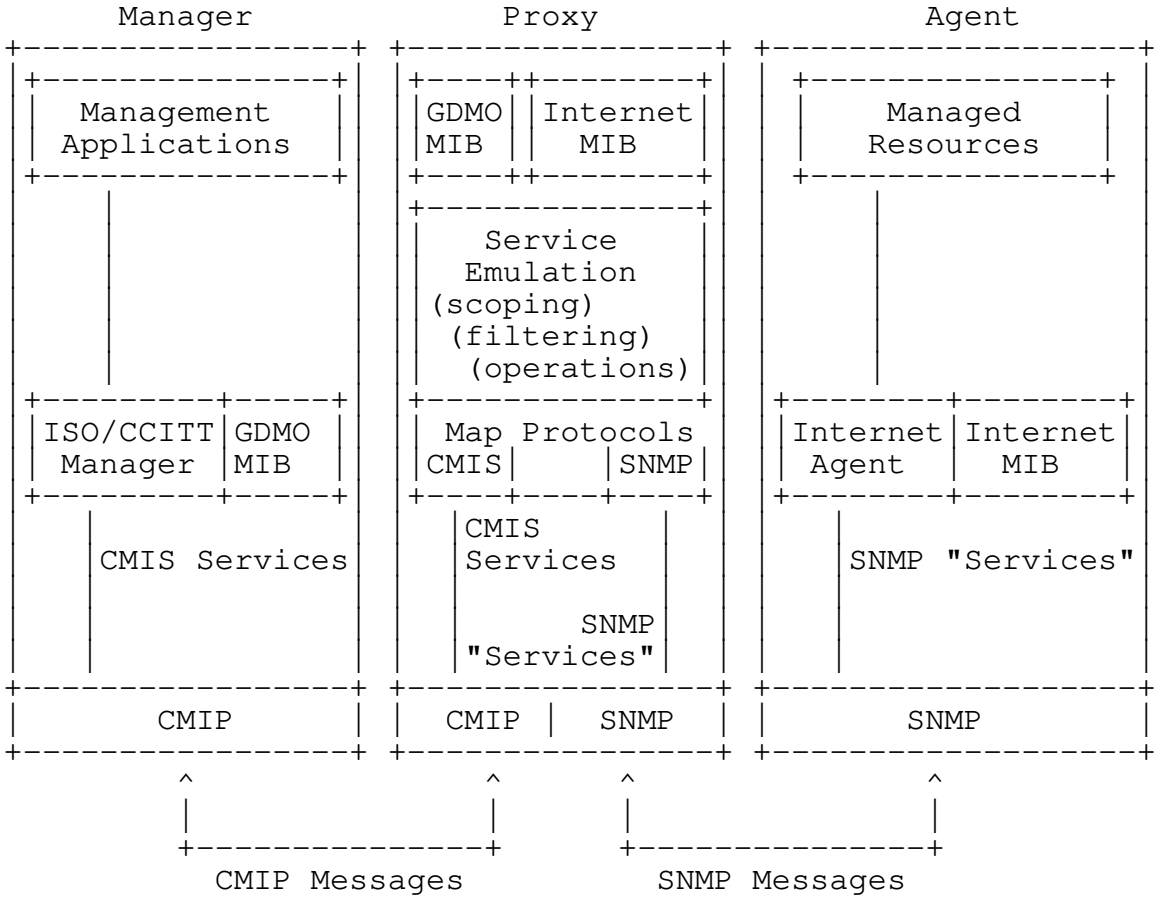
The response to the need for OAM&P of enterprise networks has been the development of network management standards within various networking communities - most notably the ISO/CCITT and Internet community. However, coordination of standards activities between these two communities has not occurred. As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

The ISO/CCITT community has developed the Common Management Information Protocol (CMIP) [ISO9596], and related SMI documents [ISO10165-1,3,4]. The Internet community has developed the Simple Network Management Protocol (SNMP) [RFC1157], and is developing its successor, SNMP-2, based on [SMPPROT]. The Internet SMI is defined in [RFC1155] and [SMPSMI]. 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 managed 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 [IIMCMIBTRANS]. 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, Secure SNMP, and SNMP-2 are all variants of the same protocol mapping process).

Finally, [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.

1.3 Purpose and 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 CMIP/SNMP proxies require the translation of Internet MIBs defined according to the Internet Structure of Management Information (SMI) [RFC 1155] [SMPSMI] into MIBs defined according to the ISO/CCITT SMI [ISO10164-1] and Guidelines for the Definition of Managed Objects (GDMO) [ISO10165-4]. Procedures for such translations are described in [IIMCIMIBTRANS].

This memo applies the procedures described in [IIMCMIBTRANS] to the translation and registration of the Internet SNMP Parties MIB defined in [RFC1353].

This memo assumes that the reader is familiar with the Internet and ISO/CCITT SMIs and terminology as well as the Internet to SMI translation defined in [IIMCIMIBTRANS].

1.4 Terms and Conventions

TBD

2. Object Class Definitions

{Editor's Note: RFC1353 identifies two groups: snmpParties and snmpSecrets and assigns them separate OIDs. This was necessary for the Internet SMI in order to control access to these "groups" on the basis of their OIDs. These two groups were not made into OSI object classes since they do not contain attributes and they do not assist in the identification or scoping of information in the OSI context. This is not in strict accordance with the IIMC MIB Translation

procedures.}

The Internet SNMP Parties MIB objects [RFC1353] are recast into OSI GDMO templates as defined in [ISO10165-4], and registered, using the procedures defined in [IIMCMIBTRANS].

According to [IIMCIMIBTRANS], OIDs registered under the internet arc are of the form:

OID = <internet> <internetEntityId>

where <internet> is the full registration path to the "internet" arc; and <internetEntityId> is the portion of the OID that uniquely identifies entities under that arc, i.e., the remainder of the OID.

Re-registration of objects is accomplished by replacing the <internet> portion of the OID with a new registration arc allocated for proxy registration such that the OID is of the form:

OID = <cmipsnmpProxyXX> <internetEntityId>

Accordingly, object class OIDs assigned in this document to [RFC1353] tables and entries are:

```
cmipsnmpProxyIMIB
    |
    +--- mgmt (2) --- mib-2 (1)  --- partyTable (20 2 1)
                          --- partyEntry (20 2 1 1)
                          --- aclTable (21 2 1)
                          --- aclEntry (21 2 1 1)
                          --- viewTable (21 3 1)
                          --- viewEntry (21 3 1 1)
```

OIDs for additional object classes and attributes are extended from the above OIDs as is done in the corresponding Internet definitions. The object identifier {cmipsnmpProxyIMIB} is defined in [IIMCIMIBTRANS].

The object identifier {cmipsnmpProxyIMIB} is defined in [IIMCIMIBTRANS].

The templates for the object classes are listed in alphabetical order.

```
aclEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
          CHARACTERIZED BY
               aclEntryPkg PACKAGE
                    BEHAVIOUR
          aclEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
```

```
          !The access privileges for a particular requesting
           SNMP party in accessing a particular target SNMP
          party.
               MULTIPLE INSTANCES
               INDEX  { aclTarget, aclSubject }
               STATUSVAR ::= aclStatus
               STATUSDELETE ::= 2 !;;
          ATTRIBUTES
               "IIMCIMIBTRANS": internetClassId GET,
               aclTarget                    GET-REPLACE,
               aclSubject                   GET-REPLACE,
               aclPrivileges                GET-REPLACE
                    DEFAULT VALUE IIMCRFC1353ProxyMIB.C3,
               aclStatus                    GET-REPLACE
                    DEFAULT VALUE
                         IIMCRFC1353ProxyMIB.Valid;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 21 2 1 1};

aclTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
          CHARACTERIZED BY
               aclTablePkg PACKAGE
                   BEHAVIOUR
               aclTableBehaviour BEHAVIOUR
          DEFINED AS !The access privileges database.!;;
          ATTRIBUTES
               "IIMCIMIBTRANS": internetClassId       GET;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 21 2 1};


partyEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
            CHARACTERIZED BY
               partyEntryPkg PACKAGE
                   BEHAVIOUR
          partyEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !NOTE: This object class deviates from the
          straightforward derivation of object classes from
          conceptual table entries according to [IIMCIMIBTRANS].
          It combines two conceptual table entries: the Internet
          "partyEntry" and the Internet PartySecretsEntry
          information.  In this aspect it is in agreement with
          RFC1351, which does not explicitly distinguish between
          public and secret information.

          The split between public and secret information is an
          artifact of the Internet access control mechanisms. The
          result is that entries in the Internet partyTable must
          be created/deleted as a side-effect of the
          creation/deletion of corresponding entries in the
          partySecretsTable.

          Locally held secret information about a
```

particular SNMP party, which is available for access by network management, is held in the partySecretsAuthPrivate and partySecretsPrivPrivate attributes. These attributes correspond to Internet objects in the SNMP partySecretsEntry to the partySecretsTable - for which no corresponding IIMC object classes are defined.

The creation/deletion of instances of this object class requires that corresponding Internet partyEntry and partySecretEntry conceptual rows be created/deleted simoultaneously in the Internet MIB representation.

A CREATE/DELETE request must specify at least one of the partySecretAuthPrivate and partySecretsPrivPrivate attributes, and one other party attribute, besides the name attribute. This is to ensure that an ISO/CCITT-Internet proxy will be able to synchronize the update of the Internet representations of conceptual table entries for the partyTable and partySecrets table.

For proxy, the value of partySecretsIdentity and partyIdentity are the same; the value of partySecretsStatus and partyStatus are the same.

Note that this table does not include all locally held information about a party. In particular, it does not include the 'last-timestamp' (i.e., the timestamp of the last authentic message received) or the 'nonce' values.

```
     MULTIPLE INSTANCES
     INDEX {partyIdentity}
     STATUSVAR ::= partyStatus
     STATUSDELETE ::= 2 !;;
ATTRIBUTES
     "IIMCIMIBTRANS": internetClassId   GET,
     partyIdentity            GET-REPLACE,
     partyTDomain             GET-REPLACE
          DEFAULT VALUE
          IIMCRFC1353ProxyMIB.RFC1351Domain,
     partyTAddress            GET-REPLACE
          DEFAULT VALUE
          IIMCRFC1353ProxyMIB.RFC1351DefaultTransport,
     partyProxyFor            GET-REPLACE
          DEFAULT VALUE
          IIMCRFC1353ProxyMIB.NoProxy,
     partyAuthProtocol        GET-REPLACE
          DEFAULT VALUE
          IIMCRFC1353ProxyMIB.MD5AuthProtocol,
     partyAuthClock           GET-REPLACE
          DEFAULT VALUE
```

```
                    IIMCRFC1353ProxyMIB.Zero,
               partyAuthPublic          GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.EmptyString,
               partyAuthLifetime        GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.C300,
               partyPrivProtocol        GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.NoPriv,
               partyPrivPublic          GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.EmptyString,
               partyMaxMessageSize      GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.C484,
               partyStatus              GET
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.Valid,
               partySecretsAuthPrivate  GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.EmptyString,
               partySecretsPrivPrivate  GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.EmptyString;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 20 2 1 1};

partyTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
          CHARACTERIZED BY
               partyTablePkg PACKAGE
                   BEHAVIOUR
               partyTablePkgBehaviour BEHAVIOUR
          DEFINED AS
          !The SNMP Party public and secret database.

          The entries in this table contain the information
          specified for the partyEntry and partySecretsEntry.

          See the partyEntry object class.!;;
          ATTRIBUTES
               "IIMCIMIBTRANS": internetClassId   GET;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 20 2 1 };

viewEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
          CHARACTERIZED BY
               viewEntryPkg PACKAGE
                   BEHAVIOUR
          viewEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !Information on a particular family of view
          subtrees included in or excluded from a particular
          SNMP party's MIB view.
```

```
               MULTIPLE INSTANCES
               INDEX  { viewParty, viewSubtree }
               STATUSVAR ::= viewStatus
               STATUSDELETE ::= 3 !;;
          ATTRIBUTES
               "IIMCIMIBTRANS": internetClassId   GET,
               viewParty                GET-REPLACE,
               viewSubtree              GET-REPLACE,
               viewStatus               GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.Included,
               viewMask                 GET-REPLACE
                    DEFAULT VALUE
                    IIMCRFC1353ProxyMIB.EmptyString;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 21 3 1 1};


viewTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
          CHARACTERIZED BY
               viewTablePkg PACKAGE
          BEHAVIOUR
               viewTableBehaviour BEHAVIOUR
          DEFINED AS
          !The table contained in the local database
          which defines local MIB views.  Each SNMP party has a
          single MIB view which is defined by two
          collections of view subtrees: the included view
          subtrees, and the excluded view subtrees.  Every
          such subtree, both included and excluded, is
          defined in this table.

          To determine if a particular object instance is in
          a particular SNMP party's MIB view, compare the
          object instance's Object Identifier with each
          entry (for this party) in this table.  If none
          match, then the object instance is not in the MIB
          view.  If one or more match, then the object
          instance is included in, or excluded from, the MIB
          view according to the value of viewStatus in the
          entry whose value of viewSubtree has the most
          sub-identifiers.  If multiple entries match and
          have the same number of sub-identifiers, then the
          lexicographically greatest instance of viewStatus
          determines the inclusion or exclusion.

          An object instance's Object Identifier X matches
          an entry in this table when the number of sub-
          identifiers in X is at least as many as in the
          value of viewSubtree for the entry, and each sub-
          identifier in the value of viewSubtree matches its
          corresponding sub-identifier in X.  Two sub-
          identifiers match either if the corresponding bit
```

```
          of viewMask is zero (the 'wild card' value), or if
          they are equal.

          Due to this 'wild card' capability, we introduce
          the term, a 'family' of view subtrees, to refer to
          the set of subtrees defined by a particular
          combination of values of viewSubtree and viewMask.
          In the case where no 'wild card' is defined in
          viewMask, the family of view subtrees reduces to a
          single view subtree.!;;
          ATTRIBUTES
               "IIMCIMIBTRANS": internetClassId       GET;;;
REGISTERED AS  { cmipsnmpProxyIMIB 2 1 21 3 1 };
```

3. Attribute Definitions

The templates for the IIMC Proxy SNMP Parties attributes are listed in alphabetical order. The object identifier {cmipsnmpProxyIMIB} is defined in [IIMCIMIBTRANS].

```
aclPrivileges ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:AclPrivileges;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          aclPrivilegesBehaviour BEHAVIOUR
          DEFINED AS
          !The access privileges which govern what
          management operations a particular target party
          may perform when requested by a particular subject
          party.  These privileges are specified as a sum of
          values, where each value specifies a SNMP PDU type
          by which the subject party may request a permitted
          operation.  The value for a particular PDU type is
          computed as 2 raised to the value of the ASN.1
          context-specific tag for the appropriate SNMP PDU
          type.  The values (for the tags defined in RFC
          1157) are defined in RFC 1351 as:

                        Get             :   1
                        GetNext         :   2
                        GetResponse     :   4
                        Set             :   8
                        Trap            :  16

          The null set is represented by the value zero.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 2 1 1 3};

aclStatus ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB:Status;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          aclStatusBehaviour BEHAVIOUR
          DEFINED AS !The status of the access privileges for a
```

```
          particular requesting SNMP party in accessing a
          particular target SNMP party.  Setting an instance
          of this object to the value 'invalid(2)' has the
          effect of invalidating the corresponding access
          privileges.

          It is an implementation-specific matter as to
          whether the agent removes an invalidated entry
          from the table.  Accordingly, management stations
          must be prepared to receive from agents tabular
          information corresponding to entries not currently
          in use.  Proper interpretation of such entries
          requires examination of the relevant aclStatus
          object.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 1 1 1 4 };

aclSubject ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          aclSubjectBehaviour BEHAVIOUR
          DEFINED AS
          !The subject SNMP party whose requests for
          management operations to be performed is
          constrained by this set of access privileges.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 2 1 1 2};

aclTarget ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
IIMCRFC1353ProxyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          aclTargetBehaviour BEHAVIOUR
          DEFINED AS
          !The target SNMP party whose performance of
           management operations is constrained by this set
          of access privileges.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 2 1 1 1};

partyAuthClock ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyASN1.Clock;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyAuthClockBehaviour BEHAVIOUR
          DEFINED AS
          !The authentication clock which represents the
          local notion of the current time specific to the
          party.  This value must not be decremented unless
          the party's secret information is changed
          simultaneously, at which time the party's nonce
          and last-timestamp values must also be reset to
          zero, and the new value of the clock,
          respectively.!;;
```

```
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 6};

partyAuthLifetime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:PartyLifetime;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partyAuthLifetimeBehaviour BEHAVIOUR
          DEFINED AS
          !The lifetime (in units of seconds) which
          represents an administrative upper bound on
          acceptable delivery delay for protocol messages
          generated by the party.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 8};

partyAuthProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCRFC1353ProxyMIB.ObjectIdentifier;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partypartyAuthProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !The authentication protocol by which all messages
          generated by the party are authenticated as to
          origin and integrity.  In this context, the value
          { noAuth } signifies that messages generated by
          the party are not authenticated.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 5};

partyAuthPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:OctetString16;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyAuthPublicBehaviour BEHAVIOUR
          DEFINED AS
          !A publically-readable value for the party.
          Depending on the party's authentication protocol,
          this value may be needed to support the party's
          authentication protocol.  Alternatively, it may be
          used by a manager during the procedure for
          altering secret information about a party.  (For
          example, by altering the value of an instance of
          this object in the same SNMP Set-Request used to
          update an instance of partyAuthPrivate, a
          subsequent Get-Request can determine if the Set-
          Request was successful in the event that no
          response to the Set-Request is received, see RFC1352.)

          The length of the value is dependent on the
          party's authentication protocol.  If not used by
          the authentication protocol, it is recommended
          that agents support values of any length up to and
          including the length of the corresponding
          partyAuthPrivate object.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 7};
```

```
partyIdentity ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCRFC1353ProxyMIB.ObjectIdentifier;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !A party identifier uniquely identifying a
          particular SNMP party.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 1 };

partyMaxMessageSize ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB:PartyMaxMessageSize;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partyMaxMessageSizeBehaviour BEHAVIOUR
          DEFINED AS
          !The maximum length in octets of a SNMP message
          which this party will accept.  For parties which
          execute at an agent, the agent initializes this
          object to the maximum length supported by the
          agent, and does not let the object be set to any
          larger value.  For parties which do not execute at
          the agent, the agent must allow the manager to set
          this object to any legal value, even if it is
          larger than the agent can generate.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 11};

partyPrivProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
IIMCRFC1353ProxyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partyPrivProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !The privacy protocol by which all protocol
          messages received by the party are protected from
          disclosure.  In this context, the value { noPriv }
          signifies that messages received by the party are
          not protected.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 9};

partyPrivPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partyPrivPublicBehaviour BEHAVIOUR
          DEFINED AS
          !A publically-readable value for the party.
          Depending on the party's privacy protocol, this
          value may be needed to support the party's privacy
          protocol.  Alternatively, it may be used by a
```

```
          manager as a part of its procedure for altering
          secret information about a party.  (For example,
          by altering the value of an instance of this
          object in the same SNMP Set-Request used to update
          an instance of partyPrivPrivate, a subsequent
          Get-Request can determine if the Set-Request was
          successful in the event that no response to the
          Set-Request is received, see RFC 1352.)

          The length of the value is dependent on the
          party's privacy protocol.  If not used by the
          privacy protocol, it is recommended that agents
          support values of any length up to and including
          the length of the corresponding partyPrivPrivate
          object.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 10};

partyProxyFor ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
IIMCRFC1353ProxyMIB.ObjectIdentifier;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyProxyForBehaviour BEHAVIOUR
          DEFINED AS
          !The identity of a second SNMP party or other
          management entity with which interaction may be
          necessary to satisfy received management requests.
          In this context, the distinguished value {noProxy}
          signifies that the party responds to received
          management requests by entirely local mechanisms.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 4};

partySecretsAuthPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCRFC1353ProxyMIB:OctetString16;
          MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partySecretsAuthPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !An encoding of the party's private authentication
          key which may be needed to support the
          authentication protocol.  Although the value of
          this variable may be altered by a management
          operation, its value can never be retrieved by a
          management operation: when read, the value of this
          variable is the zero length OCTET STRING.

          The private authentication key is NOT directly
          represented by the value of this variable, but
          rather it is represented according to an encoding.
          This encoding is the bitwise exclusive-OR of the
          old key with the new key, i.e., of the old private
          authentication key (prior to the alteration) with
          the new private authentication key (after the
```

```
          alteration).  Thus, when processing a received
          protocol Set operation, the new private
          authentication key is obtained from the value of
          this variable as the result of a bitwise
          exclusive-OR of the variable's value and the old
          private authentication key.  In calculating the
          exclusive-OR, if the old key is shorter than the
          new key, zero-valued padding is appended to the
          old key.  If no value for the old key exists, a
          zero-length OCTET STRING is used in the
          calculation.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 1 1 1 2 };

partySecretsPrivPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partySecretsPrivPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !An encoding of the party's private encryption key
          which may be needed to support the privacy
          protocol.  Although the value of this variable may
          be altered by a management operation, its value can
          never be retrieved by a management operation: when
          read, the value of this variable is the zero length
          OCTET STRING.

          The private encryption key is NOT directly
          represented by the value of this variable, but
          rather it is represented according to an encoding.
          This encoding is the bitwise exclusive-OR of the
          old key with the new key, i.e., of the old private
          encryption key (prior to the alteration) with the
          new private encryption key (after the alteration).
          Thus, when processing a received protocol Set
          operation, the new private encryption key is
          obtained from the value of this variable as the
          result of a bitwise exclusive-OR of the variable's
          value and the old private encryption key.  In
          calculating the exclusive-OR, if the old key is
          shorter than the new key, zero-valued padding is
          appended to the old key.  If no value for the old
          key exists, a zero-length OCTET STRING is used in
          the calculation.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 1 1 1 3 };

partyStatus ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB:Status;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          partyStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The status of the locally-held information on a
```

```
          particular SNMP party.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 12};

partyTAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB.OctetString;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyTAddressBehaviour BEHAVIOUR
          DEFINED AS
          !The transport service address by which the party
          receives network management traffic, formatted
          according to the corresponding value of
          partyTDomain.  For rfc1351Domain, partyTAddress is
          formatted as a 4-octet IP Address concatenated
          with a 2-octet UDP port number.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 20 2 1 1 3  };

partyTDomain ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCRFC1353ProxyMIB.ObjectIdentifier;
         MATCHES FOR      EQUALITY;
                  BEHAVIOUR
          partyTDomainBehaviour 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 {cmipsnmpProxyIMIB 2 1 20 2 1 1 2  };

viewMask ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
          viewMaskBehaviour BEHAVIOUR
          DEFINED AS
          !The bit mask which, in combination with the
          corresponding instance of viewSubtree, defines a
          family of view subtrees.

          Each bit of this bit mask corresponds to a sub-
          identifier of viewSubtree, with the most
          significant bit of the i-th octet of this octet
          string value (extended if necessary, see below)
          corresponding to the (8*i - 7)-th sub-identifier,
          and the least significant bit of the i-th octet of
          this octet string corresponding to the (8*i)-th
          sub-identifier, where i is in the range 1 through 16.

          Each bit of this bit mask specifies whether or not
          the corresponding sub-identifiers must match when
          determining if an Object Identifier is in this
          family of view subtrees; a '1' indicates that an
```

```
          exact match must occur; a '0' indicates 'wild
          card', i.e., any sub-identifier value matches.

          Thus, the Object Identifier X of an object
          instance is contained in a family of view subtrees
          if the following criteria are met:

          for each sub-identifier of the value of
          viewSubtree, either:

               the i-th bit of viewMask is 0, or

               the i-th sub-identifier of X is equal to
               the i-th sub-identifier of the value of
                              viewSubtree.

          If the value of this bit mask is M bits long and
          there are more than M sub-identifiers in the
          corresponding instance of viewSubtree, then the
          bit mask is extended with 1's to be the required
          length.

          Note that when the value of this object is the
          zero-length string, this extension rule results in
          a mask of all-1's being used (i.e., no 'wild
          card'), and the family of view subtrees is the one
          view subtree uniquely identified by the
          corresponding instance of viewSubtree.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 3 1 1 4};

viewParty ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
                 BEHAVIOUR
          viewPartyBehaviour BEHAVIOUR
          DEFINED AS
          !The SNMP party whose single MIB view includes
          or excludes a particular family of view subtrees.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 3 1 1 1};

viewStatus ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCRFC1353ProxyMIB:ViewStatus;
     MATCHES FOR      EQUALITY, ORDERING;
                 BEHAVIOUR
          viewStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The status of a particular family of view
          subtrees within the particular SNMP party's MIB
          view.  The value 'included(1)' indicates that the
          corresponding instances of viewSubtree and
          viewMask define a family of view subtrees included
          in the MIB view.  The  value 'excluded(2)'
          indicates that the corresponding instances of
```

```
            viewSubtree and viewMask define a family of view
            subtrees excluded from the MIB view.

            Setting an instance of this object to the value
            'invalid(3)' has the effect of invalidating the
            presence or absence of the corresponding family of
            view subtrees in the corresponding SNMP party's
            MIB view.

            It is an implementation-specific matter as to
            whether the agent removes an invalidated entry
            from the table.  Accordingly, management stations
            must be prepared to receive from agents tabular
            information corresponding to entries not currently
            in use.  Proper interpretation of such entries
            requires examination of the relevant viewStatus
            object.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 3 1 1 3};

viewSubtree ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCRFC1353ProxyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
                  BEHAVIOUR
            viewSubtreeBehaviour BEHAVIOUR
            DEFINED AS
            !The view subtree which, in combination with the
            corresponding instance of viewMask, defines a
            family of view subtrees.  This family is included
            a managers MIB view, according to the value of the
            corresponding instance of viewStatus.!;;
REGISTERED AS {cmipsnmpProxyIMIB 2 1 21 3 1 1 2};
```

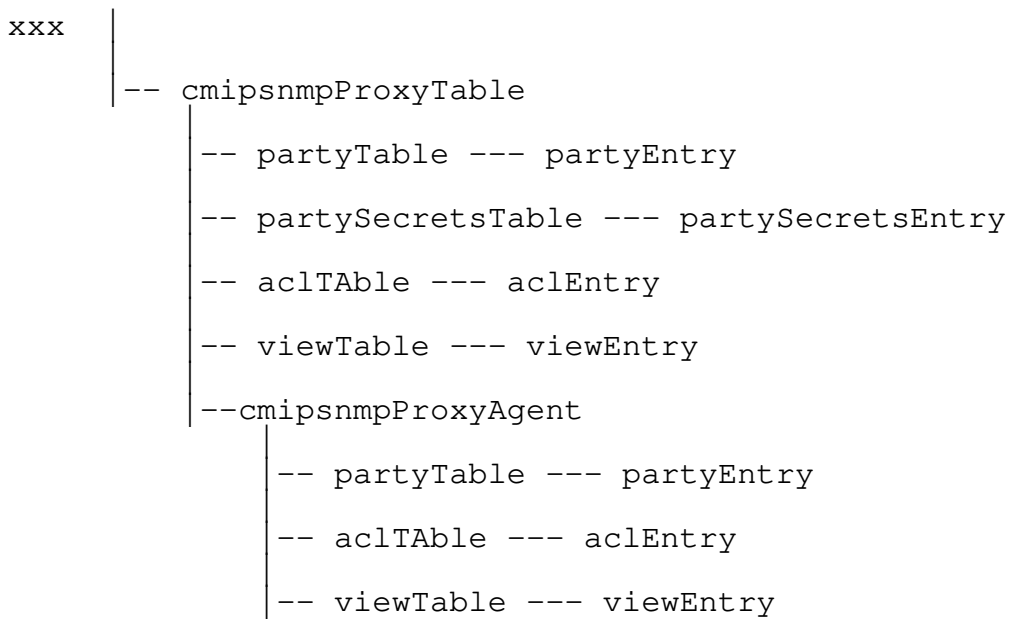
4. Notifications

No traps have been specified in Internet SNMP Party MIB [RFC1353].

5. The Containment Hierarchy

A Naming Tree diagram for IIMC Party MIB managed object classes is illustrated below. Note that the Party MIB appears in two locations in the tree. Placing them as direct subordinates of cmipsnmpProxyTable allows the proxy device to apply global authentication and access control to object types in all Internet agents. It also allows for potential use of this information for manager to proxy communication. Placing them as subordinates of cmipsnmpProxyAgent allows authentication and access control to be applied, either by the proxy device or as pass-through to the Internet agent, to all object types and their instances on a per agent basis. The policy regarding where authentication and access control is to be applied is controlled by variables in the

cmipsnmpProxyTable and cmipsnmpProxyAgent managed objects.



Name Binding templates that define the containment hierarchy for the IIMC Party MIB managed object classes are listed here in alphabetical order. The object identifier {cmipsnmpProxyNB} is defined in [IIMCIMIBTRANS].

```
aclEntry-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     aclEntry
                                            AND SUBCLASSES ;
             NAMED BY SUPERIOR OBJECT CLASS  aclTable
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
             DELETE;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 2 1 1};

aclTable-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     aclTable
                                            AND SUBCLASSES ;
             NAMED BY SUPERIOR OBJECT CLASS
                                       cmipsnmpProxyAgent
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE;
             DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 2 1 };

partySecretsEntry-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS  partySecretsEntry
                                            AND SUBCLASSES;
             NAMED BY SUPERIOR OBJECT CLASS
                                       partySecretsTable
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
```

```
            DELETE;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 1 1 1};

partySecretsTable-NB  NAME BINDING
            SUBORDINATE OBJECT CLASS     partySecretsTable
                                AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
                                 cmipsnmpProxyAgent
                                AND SUBCLASSES;
            WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
            CREATE;
            DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 1 1};

partyEntry-NB  NAME BINDING
            SUBORDINATE OBJECT CLASS  partyEntry
                                          AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS      partyTable
                                          AND SUBCLASSES;
            WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
            CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 20 2 1 1 };

partyTable-NB  NAME BINDING
            SUBORDINATE OBJECT CLASS     partyTable
                                AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
                                 cmipsnmpProxyAgent
                                AND SUBCLASSES;
            WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
            CREATE;
            DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 20 2 1};

viewEntry-NB  NAME BINDING
            SUBORDINATE OBJECT CLASS  viewEntry
                                          AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS      viewTable
                                          AND SUBCLASSES;
            WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
            CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE;
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 3 1 1 };

viewTable-NB  NAME BINDING
            SUBORDINATE OBJECT CLASS     viewTable
                                AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
                                 cmipsnmpProxyAgent
                                AND SUBCLASSES;
            WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
            CREATE;
            DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
```

```
REGISTERED AS { cmipsnmpProxyNB 1 2 1 21 3 1};


aclTable-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     aclTable
                                            AND SUBCLASSES ;
             NAMED BY SUPERIOR OBJECT CLASS
                                       cmipsnmpProxyAgent
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE;
             DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 2 2 1 21 2 1};

partySecretsTable-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     partySecretsTable
                                            AND SUBCLASSES;
             NAMED BY SUPERIOR OBJECT CLASS
                                       cmipsnmpProxyAgent
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE;
             DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 2 2 1 21 1 1};

partyTable-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     partyTable
                                            AND SUBCLASSES;
             NAMED BY SUPERIOR OBJECT CLASS
                                       cmipsnmpProxyAgent
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE;
             DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 2 2 1 20 2 1};

viewTable-NB  NAME BINDING
             SUBORDINATE OBJECT CLASS     viewTable
                                            AND SUBCLASSES;
             NAMED BY SUPERIOR OBJECT CLASS
                                       cmipsnmpProxyAgent
                                            AND SUBCLASSES;
             WITH ATTRIBUTE "IIMCIMIBTRANS": internetClassId;
             CREATE;
             DELETE          ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { cmipsnmpProxyNB 2 2 1 21 3 1};
```

6. ASN.1 Definitions

```
IIMCRFC1353ProxyMIB
DEFINITIONS ::= BEGIN
IMPORTS   Integer, OctetString, ObjectIdentifier
              FROM CmipsnmpCommonDef
```

```
          cmipsnmpProxyIMIB, cmipsnmpProxyNB,
          cmipsnmpProxyNOT
               FROM CmipsnmpProxyAssignedOIDs
          mib, private, internet    FROM RFC1155-SMI;
EXPORTS ; --     Everything


snmpParties         OBJECT IDENTIFIER ::= { mib-2 20 }
partyAdmin          OBJECT IDENTIFIER ::= { snmpParties 1 }
partyPublic         OBJECT IDENTIFIER ::= { snmpParties 2 }

snmpSecrets         OBJECT IDENTIFIER ::= { mib-2 21 }
partyPrivate        OBJECT IDENTIFIER ::= { snmpSecrets 1 }
partyAccess         OBJECT IDENTIFIER ::= { snmpSecrets 2 }
partyViews          OBJECT IDENTIFIER ::= { snmpSecrets 3 }


Clock ::= INTEGER (0..2147483647)

-- A party's authentication clock - a non-negative integer
-- which is incremented as specified/allowed by the party's
-- Authentication Protocol.
--    For noAuth, a party's authentication clock is unused and
-- its value is undefined.
--    For md5AuthProtocol, a party's authentication clock is a
-- relative clock with 1-second granularity.

TAddress ::= OCTET STRING

          -- A textual convention denoting a transport service
          -- address.
          --    For rfc1351Domain, a TAddress is 6 octets long,
          -- the initial 4 octets containing the IP-address in
          -- network-byte order and the last 2 containing the
          -- UDP port in network-byte order.

OctetString16  ::=  OCTET STRING (SIZE (0..16))

PartyAuthLifetime  ::=     INTEGER (0..2147483647)

PartyMaxMessageSize ::= INTEGER (484..65507)

Status ::=     INTEGER  { valid(1), invalid(2) }

ViewStatus ::= INTEGER  {
                          included(1),
                          excluded(2),
                          invalid(3)
                        }

AclPrivileges ::= INTEGER (0..31)

          --- Definitions of Security Protocols
```

```
partyProtocols OBJECT IDENTIFIER ::= { partyAdmin 1 }

noAuth                    -- The protocol without authentication
     OBJECT IDENTIFIER ::= { partyProtocols 1 }

noPriv                    -- The protocol without privacy
     OBJECT IDENTIFIER ::= { partyProtocols 3 }

desPrivProtocol           -- The DES Privacy Protocol
          OBJECT IDENTIFIER ::= { partyProtocols 4 }

md5AuthProtocol          -- The MD5 Authentication Protocol
     OBJECT IDENTIFIER ::= { partyProtocols 5 }


          --- definitions of Transport Domains

transportDomains
     OBJECT IDENTIFIER ::= { partyAdmin 2 }

rfc1351Domain --- RFC-1351 (SNMP over UDP, using SNMP Parties)
     OBJECT IDENTIFIER ::= { transportDomains 1 }

          --- definitions of Proxy Domains

proxyDomains
     OBJECT IDENTIFIER ::= { partyAdmin 3 }

noProxy                   --- Local operation
     OBJECT IDENTIFIER ::= { proxyDomains 1 }


---    Definition of Initial Party Identifiers

--  When devices are installed, they need to be configured
--  with an initial set of SNMP parties.  The configuration
--  of SNMP parties requires (among other things) the
--  assignment of several OBJECT IDENTIFIERs.  Any local
--  network administration can obtain the delegated
--  authority necessary to assign its own OBJECT
--  IDENTIFIERs.  However, to provide for those
--  administrations who have not obtained the necessary
--  authority, this document allocates a branch of the
--  naming tree for use with the following conventions.

initialPartyId OBJECT IDENTIFIER ::= { partyAdmin 4 }

          --  Note these are identified as "initial" party
identifiers
          --  since these allow secure SNMP communication to
proceed,
          --  thereby allowing further SNMP parties to be
configured
          --  through use of the SNMP itself.
```

```
-- Default values

RFC1351Domain  ::=       {rfc1351Domain}
RFC1351DefaultTransport ::= '000000000000'h
NoProxy        ::=       {noProxy}
MD5AuthProtocol ::=      {md5AuthProtocol}
Zero           ::=       0
EmptyString    ::=       ''h
C300           ::=       300
NoPriv         ::=       {noPriv}
C484           ::=       384
Valid          ::=       2
C3             ::=       3
Included       ::=       1

END
```

7. Use of Party MIB

7.1 Initial Values for Proxy/Agent Secure Communications

When Internet agents are installed, they need to be configured with an initial set of SNMP parties such that secure SNMP communications can proceed, and thereby allowing further SNMP parties to be configured through use of the SNMP itself.

[RFC1353] identifies party identifiers, and specify the initial values of various object instances indexed by those identifiers for use with SNMP. In addition, the initial MIB view and access control parameters assigned, by convention, to these parties are identified.

Since the initial party identifiers and associated initial table entries defined in [RFC1353] were predicated on the use of secure SNMP, new party identifiers and table entries need to be defined for use with community strings; they are TBD.

7.2 Authentication and Access Control

Enforcement of authentication and access control, on a per agent basis, may occur either in the proxy or the SNMP agent. Thus, the Party MIb for each agent may exist only in the proxy, or both the proxy and the agent. If it exists in both places, then the Party MIB contents for authentication and access control must be compatible.

7.3 Integrity and Confidentiality

If integrity or confidentiality services are used between the proxy and the SNMP agent, then those Party MIB elements relative to integrity and confidentiality must exist in both the proxy and

the SNMP agent.

8. Acknowledgements

The author thanks the following individuals for their insightful comments and contributions:

Jon Biggar – NETLABS
April Chang – NETLABS
Dean Voiss – NETLABS
Jock Embry – Opening Technologies
Steve Ng – MPR Teltech
Lisa Phfher – Bellcore

References

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

[ISO9595] ISO/IEC IS 9595, Information Technology – Open SystemInterconnection – 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.

[RFC1213] RFC1213, K. McCloghrie and M. Rose – Editors, Management Information Base for Network Management of

TCP/IP-based internets: MIB-II, March 1991.

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

[IIMCIMIBTRANS] L. LaBarre, ISO/CCITT Integrated Management (OIM): Translation of Internet MIBs to ISO/CCITT GDMO MIBs, October, 1992.

[IIMCIMIB-II] L. LaBarre, ISO/CCITT and Internet Management Coexistence: Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB, October 1992.

[IIMCPROXY] A. Chang, ISO/CCITT and Internet Management Coexistence: ISO/CCITT to Internet Management Proxy, October 1992.

[IIMCOMIBTRANS] O. Newnan, ISO/CCITT and Internet Management Coexistence: Translation of ISO/CCITT GDMO MIBs to Internet MIBs, October 1992.

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