ISO/CCITT and Internet Management Coexistence (IIMC):

ISO/CCITT to Internet Management Security

(IIMCSEC)

Draft 2
May 26, 1993

Lee LaBarre (Editor)

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

Status of this Memo

This document provides information to the network and systems management community. This document is intended as a contribution to on going 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.

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 [RFC1447] 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..... iv
1.1 Problem Statement..... 1
1.2 Overview of IIMC..... 1
1.3 MIB Translation Procedures..... 2
1.4 Native Management Model..... 3
1.5 Proxy Management Model..... 4
1.6 Scope of this Document..... 5
1.7 Terms and Conventions..... 5
2. Security and Management Considerations..... 5
2.1 General Considerations..... 6
2.1.1 Security of Management..... 6
2.1.2 Management of Security..... 6
2.1.3 Threat Characterization..... 6
2.1.3.1 Communications Path Security..... 7
2.1.3.2 Managed System Security..... 8
2.2 ISO/CCITT to Internet Security Environment..... 8
2.2.1 Security Model..... 8
2.2.2 Supported Security Capabilities..... 9
2.2.3 Internet Management Security..... 10
2.2.3.1 SNMPv1 Security..... 10
2.2.3.2 SNMPv2 Security..... 10
2.2.4 Constraints on Mapping Security Services..... 11
2.2.5 ISO Access Control Framework and SNMP Security..... 12
3. Security Specifications..... 13
3.1 ISO Manager to ISO/CCITT-Internet Proxy Security..... 13
3.2 ISO/CCITT-Internet Proxy to Internet Agent Security.. 15
3.3 ISO/CCITT-Internet Proxy Access Control Enforcement.. 15
4. IIMC Party MIB..... 15
4.1 Party MIB GDMO Templates..... 16
-- 4.1.1 Party MIB Managed Object Classes..... 16
-- 4.1.2 Party MIB Attributes..... 22
-- 4.1.3 Party MIB Name Bindings..... 39
-- 4.1.4 Party MIB Attribute Types..... 42

4.2 Party MIB ASN.1 Modules 44
5. Acknowledgments 45
References 47
Appendix A (Normative)
Managed Object Conformance Statements (MOCS) 50

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
Previous draft of this document (replaced Draft 0).

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

Major Changes Since Last Revision

1. Aligned templates with changes as per [IIMCIMIBTRANS].
   - Added scannable BEHAVIOUR to all templates.
   - ASN.1 modules now use IMPORTS from RFCs.
   - Optimized ASN.1 module to reduce constructs.
   - Put GDMO templates and ASN.1 modules together to facilitate machine parsing of text.
2. Aligned Party MIB translation with [RFC1447].
   - Registration corrected
   - Family views removed
3. Modified and expanded clause 2.2:
   - Modified 2.2.2 per Vancouver decisions.
   - Added 2.2.3 in Internet Management security
   - Modified 2.2.4 Constraints on Mapping Security Services to include an expanded definition of the problem and constraints on its solution.
   - Added 2.2.5 ISO Access Control Framework and SNMP Security. This presents the basis for specifying the proxy interface to the manager, using the characterization of the SNMP access control as a capability based scheme.
4. Modified and expanded clause 3 Security Specifications, to specify the security interface requirements more explicitly in accordance with decisions made in Vancouver, and in accordance with the capability characterization of SNMP access control.

Outstanding Issues

Review for approach correctness and completeness is requested. In addition, reviewers are asked to comment on the following question.

1. Should a security mechanism be specified for use in protecting passwords, e.g. MD5 or SHA?

1. Introduction

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

1.1 Problem Statement

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

- The ISO/CCITT community developed the Common Management Information Protocol (CMIP) [ISO9596-1], and related SMI documents [ISO10165-1,2,4].

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

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

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

1.2 Overview of IIMC

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

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

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

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

This overall strategy was outlined in a joint publication developed by the NM Forum and X/Open entitled "ISO/CCITT and Internet Management: Coexistence and Interworking Strategy" [NMFTR107]. The documents included in the IIMC package are the next level of detailed specifications which implement several of the methodologies identified in the strategy.

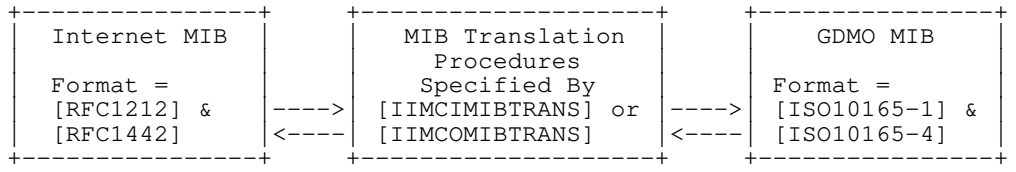
1.3 MIB Translation Procedures

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

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

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

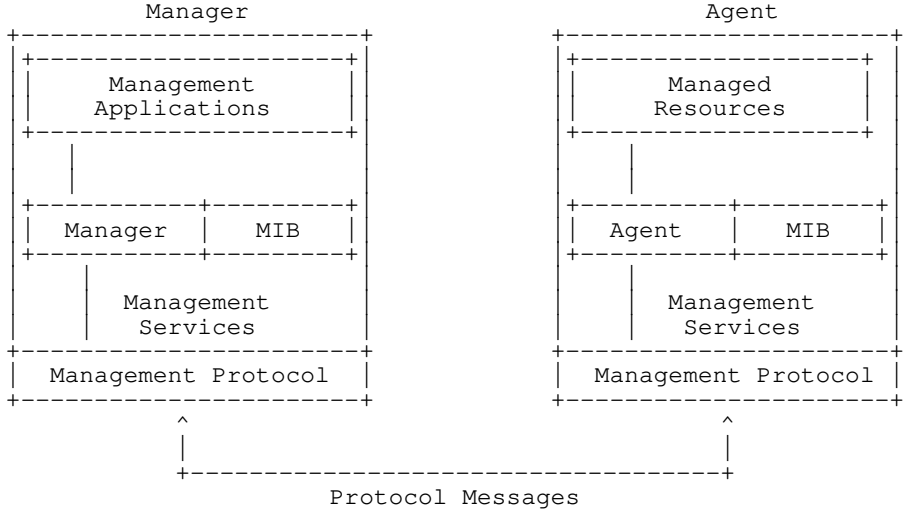
The IIMC approach is to specify direct translation procedures which yield a pair of functionally-equivalent MIBs, as shown in the following figure.



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

1.4 Native Management Model

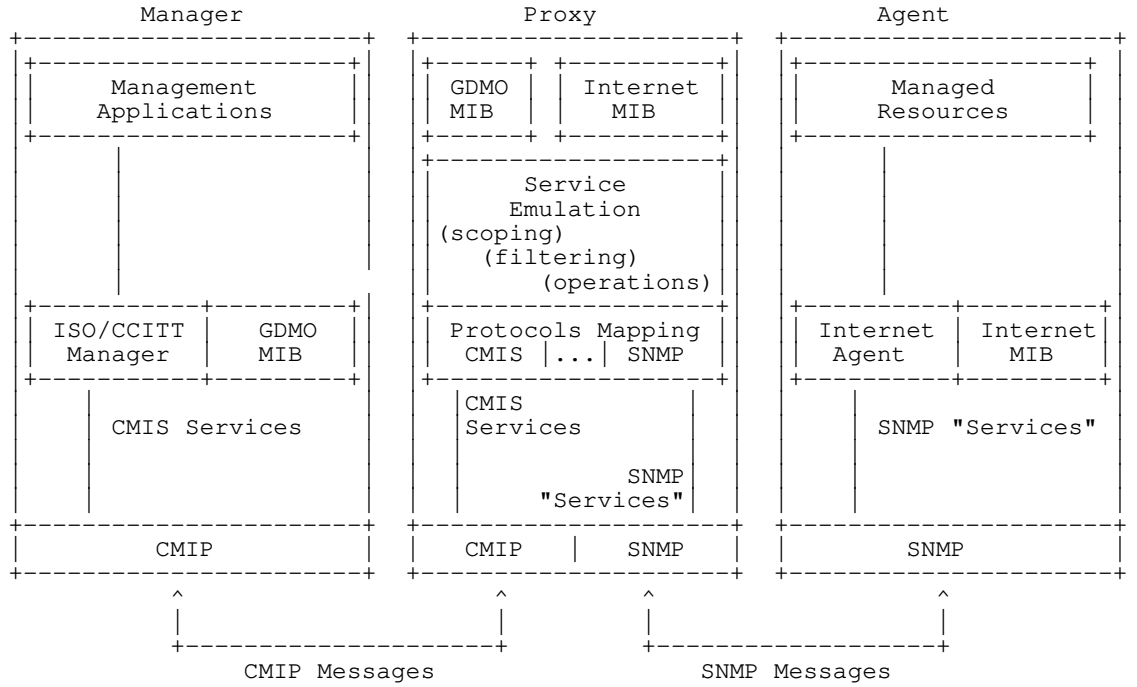
The basic model for ISO/CCITT and Internet management is illustrated in the following diagram.



Within IIMC documents, this model is referred to as the "native" management model. MIBs translated using IIMC procedures can be used by "native" agent implementations. For example, an ISO/CCITT agent can make visible TCP/IP managed resources using the translated GDMO version of the Internet MIB-II [RFC1213] specified by [IIMCMIB-II]. Dual-stack managers or agents may also be implemented which support both the original MIB and the translated MIB generated using IIMC-specified procedures.

1.5 Proxy Management Model

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



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

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

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

1.6 Scope of this Document

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 (IIMCSEC) 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 [IIMCIMIBTRANS] to the translation and registration of the Internet SNMP Parties MIB defined in [RFC1447].

1.7 Terms and Conventions

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

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 to SMI translation procedures defined in [IIMCIMIBTRANS].

Other terms and conventions used throughout this document are defined in Section 2.

2. Security and Management Considerations

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

2.1 General Considerations

2.1.1 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 considerations 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.

2.1.2 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, and
- 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.

2.1.3 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.

2.1.3.1 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 needed 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 needed for application layer routing to be unprotected while preventing other fields in the message from disclosure or modification.

2.1.3.2 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 destination
  * non-repudiation at destination.

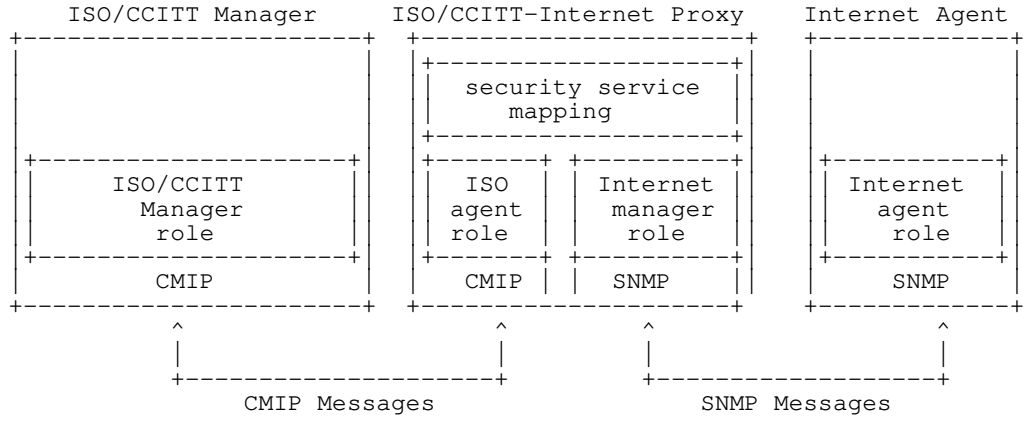
Non-repudiation services may be provided in circumstances where such accountability is needed. While the non-repudiation service does nothing to protect the network, it does provide the capability to trace the entities that are to be blamed for mis-management.

2.2 ISO/CCITT to Internet Security Environment

2.2.1 Security Model

The model for IIMC end-to-end security is illustrated in the following figure. 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.

This figure 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 (native CMIP agent), then ISO security services apply at the managed system. The mapping of any ISO security services that may still possibly 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.



- ISO peer authentication
- ISO data origin authentication*
- ISO integrity, confidentiality*
- Internet access control
- ISO access control+

- Internet data origin authentication#
- Internet integrity, confidentiality#
- Internet access control#

* OSI application layer standards are in progress. These services maybe provided by lower layers in some environments, e.g., transport and network

SNMPv1 and SNMPv2 have different mechanisms (or none)

+ ISO access control may be applied by the proxy to GDMO objects, if enforcement is at the proxy.

IIMC End-to-end Security Model

The security services do not have 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 needed at a proxy's interface to the ISO/CCITT manager. For example, data origin authentication and confidentiality services may not be needed if the two devices are close together and physical security is adequate to satisfy the security policy.

2.2.2 Supported Security Capabilities

The basic security capabilities 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/context based),

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

objects for all proxied agents, and for the MIB specific to the proxy [IIMCPROXY],

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

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

2.2.3 Internet Management Security

The security services for Internet management differs depending on the version of SNMP (SNMPv1 or SNMPv2) used.

2.2.3.1 SNMPv1 Security

The SNMPv1 security relies on the transfer of an unprotected community string that represents the capabilities that the initiator has with respect to operations on a set of objects.

2.2.3.2 SNMPv2 Security

The SNMPv2 security architecture relies on the identification of distinct, globally unique, entities, called "parties", for peers that exchange SNMP messages [RFC1445]. 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 access control, authentication, integrity and confidentiality services which may 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 access to associated resources (MIB objects). The context also specifies whether the constraints apply to local resources or to remote resources via a (yet another) proxy relationship.

The minimal SNMPv2 security service allowed is access control as specified by the source (srcParty), destination party (dstParty), and context identifiers. SNMPv2 requests that do not contain all three identifiers are discarded.

As discussed in 2.2.5, the access control scheme used by SNMP security can be considered a form of capability scheme.

2.2.4 Constraints on Mapping Security Services

The mapping of security services end-to-end is primarily constrained by the security services available at the Internet agent. The possible application level security services at Internet agents is well defined for SNMPv1, and for SNMPv2 (currently proposed draft standard). However, it cannot be assumed that all Internet agents will implement the full range of their defined security services, or that they are all required for any given environment.

Given the known potential availability of Internet security services at Internet agents, and at the Internet proxy, three major problems arise in proxy situations:

a) Selection from the security services available at the Internet proxy to Internet agent interface of those services that are appropriate to the threats at that interface, according to the established security policy.

b) Providing security services at the ISO manager to Internet proxy interface that are appropriate to the threats at that interface, according to the established security policy.

c) Transfer to the Internet proxy from the ISO manager of security parameters required for Internet proxy to Internet agent security.

Note: An exact mapping of security services between both Internet proxy interfaces may not be required. The environments at the two interfaces may be completely different, e.g., the manager and proxy may be in the same room while the agents are geographically distributed.

Assume the following environment and constraints.

i) The ISO/CCITT-Internet proxy is geographically remote from both the ISO/CCITT manager and the Internet agents, and the threats at both interfaces are the same.

ii) Only application level security services are used.

iii) The Internet agents and Internet proxy support the full range of security services defined for them. They include, for the respective SNMP versions:

Service	**SNMPv1**	**SNMPv2**
Peer Authentication	-	X
Data Origin Authentication	-	X
Access Control	x	X
Connectionless Integrity	-	X
Connectionless Confidentiality	-	X
Replay, Reorder Protection	-	X

The first problem (a) can be solved by configuring the security parameters of the Internet agents and Internet proxy, either through local or remote management mechanisms.

The second problem (b) can be solved by implementing the appropriate OSI management services in the ISO/CCITT manager and ISO/CCITT-Internet proxy, and configuring the mechanisms to provide the service. This problem is complicated by the current lack of Stable OSI security standards for data origin authentication, integrity, confidentiality, and access control.

The third problem (c) can be solved by using an access control certificate to transfer the Internet security parameters. This problem is complicated by the current lack of a stable standard for access control certificates. Given the necessity for such transfers in proxy situations, an preliminary ACC will have to be used. However, an attempt should be made to align as closely as possible with proposed ISO standards.

2.2.5 ISO Access Control Framework and SNMP Security

The SNMP access control schemes can be most nearly categorized as capability schemes using the definitions in the ISO Access Control Framework [ISO10181-3]. A capability defines a set of allowed operations that the initiator of the operation is allowed to perform on an identified set of targets.

The SNMPv1 scheme is a very weak capability scheme which uses the community string to identify which operations are permitted or not permitted on a set of objects. However the community string is not bound to the initiator, i.e., it may possibly be changed in transit. Proof of its authenticity can be inferred from the fact that the SNMPv1 agent is configured to have knowledge of the capabilities represented by the community string. In that sense, the person that configured the community string, either via local mechanisms, or via remote management mechanisms can be considered to provide the third party proof of authenticity.

The SNMPv2 scheme is a stronger capability scheme which uses the combination of SNMPv2 srcParty, dstParty, and context identifiers to identify which operations are permitted or not-permitted on a set of objects. The srcParty binds it to the initiator of the request. Proof of its authenticity can be inferred from the fact that the SNMPv2 agent is configured to have knowledge of the capabilities represented by the interrelated parameters. In that sense, the person that configured the Internet Party MIB, either via local mechanisms, or via remote management mechanisms can be considered to provide the third party proof of authenticity.

Capabilities may be carried in access control tokens or access control certificates (ACC). Tokens and certificates are similar. A token differs from a certificate in that it is not created by an entity which is a security authority, and it does not necessarily contain an indication of the time period for which it is valid. The sealing of the ACC by a security authority and the provision of a time period for which it is valid provides third party proof of its authenticity.

3. Security Specifications

3.1 ISO Manager to ISO/CCITT-Internet Proxy Security

OSI peer authentication services shall be supported in accordance with OMNIPoint 1 security specifications in [NMF016] Annex B. At minimum, the simple authentication, protected password, shall be supported as specified in [NMF016] Annex B.2.

Editor's Note: [Should a protection algorithm be specified, e.g. MD5 or SHA?]

OSI data origin authentication services, integrity services, confidentiality services, and access control services are currently beyond the scope of this document due to the lack of stable relevant ISO security standards. Specifications for these services will be defined in a future Issue when the relevant base standards become International Standards.

Support shall be provided for the transfer of Internet security service parameters from the ISO/CCITT manager to an ISO/CCITT-Internet proxy at the time of management association establishment, and with CMIP requests. Access control security parameters shall be transferred as security attributes of an Access Control Certificate (ACC), also referred to as a Privilege Attribute Certificate (PAC). Implementations shall support the ACC defined in OMNIPoint 1 [NMF016] Annex K, and the associated PICS in Annex E.8.8.

To allow for later inclusion of security attributes for OSI access control, the ACC transferred to an ISO/CCITT-Internet proxy shall be a compound ACC, with the first ACC in the sequence of contained ACCs being the ACC with the SNMP security attributes. The containing ACC of the compound ACC transferred to an ISO/CCITT-Internet proxy shall contain the privilegeAttributes sequence, in accordance with [NMF016] Annex E.8.8. The privilegeAttributes sequence may be empty.

The Internet security parameters shall be transferred using the security attributes defined by [ECMA138]. The security attributes are defined in the Security-Services-Attributes module in [ECMA138] using the attribute macro defined for the ISO Directory Services [ISO9594]. For ease of use, these definitions are repeated below; in the event of transcription

error, the original source specifications are normative.

The security parameters to be transferred in the ACC shall include the Security-capability-type-1 attribute as defined in [ECMA138].

The Security-capability-type-1 attribute grants access of the type accessType to the object(s) defined by objectDefiner.

```
Security-capability-type-1 ::= ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX CapabilityType1
     SINGLE VALUE

CapabilityType1 ::= SEQUENCE {
     objectDefiner  ObjectDefiner,
     accessType     AccessDefiner}

ObjectDefiner ::= IntegerOrString
AccessDefiner ::= IntegerOrString

IntegerOrString ::= CHOICE {
     integerPart    INTEGER,
     stringPart     IA5String}
```

The IntegerOrString choice shall be Ia5String.

The Security-capability-type-1 attribute is registered as:

```
desd-att-capability-type-1 OBJECT IDENTIFIER ::=
                    {iso(1) identified-organization(3)
                    icd-ecma(0012) standard(0)
                    desd(138)
                    attributeIdentifiers(3) 4}
```

The Security-capability-type-1 attribute shall be used for proxy to SNMPv1 agents to carry the community string indicating the accessType. The objectDefiner shall be undefined and represented as an empty string (''H).

The Security-capability-type-1 attribute shall be used for proxy to SNMPv2 agents to carry the srcParty and dstParty as the objectDefiners, and the context identifier as the accessType.

The contents of the Security-capability-type-1 attribute shall be as described below.

[ECMA138] Security Attribute	SNMPv2 Security Parameter	SNMPv1 Security Parameter
Security-capability -type-1		

objectDefiner accessType	src/dst Parties context	""H community string

The src/dst Parties are the srcParty and dstParty SNMPv2 parameters separated by a slash (/). The SNMPv2 parameters shall be ASN.1 object identifiers represented using the "dot" notation convention, where the OID sub-identifiers are represented in decimal character form and separated by decimal points. For example, {1 3 6 1 6 3 3 2 3 1 1 3} is represented as "1.3.6.1.6.3.3.2.3.1.1.3".

3.2 ISO/CCITT-Internet Proxy to Internet Agent Security

An ISO/CCITT-Internet proxy that supports SNMPv1 shall support the community string security services as defined in [RFC1157].

An ISO/CCITT-Internet proxy that supports SNMPv2 shall support the security services as defined in [RFC1447], with minimal support for the "Party No Privacy" compliance level specified by clause 3.7.1 of [RFC1447].

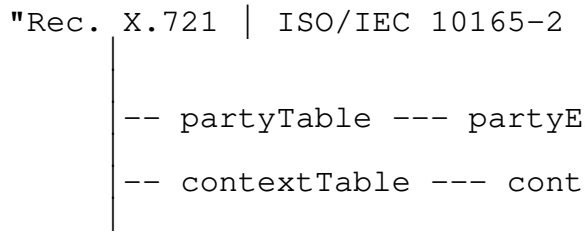
3.3 ISO/CCITT-Internet Proxy Access Control Enforcement

Access control enforcement at the ISO/CCITT-Internet proxy is currently beyond the scope of this document. However, access control enforcement at the proxy will be based on OSI access control for management as defined in [ISO10164-9] when it becomes an International Standard.

4. IIMC Party MIB

Support for the SNMPv2 security services requires that the Internet Party MIB [RFC1447] be instantiated in the ISO/CCITT-Internet proxy and SNMPv2 agents. The IIMC Party MIB is derived from the Internet Party MIB defined in [RFC1447]. Adjustments have been made to the behavior of some elements in the MIB to accommodate SNMPv1 community string based security.

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.



```
     |-- aclTable --- aclEntry
     |
     |-- viewTable --- viewEntry
```

The IIMC Party MIB elements that are specific to the proxy, i.e., local, should be instantiated under the ISO system object that is specific to the proxy. The IIMC MIB elements that are specific to each SNMP agent are actually instantiated in the SNMP agent. Duplicates of MIB elements instantiated in the SNMP agent should not be instantiated in the proxy if a stateless approach to proxy is used as described in [IIMCPROXY].

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

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

```
     iimcIIMCSEC OBJECT IDENTIFIER ::={iimcManagementDocMan 4}
```

4.1 Party MIB GDMO Templates

-- 4.1.1 Party MIB Managed Object Classes

```
aclEntry  MANAGED OBJECT CLASS
     DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
     aclEntryPkg PACKAGE
     BEHAVIOUR
          aclEntryPkgBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    aclEntry object in [RFC1447].!!;
          DESCRIPTION
          !!The access privileges for a particular requesting
          SNMP party in accessing a particular target SNMP
          party.!!;
          INDEX aclSubject, aclTarget, aclResources;
          ENDPARSE!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}:internetClassId GET,
               aclTarget                        GET,
               aclSubject                       GET,
               aclResources                     GET,
               aclPrivileges
                    DEFAULT VALUE IIMCPartyMIB.c-aclPrivileges
                    GET-REPLACE,
               aclStorageType
                    DEFAULT VALUE
```

```
                    IIMCPartyMIB.c-DEFAULTStorageType
                    GET-REPLACE,
               aclStatus                    GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 2 3 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    aclTable object in [RFC1447].!!;
          DESCRIPTION !!The access privileges database.!!;
          CONCEPTUALTABLE
          ENDPARSE!;;
          ATTRIBUTES
          {iimcManagementDocMan 1}:internetClassId GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    contextEntry object in [RFC1447].!!;
          DESCRIPTION
          !!Locally held information about a particular
          SNMPv2 context.!!;
          INDEX contextIdentity;
          ENDPARSE!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}:internetClassId GET,
               contextIdentity          GET,
               contextIndex             GET-REPLACE,
               contextLocal
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextLocal
                    GET-REPLACE,
               contextViewIndex         GET-REPLACE,
               contextLocalEntity
                    DEFAULT VALUE
                    IIMCPartyMIB.c-DEFAULTNullString
                    GET-REPLACE,
               contextLocalTime
                    DEFAULT VALUE
                    IIMCPartyMIB.c-contextLocalTime
                    GET-REPLACE,
```

```
               contextProxyDstParty     GET-REPLACE,
               contextProxySrcParty     GET-REPLACE,
               contextProxyContext      GET-REPLACE,
               contextStorageType
                    DEFAULT VALUE
                    IIMCPartyMIB.c-DEFAULTStorageType
                    GET-REPLACE,
               contextStatus            GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    contextTable object in [RFC1447].!!;
          DESCRIPTION !!The SNMPv2 Context database.!!;
          CONCEPTUALTABLE
          ENDPARSE!;;
          ATTRIBUTES
          {iimcManagementDocMan 1}:internetClassId GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 2 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    partyEntry object in [RFC1447].!!;
          DESCRIPTION
          !!Locally held information about a particular
          SNMPv2 party.!!;
          INDEX partyIdentity;
          ENDPARSE!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}:internetClassId GET,
               partyIdentity            GET,
               partyIndex               GET,
               partyTDomain
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyTDomain
                    GET-REPLACE,
               partyTAddress
                    DEFAULT VALUE
                    IIMCPartyMIB.c-partyTAddress
                    GET-REPLACE,
```

```
                partyMaxMessageSize
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyMaxMessageSize
                     GET-REPLACE,
                partyLocal
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyLocal
                     GET-REPLACE,
                partyAuthProtocol
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyAuthProtocol
                     GET-REPLACE,
                partyAuthClock
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyAuthClock
                     GET-REPLACE,
                partyAuthPrivate
                     DEFAULT VALUE
                     IIMCPartyMIB.c-DEFAULTNullString
                     GET-REPLACE,
                partyAuthPublic
                     DEFAULT VALUE
                     IIMCPartyMIB.c-DEFAULTNullString
                     GET-REPLACE,
                partyAuthLifetime
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyAuthLifetime
                     GET-REPLACE,
                partyPrivProtocol
                     DEFAULT VALUE
                     IIMCPartyMIB.c-partyPrivProtocol
                     GET-REPLACE,
                partyPrivPrivate
                     DEFAULT VALUE
                     IIMCPartyMIB.c-DEFAULTNullString
                     GET-REPLACE,
                partyPrivPublic
                     DEFAULT VALUE
                     IIMCPartyMIB.c-DEFAULTNullString
                     GET-REPLACE,
                partyCloneFrom          GET-REPLACE,
                partyStorageType
                DEFAULT VALUE
                     IIMCPartyMIB.c-DEFAULTStorageType
                     GET-REPLACE,
                partyStatus             GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 2 1 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    partyTable object in [RFC1447].!!;
          DESCRIPTION !!The SNMPv2 Party database.!!;
          CONCEPTUALTABLE
          ENDPARSE!;;
          ATTRIBUTES
          {iimcManagementDocMan 1}:internetClassId GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    viewEntry object in [RFC1447].!!;
          DESCRIPTION
          !!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.!!;
          INDEX viewIndex, viewSubtree;
          ENDPARSE!;;
          ATTRIBUTES
               {iimcManagementDocMan 1}:internetClassId GET,
               viewIndex                GET,
               viewSubtree              GET,
               viewMask
                    DEFAULT VALUE
                    IIMCPartyMIB.c-DEFAULTNullString
                    GET-REPLACE,
               viewType
                    DEFAULT VALUE
                    IIMCPartyMIB.c-viewType
                    GET-REPLACE,
               viewStorageType
                    DEFAULT VALUE
                    IIMCPartyMIB.c-DEFAULTStorageType
                    GET-REPLACE,
               viewStatus               GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 2 4 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
          !BEGINPARSE
          REFERENCE !!This managed object class maps to
                    partyTable object in [RFC1447].!!;
          DESCRIPTION !!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.!!;
          CONCEPTUALTABLE
          ENDPARSE!;;
          ATTRIBUTES
          {iimcManagementDocMan 1}:internetClassId GET;;;
REGISTERED AS  {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 };
```

-- 4.1.2 Party MIB Attributes

```
aclPrivileges ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.AclPrivileges;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclPrivilegesBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 [RFC1448]) are defined in [RFC1445] 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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 4};

aclResources ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclResourcesBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 3};

aclStatus ATTRIBUTE
```

```
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          aclStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The status of this conceptual row in the
             aclTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 6};

aclStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          aclStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The storage type for this conceptual row in the
           aclTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 5};

aclSubject ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclSubjectBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 2};

aclTarget ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          aclTargetBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
```

```
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 3 1 1 1};

contextIdentity ATTRIBUTE
     DERIVED FROM context;
     BEHAVIOUR
          contextIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!A context identifier uniquely identifying a
           particular SNMPv2 context.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 1};

contextIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 2};

contextLocal ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:truthValue;
     BEHAVIOUR
          contextLocalBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!An indication of whether this context is realized
           by this SNMPv2 entity.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 3};
```

```
contextViewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextViewIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 4};

contextLocalEntity ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalEntityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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'.!!;
```

```
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 5};

contextLocalTime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextLocalTimeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 6};

contextProxyDstParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          contextProxyDstPartyBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 7};

contextProxySrcParty ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          contextProxySrcPartyBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 8};

contextProxyContext ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextProxyContextBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 }.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 9};

contextStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          contextStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The storage type for this conceptual row in the
           contextTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 10};
```

```
contextStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          contextStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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'.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 2 1 1 11};


partyAuthClock ATTRIBUTE
     DERIVED FROM clock;
     BEHAVIOUR
          partyAuthClockBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 8};

partyAuthLifetime ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.PartyAuthLifetime;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyAuthLifetimeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
```

```
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The lifetime (in units of seconds) which
          represents an administrative upper bound on
          acceptable delivery delay for protocol messages
          generated by the party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 11};

partyAuthPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, SUBSTRINGS;
     BEHAVIOUR
          partypartyAuthPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
          defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 9};
```

```
partyAuthProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partypartyAuthProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 7};


partyAuthPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyAuthPublicBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 RFC1446.)

          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.!!;
          ENDPARSE!;;
```

```
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 10};

partyCloneFrom ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          partyCloneFromBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.)!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 15};

partyIdentity ATTRIBUTE
     DERIVED FROM party;
     BEHAVIOUR
          partyIdentityBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
```

```
          defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!A party identifier uniquely identifying a
          particular SNMP party.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 1};

partyIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 2};

partyLocal ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:truthValue;
     BEHAVIOUR
          partyLocalBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 6};

partyMaxMessageSize ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCPartyMIB.PartyMaxMessageSize;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyMaxMessageSizeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
```

```
          defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 5};

partyPrivProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivProtocolBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 12};

partyPrivPrivate ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPrivateBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 13};

partyPrivPublic ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyPrivPublicBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 1446.)

          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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 14};

partyStatus ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          partyStatusBehaviour BEHAVIOUR
          DEFINED AS
```

```
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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'.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 17};

partyStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          partyStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The storage type for this conceptual row in the
           partyTable.!!;
```

```
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 16};

partyTAddress ATTRIBUTE
     DERIVED FROM tAddress;
     BEHAVIOUR
          partyTAddressBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 4};

partyTDomain ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY;
     BEHAVIOUR
          partyTDomainBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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).!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 1 1 1 3};

viewIndex ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.Index;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewIndexBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
```

```
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 1};

viewMask ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
                    IIMCPartyMIB.ViewMask;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewMaskBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 3};

viewStatus ATTRIBUTE
     DERIVED FROM (iimcManagementDocMan 1}:rowStatus;
     BEHAVIOUR
          viewStatusBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The status of this conceptual row in the
           viewTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 6};

viewStorageType ATTRIBUTE
     DERIVED FROM storageType;
     BEHAVIOUR
          viewStorageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!The storage type for this conceptual row in the
           viewTable.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 5};

viewSubtree ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewSubtreeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!A MIB subtree.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 2};

viewType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX IIMCPartyMIB.ViewType;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          viewTypeBehaviour BEHAVIOUR
          DEFINED AS
```

```
          !BEGINPARSE
          REFERENCE !!This corresponds to the object type
           defined in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 MIB view.!!;
          ENDPARSE!;;
REGISTERED AS {iimcAutoTrans 1 3 6 1 6 3 3 2 4 1 1 4};
```

-- 4.1.3 Party MIB Name Bindings

```
aclEntry-aclTableNB  NAME BINDING
          SUBORDINATE OBJECT CLASS      aclEntry
                                             AND SUBCLASSES ;
          NAMED BY SUPERIOR OBJECT CLASS     aclTable
                                             AND SUBCLASSES;
              WITH ATTRIBUTE
                    {iimcManagementDocMan 1}: internetClassId;
          BEHAVIOUR
          aclEntry-aclTableNBBehaviour  BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX aclSubject, aclTarget, aclResources;
               CREATEDELETEATT aclStatus;
               CREATEDELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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;
          BEHAVIOUR
          contextEntry-contextTableNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX contextIdentity;
               CREATEDELETEATT contextStatus;
               CREATEDELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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;
          BEHAVIOUR
          partyEntry-partyTableNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX partyIdentity;
               CREATEDELETEATT partyStatus;
               CREATEDELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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;
          viewEntry-viewTableNBBehaviour BEHAVIOUR
               DEFINED AS
               !BEGINPARSE
               INDEX viewIndex, viewSubtree;
               CREATEDELETEATT viewStatus;
               CREATEDELETEVALUE SNMPV2ROWSTATUS;
               ENDPARSE!;;
          CREATE         WITH-AUTOMATIC-INSTANCE-NAMING,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 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,
                         WITH-REFERENCE-OBJECT;
          DELETE         DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementNB 1 3 6 1 6 3 3 2 4 1};
```

-- 4.1.4 Party MIB Attribute Types

```
party ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          partyBehaviour BEHAVIOUR
```

```
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;;

tAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.OctetString;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          tAddressBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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 [RFC1448]
          for further information on snmpUDPDomain.!!;
          ENDPARSE!;;;

clock ATTRIBUTE
     DERIVED FROM {iimcManagementDocMan 1}:clock;
     BEHAVIOUR
          clockBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;;
```

```
context ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.ObjectIdentifier;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          contextBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;;


storageType ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
               IIMCPartyMIB.StorageType;
     MATCHES FOR      EQUALITY, ORDERING;
     BEHAVIOUR
          storageTypeBehaviour BEHAVIOUR
          DEFINED AS
          !BEGINPARSE
          REFERENCE !!This corresponds to the type defined
                    in [RFC1447] by the same name.!!;
          DESCRIPTION
          !!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.!!;
          ENDPARSE!;;;
```

4.2 Party MIB ASN.1 Modules

```
IIMCPartyMIB {iimcManagementModMan 3}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
          UInteger32, snmpv2, MODULE-IDENTITY
               FROM SNMPv2-SMI
          partyAdmin, partyProtocols, noAuth,
          desPrivProtocol, v2md5AuthProtocol,
          temporalDomains, currentTime,
          restartTime, cacheTime, initialPartyId,
          initialContextId
```

```
            FROM SNMPv2-Party-MIB
        snmpUDPDomain
            FROM SNMPv2-TM
        iimcManagementDocMan, iimcManagementModMan,
        iimcAutoTrans, iimcManagementNB,
            FROM IimcAssignedOIDs {iimcManagementModMan 1};

--  The following registration identifier is assigned to this
--  document using procedures defined in [IIMCIMIBTRANS]:

iimcIIMCSEC OBJECT IDENTIFIER ::={iimcManagementDocMan 4}

-- Generic syntax

Integer   ::=  INTEGER

OctetString    ::=  OCTET STRING

ObjectIdentifier    ::=  OBJECT IDENTIFIER


-- MIB specific syntax

AclPrivileges ::= INTEGER (0..255)

Clock     ::= UInteger32

Index     ::= INTEGER (1..65535)

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
                        }

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

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


-- Default value constants

c-aclPrivileges          INTEGER ::= 35
c-contextLocal           BOOLEAN ::= TRUE
c-DEFAULTNullString      OCTET STRING ::= ''H
c-contextLocalTime       Clock ::= currentTime
```

```
c-DEFAULTStorageType     INTEGER ::= 3
c-partyTDomain           OBJECT IDENTIFIER ::= 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-partyAuthLifetime      INTEGER ::= 300
c-partyPrivProtocol      OBJECT IDENTIFIER ::= noPriv
c-viewType               INTEGER ::= 1

END
```

5. Acknowledgments

The following individuals have contributed to this effort.

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

Dean Voiss - NetLabs
David Waitzman - BBN
Graham Wisdom - Timeplex
Yoshi Yamashita - NKK Corporation

References

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

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

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

[ISO9594] ISO/IEC 9594, Information Technology - Open System Interconnection - The Directory - Parts 1-8.

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

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

[ISO10164-9] ISO DIS 10164-9, Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 9: Objects and Attributes for Access Control, ISO/IEC JTC1/SC21/N7661, March, 1993.

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

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

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

[ISO10181-3] ISO/IEC DIS 10181-3, Information Technology , OSI Security Model, Part 3: Access Control Framework, 1993.

[ISO11586-1] ISO/IEC CD 11586-1, Information Technology - Generic Upper Layers Security - Part 1: Overview, Models and Notation, November 1992.

[ISO11586-2] ISO/IEC CD 11586-2, Information Technology - Generic Upper Layers Security - Part 2: Security Exchange Service Element(SESE) Service Definition, November 1992.

[ISO11586-3] ISO/IEC CD 11586-3, Information Technology -Generic

Upper Layers Security - Part 3: Security Exchange Service Element(SESE) Protocol Specification, November 1992.

[ISO11586-4] ISO/IEC CD 11586-4, 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] RFC1157, J.D. Case, M.S. Fedor, M.L. Schoffstall,C. Davin, Simple Network Management Protocol (SNMP), May 1990.

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

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

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

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

[RFC1445] J.R. Davin, J.M. Galvin, K.McCloghrie, Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

[RFC1446] J.M. Galvin, K. McCloghrie, J.R. Davin, Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

[RFC1447] J.D. Case, K. McCloghrie, M.T. Rose, S.L. Waldbusser, Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2), April 1993.

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

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

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

[IIMCMIB-II] ISO/CCITT and Internet Management Coexistence

(IIMC): Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB, Draft 2, May 1993.

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

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

[NMF016] Network Management Forum: Forum 016, Application Services: Security of Management, Issue 1.0, August, 1992.

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

[ECMA138] ECMA-138, Security in Open Systems: Data Elements and Service Definitions, December 1989.

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

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

INTERNET DRAFT - Expires November 29, 1993