ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Security (IIMCSEC)

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 [IIMCIMIBTRANS] [IIMCMIB-II] [IIMCPROXY] and [IIMCOMIBTRANS]. 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.

Editor's Note: Readers are warned that this draft is incomplete as to the security architecture, but fairly complete as to the Party MIB translation.

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 defines the end-to-end security architecture, services, and mechanisms for use with ISO/CCITT-Internet proxies. This document also contains the ISO/CCITT GDMO definition and registration of the SNMP Parties MIB, derived from the Internet SNMP Parties MIB [SNMPv2PARTY] according to the procedures defined in "Translation of Internet MIBs to ISO/CCITT GDMO MIBs" [IIMCIMIBTRANS].

Table of Contents

Status of this Memo i
Abstract ii
Table of Contents ii
Revision History iii
1.Introduction 1
1.1 Background 1
1.2 Overview 2
1.3 Scope 4
1.4 Terms and Conventions 5
2. Security and Management Requirements 5
2.1 Security of Management 5
2.2 Management of Security 5
2.3 Threat Characterization 6
2.3.1 Communications Path Security 6
2.3.2 Managed System Security 7
3. Security Model, Requirements, and Constraints 8
3.1 Security Model 8
3.2 Requirements 9
3.3 Constraints on Mapping Security Services 10
3.4 Consequences of Requirements and Constraints 11
4. Manager to Internet Proxy Security 11
5. Internet Proxy to Internet Agent Security 12
6. Party MIB 12
6.1 Attribute Types 12
6.2 Object Class Definitions 15
6.3 Attribute Definitions 21
6.4 The Containment Hierarchy 36
6.5 ASN.1 Definitions 38
7.MOCS 40
8. Acknowledgments 40
References 41

Revision History

Draft 0 - October 9, 1992

Initial draft of this document (previously entitled "IIMC: Translation of Internet Party MIB (RFC1353) to ISO/CCITT GDMO MIB" [IIMCPARTY]).

Draft 1 - March 26, 1993

Current draft of this document (replaces Draft 0).

Major Changes Since Last Revision

1. Changed title to reflect new scope of document.
2. Added security architecture text.
3. Aligned MIB translation with latest SNMPv2 document [SNMPv2PARTY].
4. Aligned templates with changes as per [IIMCIMIBTRANS].
   - Revised OID translation procedure.
   - Revised generic notification replaces previous notifications.
   - Updated to reflect SNMPv2 changes.
   - Added parsing capability to entry type templates.
   - Revised registration of documents and modules.

Action Item Proposals Contained In This Document

#22 Revamp Party MIB (proposed)

Outstanding Issues

1. Lack of standards and implementation agreements for ISO/CCITT security.
2. Create and delete modifiers in name bindings.

Editor's Note: [All object identifier assignments in this document will be resolved before final publication of this document.]

Editor's Note: [This document will change to reflect the new scope. It is preliminary, and incomplete as to the security architecture, but fairly complete as to the Party MIB translation.]

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:

[IIMCIMIBTRANS]	Translation of Internet MIBs to ISO/CCITT GDMO MIBs
[IIMCOMIBTRANS]	Translation of ISO/CCITT GDMO MIBs to Internet MIBs
[IIMCMIB-II]	Translation of Internet MIB-II to ISO/CCITT GDMO MIB
[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

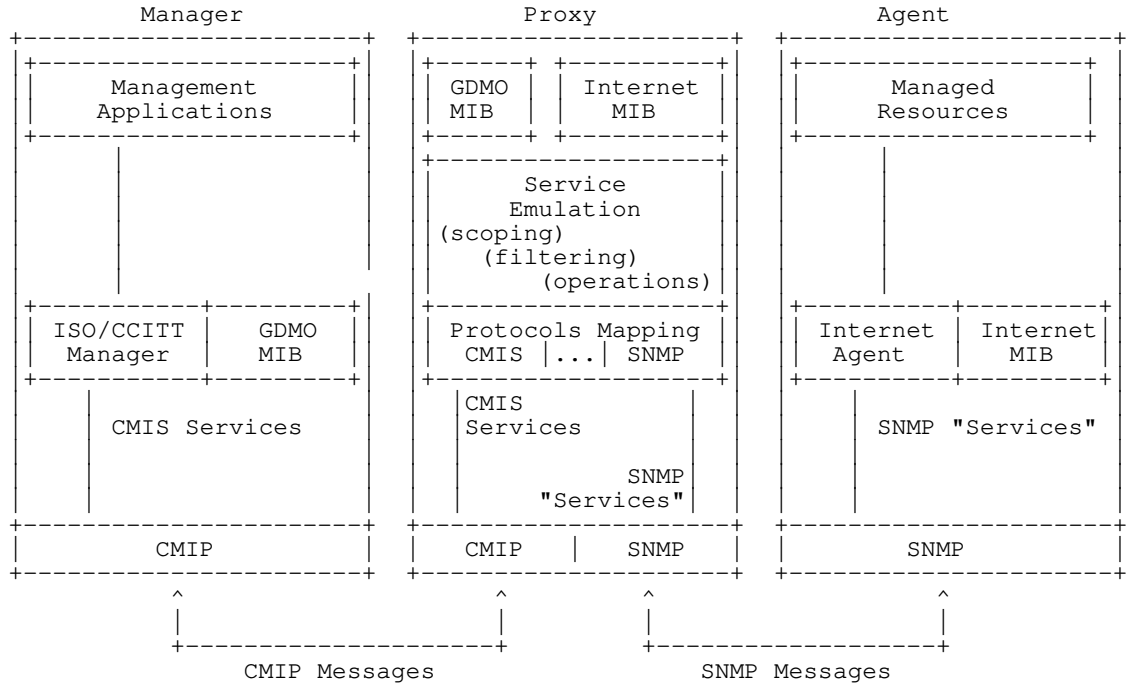
The response to the need for OAM&P of enterprise networks has been the development of network management standards within various networking communities - most notably the ISO/CCITT and Internet communities. However, coordination of standards activities between these two communities has not occurred. 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

One of the IIMC objectives is to provide for the secure end-to-end management of resources managed using ISO/CCITT and Internet management services, protocols and SMI. Security and management by their very nature are entwined such that each needs the services of the other. Security services are required to protect management services. Management services are required to monitor and control security services.

This document defines the security architecture for end-to-end security between an ISO/CCITT manager and an Internet agent via proxies such as that defined in [IIMCPROXY]. The architecture requires that information required to support Internet security mechanisms from an end-to-end perspective, and to manage it, be translated into the ISO/CCITT SMI. This document applies the procedures described in [IIMCMIBTRANS] to the translation and registration of the Internet SNMP Parties MIB defined in [SNMPv2PARTY].

This document assumes that the reader is familiar with the ISO/CCITT and Internet management security services, protocols and mechanisms.

This document 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].

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

```
iimcSEC OBJECT IDENTIFIER ::={iimcManagementDocMan 3}
```

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. Security and Management Requirements

Security and management are entwined by their very nature such that each needs the services of the other. Security services are required to protect management services. Management services are required to monitor and control security services. These requirements are briefly presented in this section.

2.1 Security of Management

Management is most vulnerable to security attacks at the manager user interface, the communications path over which management messages are transmitted, and at the managed system that contains the resources being managed. Accordingly, management's security requirements are to overcome these threats by:

- Preventing unauthorized operator access to manager applications and associated management information contained in a manager workstation,

- Protecting management information in transit between managers and agents, and

- Enforcing management policy regarding access to information within the managed system.

Preventing unauthorized access to manager applications is beyond the scope of this document, and therefore will not be discussed. The characterization of the security threats in relation to the other two vulnerable areas are discussed more fully in the following sections.

2.2 Management of Security

Security requires management support for three basic activities:

- monitoring and control of security mechanisms,
- detection of security related events through security alarm generation, reporting and audit trail analysis,
- damage assessment and recovery from a security attack.

Security mechanisms and algorithm resources are modeled as managed objects and the management information is stored in a secure portion of the management information base. The same management and security mechanisms used to manage non-security managed objects may be applied to the management of

security objects.

2.3 Threat Characterization

Security threats for management are the same as for any distributed application. Security threats can be characterized as being active or passive. Active threats to a management system may effect changes to the state or operation of the managed resource. Examples of active threats are malicious changes to the routing tables of a system, or to the objects used to control decisions related to policies, such as security policies relating to resource access.

Active threats include:

- masquerade,
- modification and fabrication of messages and stored data,
- replay and reordering of messages, and
- denial of management services.

Passive threats are those which, if realized, would not result in any modifications to information contained in the system, e.g., management information, and where neither the operation nor the state of the system is changed.

Passive threats include:

- disclosure of message contents and stored data,
- traffic analysis, and
- repudiation.

2.3.1 Communications Path Security

The threats to the communications path used for manager to agent communications, and applicable security services include:

- modification and fabrication of management messages
  * integrity

- disclosure of management message data
  * confidentiality, selective field confidentiality

- replay and reordering of messages
  * integrity

- denial of management services
  * continuity of operations

- traffic analysis
  * confidentiality

Note that the communications path from the manager to an agent may be direct, or indirect via the management applications of an intermediate manager or proxy. In the indirect case, the portion of the message that must be exposed in the intermediate manager for the purpose of application layer relaying is subject to unauthorized disclosure and modification. Such entities must be trusted not to perform such modifications or to disclose the contents of the management messages. Selective field confidentially services may be required if intermediate managers or proxies are acting as application layer relays in the path. Such selective field services allow only the information in management messages required for application layer routing to be unprotected while preventing other fields in the message from disclosure or modification.

2.3.2 Managed System Security

The threats to the managed system include:

- masquerade of a manager application or operator
  * peer authentication, data origin authentication
- modification and fabrication of data residing in the management information base
  * access control, data integrity
- disclosure of management data in the managed system
  * access control, confidentiality
- repudiation of management requests at the destination
  * non-repudiation at destination.

Non-repudiation services may be provided in circumstances where such accountability is required. While the non-repudiation service does nothing to protect the network, it does provide the capability to trace the entities that are to be blamed for mis-management.

3. Security Model, Requirements, and Constraints

3.1 Security Model

The model for IIMC end-to-end security is illustrated in Figure 2. The objective is to provide continuity of security services from the ISO/CCITT Manager through to the Internet Agent. The end-to-end solution is constrained by the security services available at the Internet agent and those available at the ISO/CCITT Manager. The mapping of security services is provided by the ISO/CCITT-Internet proxy. The mapping of those services at the proxy will depend upon the availability of the services and the compatibility of the mechanisms used to provide the

services.

Figure 2 illustrates the proxy in a separate device from the manager or the agent. If the proxy function is performed in the manager, then how the manager's internal security mechanisms map to Internet security services is beyond the scope of this document. If ISO management services and protocol are provided in the managed device, and the proxy function is still applied, then ISO security services apply at the managed system. The mapping of ISO security services that still apply at the internal proxy to Internet agent interface into equivalent Internet services, e.g., authentication and access control, is beyond the scope of this document.

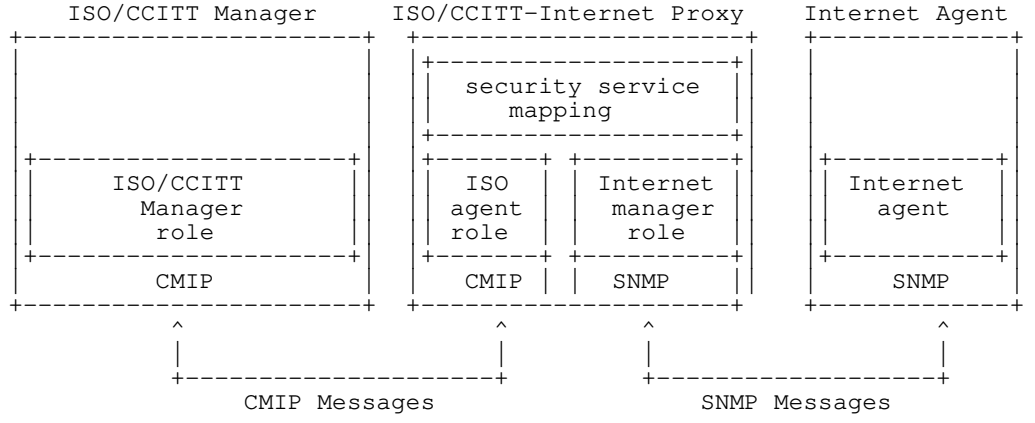


Figure 2: IIMC End-to-end Security Model

The security services are not required to be provided at the same layers in the protocol suites on the two external proxy interfaces. For example, integrity and confidentiality services may be applied at the transport or network layer at the interface to the ISO/CCITT manager, and at the application layer at the interface to the Internet agent.

Depending on the environment, some security services may not be required proxy's interface to the ISO/CCITT manager. For example, data origin authentication and confidentiality services may not be required if the two devices are close together and physical security is adequate to satisfy the security policy.

3.2 Requirements

The basic requirements to be met by the architecture for providing end-to-end security services are support for:

- enforcement of SNMPv1 security services at the agent (community string).

- enforcement of SNMPv2 security services at the agent (party based).

- optional enforcement of access control at the proxy on either SNMPv1 or SNMPv2 agents. Since SNMPv1 does not support access control, this implies that SNMPv2 party based access control shall be enforced at the proxy for both SNMPv1 and SNMPv2 agents.

- optional enforcement of access control at the proxy using OSI access control mechanisms (ISO 10164-9) to the ISO/CCITT managed objects derived from Internet objects for all proxied agents.

- enforcement of access control at the proxy for MIB objects and attributes defined specifically for the proxy operation.

- OSI security services between the ISO/CCITT manager and the proxy.

- mapping of OSI security services into Internet security services, where possible, and forwarding form the ISO/CCITT manager of information required Internet security mechanisms.

3.3 Constraints on Mapping Security Services

The major constraint on mapping security services is that there is no way that all information required for Internet security services can be derived from parameters provided with OSI security services. The security mechanisms are dissimilar enough that mappings do not exist. The result is that the ISO/CCITT manager must be aware of Internet security services used by the proxy, and transfer the information required for those services to operate.

The Internet management SNMPv2 security architecture relies on the identification of distinct entities, called "parties", for peers that exchange SNMP messages [SNMPv2ADMIN]. Multiple parties may exist at the manager and at the agent.

Each distinct SNMPv2 peer is identified by a "party identifier", an OID. Associated with the party identifier

are it's agent address, and parameters for authentication, integrity and confidentiality services to be used when communicating with other parties. Since parties form a peer relationship, these security service parameters for peer parties must be compatible.

The peer relationship between SNMPv2 parties is established via an associated "context", identified by an OID, which provides a means to identify constraints on valid management operations and associated resources (MIB objects). The context also specifies whether the constraints apply to local resources or to remote resources via a proxy relationship.

Therefore, SNMPv2 security requires that the peer parties and their context be identified before an SNMPv2 message will even be accepted by an agent - even if no security services are to be invoked. Only then may data authentication, integrity, confidentiality, and access control services be invoked.

The problem, from a decoupling perspective, is that there is no way that party and context information required for Internet security services can be derived from parameters provided with OSI security services. The same concepts simply do not exist. The result is that the ISO/CCITT manager must be aware of Internet security services used by the proxy, and transfer the party and context information required for those services to operate.

Note, however, that the Internet has registered a set of default parties and contexts that cover a few basic security policies when communicating directly with SNMPv2 agents. These include: no authentication and confidentiality with restricted monitoring privileges; authentication (using MD5) without confidentiality but with full management privileges, and authentication(using MD5) with confidentiality using DES and with full management privileges. If the ISO/CCITT manager specifies to the Internet agent (or proxy) which of these default sets of parties and contexts to use, then the specific parties and contexts need not be known to the manager.

Editor's Note: [We could provide the capability of specifying the default community string, parties and contexts to use when the proxy communicates to agents. This capability could relieve the ISO/CCITT manager from being aware of specific community string or party based security service requirements. The sets could be specified in attributes of the cmipsnmpProxyagent object. In the absence of security parameters being provided by the ISO/CCITT manager, the default parties and context would be in effect. Of course, agents must be configured to support these sets, and the manager would be constrained to work within the

limits of these sets.]

3.4 Consequences of Requirements and Constraints

The consequences of the constraint described in 3.3 are:

- the ISO/CCITT-Internet proxy shall use community string and party/context information provided by the ISO/CCITT manager to determine security services to be invoked relative to an Internet agent.

- if access control mechanisms are used by the proxy on behalf of Internet agents, then the security parameters that would be required by the agent to enforce access control shall be maintained by the proxy. This applies whether Internet or OSI access control mechanisms are used.

4. Manager to Internet Proxy Security

OSI peer authentication services shall be supported in accordance with OMNIPoint 1 security specifications. [NMFSEC]

OSI data origin authentication services shall optionally be supported in accordance with (TBD)

Editor's Note: [To Be Provided.]

Integrity services shall optionally be supported using (TBD).

Editor's Note: [To Be Provided.]

Confidentiality services shall optionally be supported using (TBD).

Editor's Note: [The use of security services for transport (TLSP), network (NLSP), or the generic upper layer security(GULS) [ISO11586-1,2,3,4] to provide these services might be appropriate.]

OSI access control services shall optionally be supported in accordance with [ISO10164-9].

Internet security services shall optionally be supported as follows:

- the following privileged attribute certificate (PAC) shall be used to convey Internet security parameters:

Editor's Note: [Format is TBD. Contents shall include party and context, or community string information.]

5. Internet Proxy to Internet Agent Security

All SNMPv1 and SNMPv2 security services shall be supported.

Editor's Note: [Should we have conformance classes?]

6. Party MIB

The IIMC Party MIB is derived from the Internet Party MIB defined in[SNMPv2PARTY]. Adjustments have been made to the behavior of some elements in the MIB to accommodate SNMPv1 community string based security.

6.1 Attribute Types

```
party ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB:ObjectIdentifier;
     MATCHES FOR      EQUALITY ORDERING;
     BEHAVIOUR
          partyBehaviour BEHAVIOUR
          DEFINED AS
          !Denotes a SNMPv2 party identifier.  Note that
          agents may impose implementation limitations on the
          length of OIDs used to identify Parties. As such,
          management stations creating new parties should be
          aware that using an excessively long OID may result
          in the agent refusing to perform the set operation
          and instead returning the appropriate error
          response, e.g., noCreation.!

tAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB:OctetString;
     MATCHES FOR      EQUALITY ORDERING;
     BEHAVIOUR
          tAddressBehaviour BEHAVIOUR
          DEFINED AS
          !Denotes a transport service address.  For
          snmpUDPDomain, 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.  Consult [5] for
          further information on snmpUDPDomain.!

clock ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:UInteger32;
     BEHAVIOUR
          clockBehaviour BEHAVIOUR
          DEFINED AS
          !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 v2md5AuthProtocol, a party's authentication
          clock is a relative clock with 1-second
          granularity.!


context ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB:ObjectIdentifier;
     MATCHES FOR      EQUALITY ORDERING;
     BEHAVIOUR
          contextBehaviour BEHAVIOUR
          DEFINED AS
          !Denotes a SNMPv2 context identifier. Note that
          agents may impose implementation limitations on the
          length of OIDs used to identify Parties. As such,
          management stations creating new parties should be
          aware that using an excessively long OID may result
          in the agent refusing to perform the set operation
          and instead returning the appropriate error
          response, e.g., noCreation.!


storageType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB:StorageType;
     MATCHES FOR      EQUALITY ORDERING;
     BEHAVIOUR
          storageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !Describes the memory realization of a conceptual
          row.  A row which is volatile(2) is lost upon
          reboot.  A row which is nonVolatile(3) is backed
          up by stable storage.  A row which is permanent(4)
          cannot be changed nor deleted.!
```

6.2 Object Class Definitions

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 [IIMCIMIBTRANS].

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

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

Editor's Note: [The OID fragment "iimcAutoTrans-partyMIB" will be resolved when the iimcAutotrans and partyMIB OID are allocated.]

```
aclEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     aclEntryPkg PACKAGE
     BEHAVIOUR
          aclEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !PARSE
           REFERENCE !!This managed object class maps to
                     aclEntry object in [SNMPv2PARTY].!!;
          MULTIPLEINSTANCES
          INDEX aclSubject, aclTarget, aclResources;
          CREATEDELETEATT aclStatus;
          CREATEDELETEVALUE SNMPV2ROWSTATUS;
          ENDMULTIPLEINSTANCES
          ENDPARSE

          The access privileges for a particular requesting
          SNMP party in accessing a particular target SNMP
          party.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId GET,
               aclTarget                      GET,
               aclSubject                     GET,
               aclResources                   GET,
               aclPrivileges                  GET-REPLACE
                    DEFAULT VALUE IIMCPartyMIB.c-aclPrivileges,
               aclStorageType                 GET-REPLACE
                    DEFAULT VALUE IIMCPartyMIB.c-aclStorageType,
               aclStatus                      GET-REPLACE;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 3 11};

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
               {iimcManagementDocMan 1}:internetClassId GET;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 3 1};

contextEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     contextEntryPkg PACKAGE
     BEHAVIOUR
          contextEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !PARSE
           REFERENCE !!This managed object class maps to
                     contextEntry object in [SNMPv2PARTY].!!;
          MULTIPLEINSTANCES
          INDEX contextIdentity;
          CREATEDELETEATT contextStatus;
          CREATEDELETEVALUE SNMPV2ROWSTATUS;
          ENDMULTIPLEINSTANCES
          ENDPARSE

          Locally held information about a particular
          SNMPv2 context.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}:internetClassId   GET,
               contextIdentity          GET,
               contextIndex             GET-REPLACE,
               contextLocal             GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextLocal,
               contextViewIndex         GET-REPLACE,
               contextLocalEntity       GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextLocalEntity,
               contextLocalTime         GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextLocalTime,
               contextProxyDstParty     GET-REPLACE,
               contextProxySrcParty     GET-REPLACE,
               contextProxyContext      GET-REPLACE,
               contextStorageType       GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextStorageType,
               contextStatus            GET-REPLACE;;;
REGISTERED AS  { partyMIB 2 2 1 1};

contextTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     contextTablePkg PACKAGE
     BEHAVIOUR
```

```
          contextTablePkgBehaviour BEHAVIOUR
     DEFINED AS
     !The SNMPv2 Context database.!;;
     ATTRIBUTES
          {iimcManagementDocMan 1}: internetClassId
GET;;;
REGISTERED AS  { partyMIB 2 2 1};


familyEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
     CHARACTERIZED BY
     familtEntryPkg PACKAGE
     BEHAVIOUR
          familyEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !PARSE
           REFERENCE !!This managed object class maps to
                     familyEntry object in [SNMPv2PARTY].!!;
          MULTIPLEINSTANCES
          INDEX familyIndex;
          CREATEDELETEATT familyStatus;
          CREATEDELETEVALUE SNMPV2ROWSTATUS;
          ENDMULTIPLEINSTANCES
          ENDPARSE

          Information on a particular family of view
          subtrees.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId GET,
               familyIndex              GET,
               familySubtree            GET-REPLACE,
               familyMask               GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-familyMask,
               familyStorageType        GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-familyStorageType,
               familyStatus             GET-REPLACE;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 4 21};

familyTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     familyTablePkg PACKAGE
     BEHAVIOUR
               familyTablePkgBehaviour BEHAVIOUR
          DEFINED AS
          !Locally held information about a family of view
                       subtrees.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId
GET;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 4 2 };
```

```
partyEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
     CHARACTERIZED BY
     partyEntryPkg PACKAGE
     BEHAVIOUR
          partyEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !PARSE
           REFERENCE !!This managed object class maps to
                     partyEntry object in [SNMPv2PARTY].!!;
          MULTIPLEINSTANCES
          INDEX partyIdentity;
          CREATEDELETEATT partyStatus;
          CREATEDELETEVALUE SNMPV2ROWSTATUS;
          ENDMULTIPLEINSTANCES
          ENDPARSE

          Locally held information about a particular
          SNMPv2 party.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId GET,
               partyIdentity            GET-REPLACE,
               partyIndex               GET,
               partyTDomain             GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyTDomain,
               partyTAddress            GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyTAddress,
               partyMaxMessageSize      GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyMaxMessageSize,
               partyLocal               GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyLocal,
               partyAuthProtocol        GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyAuthProtocol,
               partyAuthClock           GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyAuthClock,
               partyAuthPrivate         GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyAuthPrivate,
               partyAuthPublic          GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyAuthPublic,
               partyAuthLifetime        GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyAuthLifetime,
               partyPrivProtocol        GET-REPLACE
                    DEFAULT VALUE
```

```
                    IIMCPartyMIB.c-partyPrivProtocol,
               partyPrivPrivate         GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyPrivPrivate,
               partyPrivPublic          GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyPrivPublic,
               partyCloneFrom           GET-REPLACE,
               partyStorageType         GET-REPLACE
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyStorageType,
               partyStatus              GET-REPLACE;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 1 11};

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 SNMPv2 Party database.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId
GET;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 1 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
          !PARSE
           REFERENCE !!This managed object class maps to
                     viewEntry object in [SNMPv2PARTY].!!;
          MULTIPLEINSTANCES
          INDEX viewIndex, viewSubtree;
          CREATEDELETEATT viewStatus;
          CREATEDELETEVALUE SNMPV2ROWSTATUS;
          ENDMULTIPLEINSTANCES
          ENDPARSE

           Information on a particular family of view
           subtrees included in or excluded from a
           particular SNMPv2 context's MIB view.

           Implementations must not restrict the number of
           families of view subtrees for a given MIB view,
           except as dictated by resource constraints on the
           overall number of entries in the viewTable.!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}: internetClassId GET,
```

```
                viewIndex                GET,
                viewSubtree              GET,
                viewMask                 GET-REPLACE
                     DEFAULT VALUE
                     IIMCPartyMIB.c-viewMask,
                viewType                 GET-REPLACE
                     DEFAULT VALUE
                     IIMCPartyMIB.c-viewType,
                viewStorageType          GET-REPLACE
                     DEFAULT VALUE
                     IIMCPartyMIB.c-viewStorageType,
                viewStatus               GET-REPLACE;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 4 11};


viewTable  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     viewTablePkg PACKAGE
     BEHAVIOUR
               viewTableBehaviour BEHAVIOUR
          DEFINED AS
          !Locally held information about the MIB views
           known to this SNMPv2 entity.

           Each SNMPv2 context which is locally accessible
           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 MIB view, compare the object
           instance's OBJECT IDENTIFIER with each of the MIB
           view's entries 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 viewType 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 viewType
           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
             {iimcManagementDocMan 1}: internetClassId
GET;;;
REGISTERED AS  { iimcAutoTrans-partyMIB 2 4 1 };
```

6.3 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 IIMCPartyMIB:AclPrivileges;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclPrivilegesBehaviour BEHAVIOUR
          DEFINED AS
          !The access privileges which govern what
           management operations a particular target party
           may perform with respect to a particular SNMPv2
           context when requested by a particular subject
           party.  These privileges are specified as a sum of
           values, where each value specifies a SNMPv2 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
           SNMPv2 PDU type.  The values (for the tags defined
           in [5]) are defined in [3] as:

                       Get         :   1
                       GetNext     :   2
                       Response    :   4
                       Set         :   8
                       unused      :  16
                       GetBulk     :  32
                       Inform      :  64
                       SNMPv2-Trap : 128

           The null set is represented by the value zero.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 4};

aclResources ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
```

```
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclResourcesBehaviour BEHAVIOUR
          DEFINED AS
          !The value of an instance of this object
           identifies a SNMPv2 context in an access control
           policy, and has the same value as the instance of
           the contextIndex object for that SNMPv2 context.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 3};

aclStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          aclStatusBehaviour BEHAVIOUR
          DEFINED AS !The status of this conceptual row in the
                      aclTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 6};

aclStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          aclStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The storage type for this conceptual row in the
           aclTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 5};

aclSubject ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclSubjectBehaviour BEHAVIOUR
          DEFINED AS
          !The value of an instance of this object
           identifies a SNMPv2 party which is the subject of
           an access control policy, and has the same value
           as the instance of the partyIndex object for that
           SNMPv2 party.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 2};

aclTarget ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclTargetBehaviour BEHAVIOUR
          DEFINED AS
          !The value of an instance of this object
           identifies a SNMPv2 party which is the target of
           an access control policy, and has the same value
           as the instance of the partyIndex object for that
           party.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 3 1 1 1};

contextIdentity ATTRIBUTE
```

```
     DERIVED FROM context;
     BEHAVIOUR
          contextIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !A context identifier uniquely identifying a
           particular SNMPv2 context.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 1};

contextIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextIndexBehaviour BEHAVIOUR
          DEFINED AS
          !A unique value for each SNMPv2 context. The
           value for each SNMPv2 context must remain constant
           at least from one re-initialization of the
           entity's network management system to the next
           re-initialization.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 2};

contextLocal ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:truthValue;
     BEHAVIOUR
          contextLocalBehaviour BEHAVIOUR
          DEFINED AS
          !An indication of whether this context is realized
           by this SNMPv2 entity.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 3};

contextViewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextViewIndexBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of an instance of this object is
           zero, then this corresponding conceptual row in
           the contextTable refers to a SNMPv2 context which
           identifies a proxy relationship; the values of the
           corresponding instances of the
           contextProxyDstParty, contextProxySrcParty, and
           contextProxyContext objects provide further
           information on the proxy relationship.

           Otherwise, if the value of an instance of this
           object is greater than zero, then this
           corresponding conceptual row in the contextTable
           refers to a SNMPv2 context which identifies a MIB
           view of a locally accessible entity; the value of
           the instance identifies the particular MIB view
           which has the same value of viewIndex; and the
           value of the corresponding instances of the
           contextLocalEntity and contextLocalTime objects
```

```
          provide further information on the local entity
          and its temporal domain.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 4};

contextLocalEntity ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalEntityBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of the corresponding instance of the
           contextViewIndex is greater than zero, then the
           value of an instance of this object identifies the
           local entity whose management information is in
           the SNMPv2 context's MIB view.  The empty string
           indicates that the MIB view contains the SNMPv2
           entity's own local management information;
           otherwise, a non-empty string indicates that the
           MIB view contains management information of some
           other local entity, e.g.,'Repeater1'.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 5};

contextLocalTime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalTimeBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of the corresponding instance of the
           contextViewIndex is greater than zero, then the
           value of an instance of this object identifies the
           temporal context of the management information in
           the MIB view.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 6};

contextProxyDstParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          contextProxyDstPartyBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of the corresponding instance of the
           contextViewIndex is equal to zero, then the value
           of an instance of this object identifies a SNMPv2
           party which is the proxy destination of a proxy
           relationship.

           If the value of the corresponding instance of the
           contextViewIndex is greater than zero, then the
           value of an instance of this object is zero.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 7};

contextProxySrcParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
```

```
          contextProxySrcPartyBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of the corresponding instance of the
           contextViewIndex is equal to zero, then the value
           of an instance of this object identifies a SNMPv2
           party which is the proxy source of a proxy
           relationship.

           Interpretation of an instance of this object
           depends upon the value of the transport domain
           associated with the SNMPv2 party used as the proxy
           destination in this proxy relationship.

           If the value of the corresponding instance of the
           contextViewIndex is greater than zero, then the
           value of an instance of this object is zero.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 8};

contextProxyContext ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextProxyContextBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of the corresponding instance of the
           contextViewIndex is equal to zero, then the value
           of an instance of this object identifies the
           context of a proxy relationship.

           Interpretation of an instance of this object
           depends upon the value of the transport domain
           associated with the SNMPv2 party used as the proxy
           destination in this proxy relationship.

           If the value of the corresponding instance of the
           contextViewIndex is greater than zero, then the
           value of an instance of this object is { 0 0 }.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 9};


contextStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          contextStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The storage type for this conceptual row in the
           contextTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 10};

contextStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          contextStatusBehaviour BEHAVIOUR
          DEFINED AS
```

```
          !The status of this conceptual row in the
           contextTable.

           A context is not qualified for activation until
           instances of all corresponding columns have the
           appropriate value.  In  particular, if the
           context's contextViewIndex is greater than zero,
           then the viewStatus column of the associated
           conceptual row(s) in the viewTable must have the
           value `active'.  Until instances of all
           corresponding columns are appropriately
           configured, the value of the corresponding
           instance of the contextStatus column is
           `notReady'.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1 1 11};

familyIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          familyIndexBehaviour BEHAVIOUR
          DEFINED AS
          !A unique value for each family of view subtrees.
           The value for each family of view subtrees must
           remain constant at least from one re-
           initialization of the entity's network management
           system to the next re-initialization.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 2 1 1};

familySubtree ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          familySubtreeBehaviour BEHAVIOUR
          DEFINED AS
          !An object identifier which, in combination with
           the corresponding instance of familyMask, defines a
           family of view subtrees.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 2 1 2};

familyMask ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCPartyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          familyMaskBehaviour BEHAVIOUR
          DEFINED AS
          !The bit mask which, in combination with the
           corresponding instance of familySubtree, defines a
           family of view subtrees.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 2 1 3};

familyStorageType ATTRIBUTE
     DERIVED FROM storageType;
```

```
     BEHAVIOUR
          familyStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The storage type for this conceptual row in the
           familyTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 2 1 4};

familyStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          familyStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The status of this conceptual row in the
           familyTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 2 1 5};

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 {iimcAutoTrans-partyMIB 2 1 1 1 8};

partyAuthLifetime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB: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 {iimcAutoTrans-partyMIB 2 1 1 1 11};

partyAuthPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, SUBSTRINGS;
     BEHAVIOUR
          partypartyAuthPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !If the value of partyAuthProtocol is
          {snmpv1CommString} then this attribute contains the
          community string to be used with SNMPv1 security.
```

```
            If the value of partyAuthProtocol is not
            {snmpv1CommString} then this attribute contains 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 (e.g., a SNMPv2 Set-Request), 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 {iimcAutoTrans-partyMIB 2 1 1 1 9};

partyAuthProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCPartyMIB.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.

          The value {snmpv1CommString} indicates that SNMPv1
          community string is to be used. The community string
          shall be present in partyAuthPrivate!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 7};

partyAuthPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB:OctetString16;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
```

```
          partyAuthPublicBehaviour BEHAVIOUR
          DEFINED AS
          !A publicly-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 {iimcAutoTrans-partyMIB 2 1 1 1 10};

partyCloneFrom ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          partyCloneFromBehaviour BEHAVIOUR
          DEFINED AS
          !The identity of a party to clone authentication
           and privacy parameters from.  When read, the value
           { 0 0 } is returned.

           This value can only be written when the associated
           instance of partyStatus either does not exist or
           has the value `notReady'.  When written, the value
           identifies a party, the cloning party, whose
           status column has the value `active'.  The cloning
           party is used in two ways.

           One, if instances of the following objects do not
           exist for the party being created, then they are
           created with values identical to those of the
           corresponding objects for the cloning party:

               partyAuthProtocol
          partyAuthPublic
     partyAuthLifetime
     partyPrivProtocol
     partyPrivPublic

           Two, instances of the following objects are
           updated using the corresponding values of the
           cloning party:
```

```
                partyAuthPrivate
                partyPrivPrivate

            (e.g., the value of the cloning party's instance
            of the partyAuthPrivate object is XOR'd with the
            value of the partyAuthPrivate instances of the
            party being created.)!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 15};

partyIdentity ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          partyIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !A party identifier uniquely identifying a
          particular SNMP party.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 1};

partyIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyIndexBehaviour BEHAVIOUR
          DEFINED AS
          !A unique value for each SNMPv2 party.  The value
           for each SNMPv2 party must remain constant at
           least from one re-initialization of the entity's
           network management system to the next re-
           initialization.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 2};

partyLocal ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:truthValue;
     BEHAVIOUR
          partyLocalBehaviour BEHAVIOUR
          DEFINED AS
          !An indication of whether this party executes at
           this SNMPv2 entity.  If this object has a value of
           true(1), then the SNMPv2 entity will listen for
           SNMPv2 messages on the partyTAddress associated
           with this party.  If this object has the value
           false(2), then the SNMPv2 entity will not listen
           for SNMPv2 messages on the partyTAddress
           associated with this party.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 6};

partyMaxMessageSize ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCPartyMIB: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 {iimcAutoTrans-partyMIB 2 1 1 1 5};

partyPrivProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.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 {iimcAutoTrans-partyMIB 2 1 1 1 12};

partyPrivPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPrivateBehaviour 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 (e.g., a
           SNMPv2 Set-Request), 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 {iimcAutoTrans-partyMIB 2 1 1 1 13};


partyPrivPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB:OctetString16;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPublicBehaviour BEHAVIOUR
          DEFINED AS
          !A publicly-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 {iimcAutoTrans-partyMIB 2 1 1 1 14};

partyStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          partyStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The status of this conceptual row in the
           partyTable.

           A party is not qualified for activation until
           instances of all columns of its partyEntry row
           have an appropriate value.  In particular:

           A value must be written to the Party's
           partyCloneFrom object.

           If the Party's partyAuthProtocol object has the
           value md5AuthProtocol,
           then the corresponding instance of
           partyAuthPrivate must contain a secret of the
           appropriate length.  Further, at least one
           management protocol set operation updating the
           value of the party's partyAuthPrivate object
           must be successfully processed, before the
```

```
          partyAuthPrivate column is considered
          appropriately configured.

          If the Party's partyPrivProtocol object has the
          value desPrivProtocol,
          then the corresponding instance of
          partyPrivPrivate must contain a secret of the
          appropriate length.  Further, at least one
          management protocol set operation updating the
          value of the party's partyPrivPrivate object
          must be successfully processed, before the
          partyPrivPrivate column is considered
          appropriately configured.

          Until instances of all corresponding columns are
          appropriately configured, the value of the
          corresponding instance of the partyStatus column is
          `notReady'.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 17};

partyStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          partyStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The storage type for this conceptual row in the
           partyTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1 16};

partyTAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, SUBSTRINGS;
     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 {iimcAutoTrans-partyMIB 2 1 1 1 4};

partyTDomain ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.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 {iimcAutoTrans-partyMIB 2 1 1 1 3};



viewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Integer64k;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewIndexBehaviour BEHAVIOUR
          DEFINED AS
          !A unique value for each MIB view.  The value for
           each MIB view must remain constant at least from
           one re-initialization of the entity's network
           management system to the next re-initialization.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1 1};

viewMask ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCPartyMIB: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 {iimcAutoTrans-partyMIB 2 4 1 1 3};

viewStatus ATTRIBUTE
     DERIVED FROM (iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          viewStatusBehaviour BEHAVIOUR
          DEFINED AS
          !The status of this conceptual row in the
           viewTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1 6};

viewStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          viewStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The storage type for this conceptual row in the
           viewTable.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1 5};

viewSubtree ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB:ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewSubtreeBehaviour BEHAVIOUR
          DEFINED AS
          !A MIB subtree.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1 2};

viewType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB:ViewType;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewTypeBehaviour BEHAVIOUR
          DEFINED AS
          !The status of a particular family of view
           subtrees within the particular SNMPv2 context'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 MIBview.!;;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1 4};
```

6.4 The Containment Hierarchy

A Naming Tree diagram for IIMC Party MIB managed object classes is illustrated below. The IIMC Party MIB is subordinate to the ISO/CCITT system managed object that represents the Internet agent or proxy.

```
"Rec. X.721 | ISO/IEC 10165-2 : 1992" : system
     |
     |
     |-- partyTable --- partyEntry
     |
     |-- contextTable --- contextEntry
     |
     |-- aclTAble --- aclEntry
     |
     |-- viewTable --- viewEntry
```

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 {iimcAutotrans} is assigned in [IIMCIMIBTRANS].

Editor's Note: [The OID fragment "iimcAutoTrans-partyMIB" will be resolved when the iimcAutotrans and partyMIB OID are allocated.]

```
aclEntry-aclTableNB  NAME BINDING
              SUBORDINATE OBJECT CLASS     aclEntry
                                             AND SUBCLASSES ;
              NAMED BY SUPERIOR OBJECT CLASS  aclTable
                                             AND SUBCLASSES;
              WITH ATTRIBUTE
                    {iimcManagementDocMan 1}: internetClassId;
              CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
              DELETE;
REGISTERED AS { iimcAutoTrans-partyMIB 2 3 1 1};

aclTable-systemNB  NAME BINDING
              SUBORDINATE OBJECT CLASS     aclTable
                                             AND SUBCLASSES ;
              NAMED BY SUPERIOR OBJECT CLASS
               "Rec. X.721 | ISO/IEC 10165-2 : 1992" :
system
                                    AND SUBCLASSES;
              WITH ATTRIBUTE
                    {iimcManagementDocMan 1}: internetClassId;
              CREATE     WITH-AUTOMATIC-INSTANCE-NAMING;
```

```
            DELETE     ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { iimcAutoTrans-partyMIB 2 3 1};



contextEntry-contextTableNB  NAME BINDING
            SUBORDINATE OBJECT CLASS  contextEntry
                                          AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
                                     contextTable
                                          AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE;
REGISTERED AS { iimcAutoTrans-partyMIB 2 2 1 1};

contextTable-systemNB  NAME BINDING
            SUBORDINATE OBJECT CLASS  contextTable
                                 AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
         "Rec. X.721 | ISO/IEC 10165-2 : 1992" :system
                                 AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE     WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE     ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoTrans-partyMIB 2 2 1};

partyEntry-partyTableNB  NAME BINDING
            SUBORDINATE OBJECT CLASS  partyEntry
                                          AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS      partyTable
                                          AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1 1};

partyTable-systemNB  NAME BINDING
            SUBORDINATE OBJECT CLASS partyTable
                                 AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
         "Rec. X.721 | ISO/IEC 10165-2 : 1992" :system
                                 AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE     WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE     ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoTrans-partyMIB 2 1 1};


viewEntry-viewTableNB  NAME BINDING
```

```
            SUBORDINATE OBJECT CLASS  viewEntry
                                          AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS      viewTable
                                          AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE          WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1 1};

viewTable-systemNB  NAME BINDING
            SUBORDINATE OBJECT CLASS viewTable
                                AND SUBCLASSES;
            NAMED BY SUPERIOR OBJECT CLASS
         "Rec. X.721 | ISO/IEC 10165-2 : 1992" :system
                                AND SUBCLASSES;
            WITH ATTRIBUTE
                  {iimcManagementDocMan 1}: internetClassId;
            CREATE     WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE     ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iimcAutoTrans-partyMIB 2 4 1};
```

6.5 ASN.1 Definitions

```
IIMCPartyMIB {iimcManagementModMan 3}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS   Integer, OctetString, ObjectIdentifier
               FROM IimcCommonDef
          iimcAutoTrans, iimcManagementDoc
               FROM IimcAssignedOIDs
          mib-2, private, internet
               FROM RFC1155-SMI;

iimcSEC OBJECT IDENTIFIER ::= {iimcManagementDocMan 3}

partyMIB OBJECT IDENTIFIER ::= {  TBD }


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 v2md5AuthProtocol, 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 snmpUDPDomain, 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.

Integer64k     ::=       INTEGER (1..65535)

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

PartyAuthLifetime  ::=     INTEGER (0..2147483647)

PartyMaxMessageSize ::= INTEGER (484..65507)

StorageType ::= INTEGER {
                        other(1),       -- eh?
                        volatile(2),    -- e.g., in RAM
                        nonVolatile(3), -- e.g., in NVRAM
                        permanent(4)    -- e.g., in ROM
                        }

ViewType ::= INTEGER  {
                         included(1),
                         excluded(2)
                      }

AclPrivileges ::= INTEGER (0..31)

-- assigned OIDs

snmpv2 OBJECT IDENTIFIER ::= {  TBD  }
snmpUDPDomain  OBJECT IDENTIFIER  ::= {snmpv2 1 1 1}
partyAdmin     OBJECT IDENTIFIER ::= { partyMIB 1 }
partyProtocols OBJECT IDENTIFIER ::= { partyAdmin 1 }
noAuth         OBJECT IDENTIFIER ::= { partyProtocols 1 }
noPriv         OBJECT IDENTIFIER ::= { partyProtocols 2 }
desPrivProtocol OBJECT IDENTIFIER ::= { partyProtocols 3 }
v2md5AuthProtocol OBJECT IDENTIFIER ::= { partyProtocols 4 }
temporalDomains OBJECT IDENTIFIER ::= { partyAdmin 2 }
currentTime    OBJECT IDENTIFIER ::= { temporalDomains 1 }
restartTime    OBJECT IDENTIFIER ::= { temporalDomains 2 }
cacheTime      OBJECT IDENTIFIER ::= { temporalDomains 3 }
initialPartyId OBJECT IDENTIFIER ::= { partyAdmin 3 }
initialContextId OBJECT IDENTIFIER ::= { partyAdmin 4 }

-- Default value constants

c-aclPrivileges          INTEGER ::= 35
c-aclStorageType         INTEGER ::= 3
c-contextLocal           BOOLEAN ::= TRUE
c-contextLocalEntity     OCTET STRING ::= ''h
c-contextLocalTime       OBJECT IDENTIFIER ::= {currentTime}
c-contextStorageType     INTEGER ::= 3
c-familyMask             OCTET STRING ::= ''h
c-familyStorageType      INTEGER ::= 3
c-partyTDomain           snmpUDPDomain
```

```
c-partyTAddress          OCTET STRING  ::= '000000000000'h
c-partyMaxMessageSize    INTEGER ::= 484
c-partyLocal             BOOLEAN ::= FALSE
c-partyAuthProtocol      OBJECT IDENTIFIER ::=
{v2md5AuthProtocol}
c-partyAuthClock         INTEGER ::= 0
c-partyAuthPrivate       OCTET STRING ::= ''h
c-partyAuthPublic        OCTET STRING ::= ''h
c-partyAuthLifetime      INTEGER ::= 300
c-partyPrivProtocol      OBJECT IDENTIFIER ::= {noPriv}
c-partyPrivPrivate       OCTET STRING ::= ''h
c-partyPrivPublic        OCTET STRING ::= ''h
c-partyStorageType       INTEGER ::= 3
c-viewMask               OCTET STRING ::= ''h
c-viewType               INTEGER ::= 1
c-viewStorageType        INTEGER ::= 3


END
```

7.MOCS

Editor's Note: [To Be Provided.]

8. 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.

[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.

[ISO10164-9] ISO DIS 10165-9, Information Processing Systems -Open Systems Interconnection - Structure of Management Information - Part 9: Objects and Attributes for Access Control, 1993

[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.

[ISO11586-1] ISO/IEC CD11586-1, Information Technology - Generic Upper Layers Security - Part 1: Overview, Models and Notation, November 1992.

[ISO11586-2] ISO/IEC CD11586-2, Information Technology - Generic Upper Layers Security - Part 2: Security Exchange Service Element(SESE) Service Definition, November 1992.

[ISO11586-3] ISO/IEC CD11586-3, Information Technology - Generic Upper Layers Security - Part 3: Security Exchange Service Element(SESE) Protocol Specification, November 1992.

[ISO11586-4] ISO/IEC CD11586-4, Information Technology - Generic Upper Layers Security - Part 4: Protecting Transfer Syntax Specification, November 1992.

[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-basedinternets: MIB-II, March 1991.

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

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

[NMFSEC] Network Management Forum: Forum 016, Application Services: Security of Management, Issue 1.0, August 1992.