ISO/CCITT and Internet Management Coexistence (IIMC):

ISO/CCITT to Internet Management Proxy

(IIMCPROXY)

March 28, 1993

April Chang (Author, Editor)

NetLabs, Inc.
4920 El Camino Real
Los Altos, CA 94022
april@netlabs.com

David Liu (Co-Editor)

Northern Telecom, Inc.
35 Davis Drive
Research Triangle Park, NC 27709
dliu@bnr.com

Status of this Memo

This document provides information to the network and systems management community. This document is intended as a contribution to ongoing work in the area of multi-protocol management coexistence and interworking. This document is part of a package; see also [IIMCOMIBTRANS] [IIMCMIB-II] [IIMCIMIBTRANS] and [IIMCSEC]. Distribution of this document is unlimited. Comments should be sent to the Network Management Forum IIMC working group (iimc@thumper.bellcore.com).

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress.''

Please check the 1id-abstracts.txt listing contained in the internet-drafts Shadow Directories on nic.ddn.mil, nnsc.nsf.net, nic.nordu.net, ftp.nisc.sri.com, munnari.oz.au to learn the current status of any Internet Draft.

Abstract

This document is intended to facilitate the use of the ISO/CCITT Common Management Information Protocol (CMIP) for integrated management of networks via proxy management of TCP/IP networks that are managed using Simple Network Management Protocol (SNMP). This document describes an ISO/CCITT to Internet "proxy" which allows interworking between CMIP-based managers and SNMP-based agents. The proxy emulates CMIS service requests by mapping between corresponding ISO/CCITT GDMO and Internet MIB definitions, and generating SNMP message(s) needed to emulate the service. The proxy also emulates CMIS service responses and notifications by converting incoming SNMP response and trap message(s) in a similar fashion. Thus, the proxy appears as a CMIP-based agent to the manager, and as an SNMP-based manager to the agent. The proxy depends on the availability of corresponding MIB definitions translated as described in [IIMCIMIBTRANS].

Table of Contents

Status of this Memo..... i
Abstract ii
Table of Contents..... ii
Revision History..... iii
1 Introduction..... 4
1.1 Background..... 4
1.2 Overview..... 5
1.3 Scope 7
1.4 Document Registration..... 10
1.5 Terms and Conventions..... 10
1.6 Security..... 11
2 ISO/Internet Proxy Configuration..... 12
2.1 Translated MIB Schema Information..... 12
2.2 Party MIB Objects..... 14
2.3 IIMC Proxy MIB..... 15
2.4 Retained Information..... 16
2.5 Usage 16
3 Elements of CMIS Service Emulation..... 17
3.1 Association Service..... 17
3.2 Object Selection - Scoping and Filtering..... 18
3.3 Management Operation Services..... 20
3.4 Synchronization..... 22
3.5 M-GET Service..... 23
3.6 M-CANCEL-GET Service..... 23
3.7 M-SET Service..... 24
3.8 M-ACTION Service..... 25
3.9 M-CREATE Service..... 25
3.10 M-DELETE Service..... 27
3.11 Management Notification Services..... 28
4 Common Procedures For CMISE Service Emulation..... 29
4.1 Verifying Existence Of An Object Instance..... 29
4.2 Translating Timestamps..... 29

4.4 Derivation Of CMIS Parameters..... 31
5 Error Message Translation..... 35
5.1 Translating SNMP Error Messages..... 35
5.2 CMIS Processing Failure..... 39
6 ISO/CCITT Systems Management Functions..... 40
6.1 Object Management Function..... 40
6.2 State management function..... 40
6.3 Attributes For Representing Relationships..... 41
6.4 Alarm Reporting Function..... 41
6.5 Event Report Management Function..... 41
6.6 Log Control Function..... 42
6.7 Security Alarm Reporting Function..... 43
6.8 Security Audit Trail Function..... 43
7 ISO/CCITT-Internet Proxy MIB..... 43
7.1 Proxy MIB Managed Object Class Definitions..... 43
7.2 Proxy MIB Attribute Definitions..... 46
7.3 Proxy MIB Name Bindings..... 48
7.4 Proxy MIB ASN.1 Modules..... 49
7.5 Party MIB MOCS..... 50
8 Conformance Requirements..... 51
8.1 Management Communication Requirements..... 51
8.2 Management Function Requirements..... 51
8.3 Management Information Requirements..... 51
8.4 Service Emulation Requirements..... 52
9 Abbreviations..... 54
10 Acknowledgments..... 54
Appendix A: Example Operation..... 55
Appendix B: Translated MIB Naming Proposals..... 58
References 61

Revision History

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

Draft 1 - March 28, 1993
Current draft of this document (replaces Draft 0).

Major Changes Since Last Revision

1 Restructured the document, in particular the service emulation and protocol mapping sections.
2. Added text to reflect the support of SNMPv2, including additional error values from SNMPv2.
3. Added a section to describe the support of Systems Management Functions in the proxy
4. Expanded the section on "association service" to address the relationship of CMIP management association and SNMP connectionless transport.
5. Expanded the section on "M-Cancel-Get service" to further the processing requirements on the proxy.
6. Imported the "proxy MIB" from the [IIMCIMIBTRANS]. The "proxy MIB" definition is modified.
7. Changed the Management Notification Service.

Action Item Proposals Contained In This Document

#1 Add run-time support for mapping SNMPv2
#2 Add optional requirement for EFD, Log SMFs
#3 Propose association control extensions/requirements
#4 Proxy State requirements
#5 Proxy MIB knowledge requirements
#6 Revise Scoping algorithm
#7 Revise Trap to Notification mapping.
#8 Minimum requirement for CMIP conformance (2 alternatives)
#9 Reply to INTAP re: SMF support (see also action #2)
#11 Propose additional text re: actualClass support
#23 Isolation of service emulation and protocol mapping

Outstanding Issues

In addition to the proposed action item resolutions listed above, the following issues remain outstanding. Comment on both these issues (and the proposed action resolutions) are solicted during review.

1. Conformance to this document (Action #8)
2. MIB naming hierarchy (Actions #15, #18)
3. Security model (Action #22)
6. Propose "stateful" optimizations (Action #10)

1 Introduction

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

1.1 Background

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

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

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

This work was initiated, in part, by NM Forum efforts to translate RFC 1214 for use with OMNIPoint 1 implementations. Through this effort, it became obvious that end-to-end management requires an integrated, unified view of the managed network, despite differences in management protocol and information structure. Integrated management can be facilitated by the development of "proxy" mechanisms which translate between functionally equivalent service, protocol, and SMI differences to create this unified view. MIB translation procedures can be used to support proxy management, as well as to take advantage of existing MIB definition and avoid duplication of effort. In this way, commercial investment in both ISO/CCITT and Internet-based management technologies can be preserved through deployment of common methods and tools which support integration.

This overall strategy was outlined in a joint publication developed by the NM Forum and X/Open entitled "ISO/CCITT and Internet Management: Coexistence and Interworking Strategy"

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

1.2 Overview

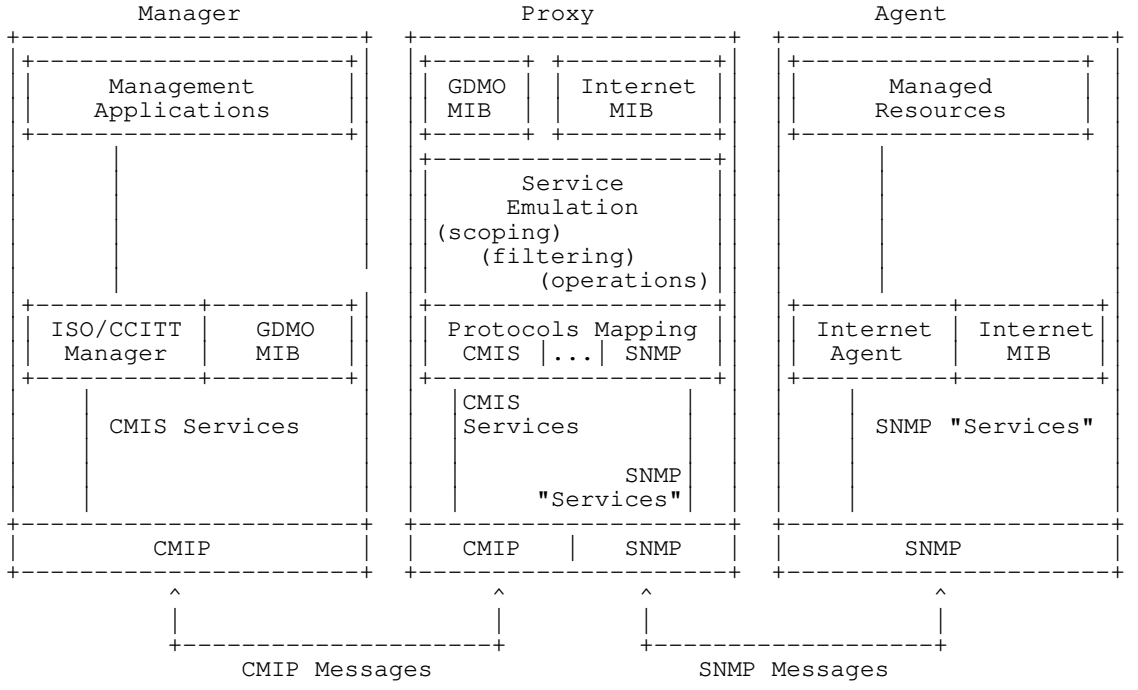
The response to the need for OAM&P of enterprise networks has been the development of network management standards within various networking communities - most notably the ISO/CCITT and Internet communities. However, coordination of standards activities between these two communities has not occurred.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

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

The focus on the need for end-to-end enterprise management has indicated the need to integrate the management of components accessed by ISO/CCITT management, Internet management and proprietary management mechanisms in a manner which presents a unified view of the network, despite protocol and SMI differences. One way to integrate management is by the development of "proxy" mechanisms which translate between functionally equivalent services, protocol and SMI differences to create this unified view.

A body of telecommunications and computer vendors, represented by organizations such as the Network Management Forum (NMF), and the U.S. government, as specified in the Government Network Management Profile (GNMP) have based their integrated management model on the ISO/CCITT management model using CMIP and the ISO/CCITT SMI. These organizations are particularly interested in the development of proxies for devices that use the Internet management protocols and SMI. Their interest is primarily due to the widespread commercial implementation and use of such devices within their enterprises, especially devices that use the Internet TCP/IP protocol suite.

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



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

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

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

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

Finally, note that MIBs translated by procedures such as those defined by [IIMCIMIBTRANS] and [IIMCOMIBTRANS] may also be used without a proxy. For example, a translated MIB may be used to take advantage of existing MIB definitions when business needs require deployment in a different management environment. Translated MIBs may also be used to provide uniformity when multiple management environments are supported by a single system (e.g., dual stack managers).

1.3 Scope

The intent of the document is to facilitate the use of ISO/CCITT CMIP-based managers to perform integrated management of networks via proxy management of networks that are accessed using Internet SNMP-based agents. There are two major differences between CMISE and SNMP services: the structure of management information, and the management operations supported by the underlying protocols. The ISO/Internet Proxy architecture as shown in section 1.2 provides CMISE service emulation. In another words, the ISO/Internet Proxy acts as a CMIP-based agent with respect to the manager, allowing management of Internet objects by the ISO/CCITT manager. CMIS requests are processed by the ISO/Internet proxy and CMIS responses are returned by the ISO/Internet proxy. SNMP traps and Inform requests are converted to CMIS notifications by the ISO/Internet proxy. The implementation of the proxy requires that the Internet MIBs be mapped to ISO/CCITT GDMO definitions.

1.3.1 Approaches to Service Emulation

As described by [NMFMC92], there are different approaches for mapping Internet MIBs and ISO/CCITT MIBs.

- The "direct translation" approach maps each Internet object to a newly defined ISO/CCITT GDMO object that contains: 1) the same information as contained in the Internet object; and 2) the attributes that are inherited from the ISO/CCITT Top object class.

- The "abstract translation" approach maps Internet objects to different ISO/CCITT GDMO objects. For example, the MIB-II system object is similar to, and could be represented by, the ISO/CCITT system object. The abstract translation approach can also be used to map several Internet objects to a single ISO/CCITT GDMO object which provides only a summary view of the original Internet objects.

Either or both approaches could be used by an ISO/CCITT manager to manage Internet agents. This document uses the "direct translation" approach.

To perform the CMISE service emulation, the ISO/Internet proxy can use either of the approaches described by [NMFMC92] to retrieve or modify Internet MIB information.

- In the "stateless" approach, the proxy does not maintain the Internet agent's MIB data. Instead, for each received CMIS request, the ISO/Internet proxy generates one or more SNMP requests to the Internet agent in order to achieve the same intent of the CMIS request.

- The "stateful" approach requires the proxy to replicate an Internet agent's MIB locally, and to send periodic (unsolicited) requests to Internet agents to keep the replicated MIB current. The ISO/Internet proxy then tries to fulfill each incoming CMIS request by using locally-replicated MIB data, instead of sending SNMP requests to the Internet agent.

The stateful approach will usually provide better response time, but has the drawback that the data retrieved might not be current. In this approach, the poll frequency used to update the locally-replicated MIB has a significant effect on the accuracy of the response.

This document uses the state-less approach in which the proxy responds to incoming CMIS requests by generating appropriate SNMP requests. Furthermore, SNMP traps and inform requests are converted to CMIS notifications.

If necessary, the static Internet MIB data retrieved by the ISO/Internet proxy could be cached by the proxy in order to improve the response time of an operation. This document makes no assumption that the proxy caches static information, and so takes no advantage of information which might be cached.

1.3.2 Proxy Inputs and Outputs

This document describes a proxy which emulates CMIS services through generation of appropriate SNMP protocols. The proxy is based on certain inputs and outputs, as shown below in Tables 1 and 2.

CMIS services [ISO9595] are supported by CMIP version 2 protocol [ISO9596-1]. SNMP protocols are as defined for SNMPv1 [RFC1157] and SNMPv2 [SNMPv2PROT]. This specification assumes that the reader is familiar with the ISO/CCITT SMI [ISO10165-1] and the Internet SMIs [RFC1155] and [SNMPv2SMI], and the terminology of each. The emulation is slightly different, depending upon whether SNMPv1 or SNMPv2 protocols are being used. The term SNMP will be used throughout this specification to indicate either SNMPv1 or SNMPv2, unless a distinction needs to be made.

Service	Source
ACSE Associate Indication	CMIP Stack
ACSE Release Indication	CMIP Stack
ACSE Abort Indication	CMIP Stack
CMIS Get Indication	CMIP Stack
CMIS Cancel Get Indication	CMIP Stack
CMIS Set Indication	CMIP Stack
CMIS Create Indication	CMIP Stack
CMIS Delete Indication	CMIP Stack
CMIS Action Indication	CMIP Stack
CMIS Event Report Confirm	CMIP Stack
SNMPv1 Get Response	SNMPv1 Stack
SNMPv1 Trap	SNMPv1 Stack
SNMPv1 Error	SNMPv1 Stack
SNMPv2 Trap	SNMPv2 Stack
SNMPv2 Get Response	SNMPv2 Stack
SNMPv2 Get Bulk Response	SNMPv2 Stack
SNMPv2 Inform Request	SNMPv2 Stack
SNMPv2 Error	SNMPv2 Stack

Table 1 - Proxy Inputs

Service	Target
ACSE Associate Response	CMIP Stack
ACSE Release Response	CMIP Stack
ACSE Abort Request	CMIP Stack
CMIS Get Response	CMIP Stack
CMIS Cancel Get Response	CMIP Stack
CMIS Set Response	CMIP Stack
CMIS Create Response	CMIP Stack
CMIS Delete Response	CMIP Stack
CMIS Action Response	CMIP Stack
CMIS Event Report Indication	CMIP Stack
SNMPv1 Get Request	SNMPv1 Stack
SNMPv1 Set Request	SNMPv1 Stack
SNMPv1 Get Next Request	SNMPv1 Stack
SNMPv2 Get Request	SNMPv2 Stack
SNMPv2 Set Request	SNMPv2 Stack
SNMPv2 Get Next Request	SNMPv2 Stack
SNMPv2 Get Bulk Request	SNMPv2 Stack

Table 2 - Proxy Outputs

This document assumes that CMIP PDUs and SNMP PDUs received by the ISO/Internet proxy are always properly encoded (i.e., the underlying protocol stacks ensure the correctness of the service indications and confirmations that are passed up to the ISO/Internet proxy).

1.4 Document Registration

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

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

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

1.5 Terms and Conventions

1. I**SO/CCITT manager:** An application entity that implements [ISO9596-1] and acting in the manager role.

2. **Internet agent:** An application entity that supports the agent role of one or more of the SNMP protocols, such as [RFC1157] or [SNMPv2PROT].

3. **ISO/Internet Proxy:** An application entity that is responsible for emulating CMIS requests by a) generating SNMP requests, b) using SNMP responses to generate CMIS responses, and c) mapping SNMP Traps and InformRequests

to CMIS notifications, all between a given (ISO/CCITT manager, Internet agent) pair. A proxy may concurrently support more than one (ISO/CCITT manager and Internet agent) pair.

4. **Known Internet agents:** A set of one or more Internet agents that an ISO/Internet proxy has knowledge of. Each known Internet agent is represented by an instance of the proxy object. This document defines the cmipsnmpProxyAgent object class.

5. **Known SNMP Parties:** A set of one or more SNMP parties that an ISO/Internet proxy has knowledge of. Each known SNMP party is represented by an instance of the partyTable object. The partyTable object class is defined in [IIMCSEC].

6. **Pseudo Object:** A GDMO object class that does not contain any attributes which may be retrieved from an Internet agent (for example, a GDMO object class that represents a group in the Internet MIB-II, or any GDMO object classes representing Internet MIB tables).

7. **Local object (instance):** An object instance that is implemented by the proxy itself (for example, the cmipsnmpProxyTable and cmipsnmpProxyAgent classes defined in section 7).

8. **Remote object (instance):** An object instance that physically resides within an Internet agent is considered a "remote object" (for example, Internet MIB-II objects like system, tcp, and udp).

9. **Multiple Instance Object:** An object class that may have more than one object instance. For example, Internet MIB table entries.

10. **Delete Information:** The object identifier of the attribute and its attribute value used to indicate that a particular row of a table is deleted.

1.6 Security

The security architecture, services, protocols, and mechanisms for the ISO/Internet proxy shall be as defined in [IIMCSEC].

2 ISO/Internet Proxy Configuration

In order for the ISO/Internet proxy to interwork with the known Internet agents, the proxy needs to know initialization information such as the transport address, network address, protocol version, and security policy for each of the known Internet agents. Such configuration may be done through an off-line process, or through an on-line management exchange not specified by this document.

2.1 Translated MIB Schema Information

To perform CMISE service emulation, the ISO/Internet proxy requires the Internet MIB's schema information, described in ISO/CCITT GDMO templates. These templates shall be derived from the original Internet MIB according to the procedures defined by [IIMCIMIBTRANS].

The proxy run-time translation of parameters and protocol translation procedures defined in this document depend on the MIB translation, naming and registration procedures defined in [IIMCIMIBTRANS]. The translation and registration procedures defined in that document are structured such that the maximum amount of information is preserved to facilitate the translation process.

2.1.1 Translated MIB Containment Tree

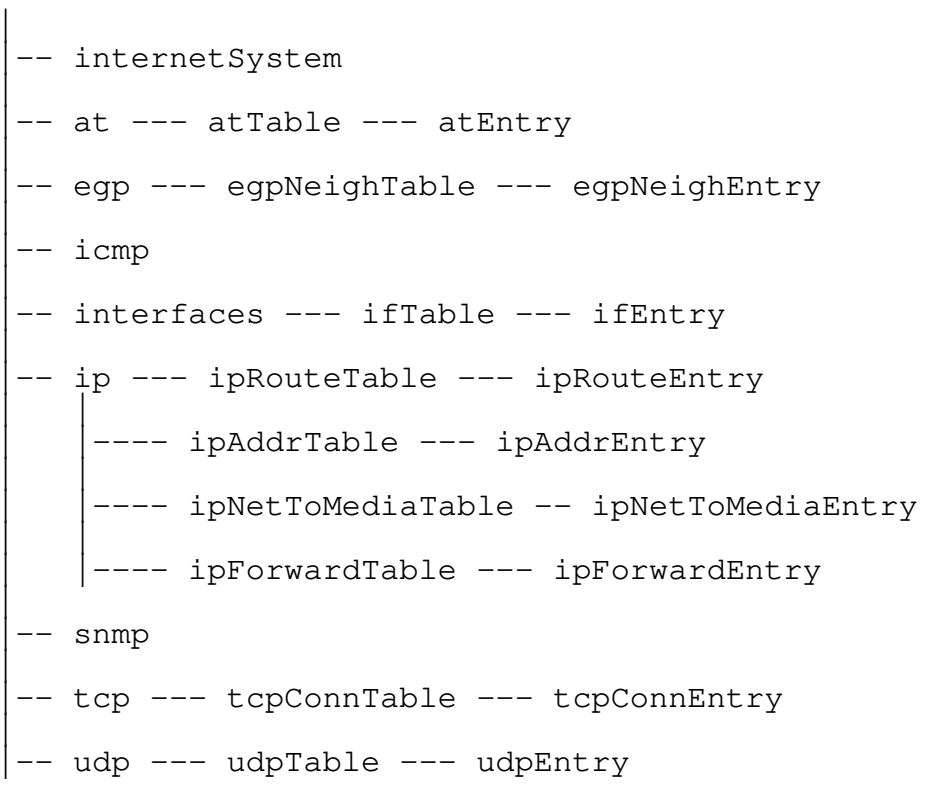
The proxy shall support a forest of object instance trees (i.e., multiple trees rooted at the ISO/CCITT system managed object defined by [ISO10165-2]), with one system object instance for each supported Internet agent, and one system object instance for the proxy itself. The ISO/CCITT system objects are distinguished by the value of the systemTitle attribute, which contains the name associated with the Internet agent or proxy application.

Editor's Note: [The above proposal is presented in Draft 1 documents. However, there are a few other alternatives to this proposal. Readers should refer to Appendix B, and comment on both proposals.]

2.1.2 Example Containment Tree

An example containment tree for an agent supporting the ISO/CCITT GDMO Internet MIB-II [IIMCMIB-II] (derived from the Internet MIB-II [RFC1213]) is illustrated below. A proxy would have multiple instances of such a tree for each Internet agent supported. (The actual structure of the each containment tree depends upon the MIB(s) supported by the proxy.)

"Rec. X.721 | ISO/IEC 10165-2 : 1992" : system



As specified in [IIMCIMIBTRANS], name bindings for ISO/CCITT GDMO object classes derived from Internet MIB table and entry types can be automatically inferred from the Internet registration hierarchy. Thus, object classes derived from Internet conceptual table objects are bound to the object class derived from the group with which the table is associated. Object classes derived from Internet conceptual table entries are bound to the table object classes with which the tables entries are associated. Also, object classes derived from Internet groups are bound to the ISO/CCITT system object class.

2.1.3 Creation/Deletion of System Objects

The system object particular to an Internet agent system shall be automatically created/deleted by the proxy when an associated cmipsnmpProxyAgent object instance in the Proxy MIB is created/deleted. Creation/deletion of the system object via management operation is not allowed.

The system object particular to the proxy shall be created automatically by the proxy during its local initialization.

2.1.4 Creation/Deletion of Capability Objects

If used, the "OP1 Library Vol.4":capability object particular to an Internet agent system shall be automatically created and deleted by the proxy when the associated cmipsnmpProxyAgent object in the Proxy MIB is created and deleted.

Editor's Note: [Use of the capabilityObject defined by [OP1LIBV4] is proposed in section 2.3. Please comment on this proposal.]

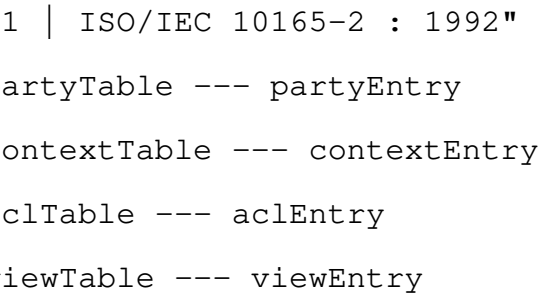
2.2 Party MIB Objects

Information regarding security policy when accessing agents is contained in Party MIB objects. Binding the Party MIB objects as subordinates of the system object which represents an individual Internet agent allows security policy to be applied on a per Internet agent basis. The Party MIB information can be used by the proxy in a manager role when security services enforcing security policy are implemented in the Internet agent. The services enforced may be authentication, access control, confidentiality and integrity as defined in [SNMPv2SEC].

In those situations where the agents may not implement the access control security service on requests from the ISO/CCITT manager (e.g., SNMPv1 agents), the proxy may enforce those services on behalf of the Internet agent. The policy regarding where access control is to be applied is controlled by variables in the cmipsnmpProxyTable and cmipsnmpProxyAgent managed objects defined in section 7.

The policy regarding security services other than access control (e.g., authentication, data origin integrity, and confidentiality), must always be enforced by the Internet agent.

A containment 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.



Editor's Note: [It may not be appropriate to bind these tables under the system object. Another alternative is to bind these tables under the associated cmipsnmpProxyAgent object.]

2.3 IIMC Proxy MIB

The IIMC Proxy MIB defines a set of objects for specifying the information that is needed for both community-based and party-based SNMP management on a per Internet agent basis.

The Proxy MIB consists of a cmipsnmpProxyTable managed object class which contains cmipsnmpProxyAgent object classes, one for each agent being managed by the proxy. The cnipsnmpProxyTable object class is an immediate subordinate of the ISO/CCITT system object class that represents the proxy.

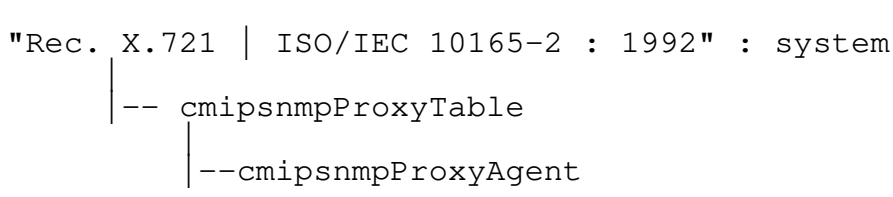
The cmipsnmpProxyAgent has information to identify Internet agents and how they may be reached. Its naming attribute, which contains the administratively-assigned name of the managed device where the Internet agent is located, is used in the naming tree to identify the SNMP managed device.

Creation of a cmipsnmpProxyAgent object instance to represent an Internet agent shall result in the instantiation of a corresponding ISO system object for the Internet agent. The naming attribute value of the ISO system shall be the same as the corresponding cmipsnmpProxyAgent. It is recommended that a "OP1 Library Vol. 4":capabilityObject be created for the proxy also.

The cmipsnmpProxyAgent object may be created by management operation, or automatically. For example, the proxy may support discovery of Internet agents, whereby the discovered Internet agents, associated system object, and capability object shall be created automatically by the proxy itself.

This document refers to instances of IIMC Proxy MIB object classes as "local objects" or as "local object instances".

A containment tree diagram for ISO/CCITT proxy MIB managed object classes is illustrated below.



IIMC Proxy MIB GDMO definitions are described in section 7.

2.4 Retained Information

The proxy must retain information obtained from the ISO/CCITT manager during association establishment, and for individual CMIS requests on the association.

For each outstanding CMIS request, the proxy needs to maintain the ISO/CCITT invoke id, object class and object instance. When SNMP responses are received, the proxy shall use the retained information to form the associated parameters in CMIS responses.

For scoped CMIP requests, the proxy shall maintain some state information to keep track of the portion of the Internet MIBs that is being traversed.

2.5 Usage

The information described in sections 2.1 through 2.4 is maintained by the proxy and used to perform run-time translation between corresponding CMIS and SNMP parameters.

The following definitions are extracted from [IIMCIMIBTRANS] clause 2.3.1, where (c) and (a) refer to class and attribute, respectively.

From [IIMCIMIBTRANS] clause 2.1:

```
{classOID}     ::= {iimcAutoTrans
                    <internetEntityId>(c)}

and

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

From [IIMCIMIBTRANS] clause 2.2, the ISO/CCITT naming attribute value contains the OID formed as:

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

where the "()" indicates "contents of".

The <internet instanceId> (the OID created uniquely for each Internet object instance) is "0" for object classes that may only have a single instance. The <internet instanceId> for object classes that may have multiple instances is an OID fragment derived from the values of the internet objects identified in the INDEX (or AUGMENTS) clause of the Internet Macro from which the object class is derived, as defined in [RFC1155] or [SNMPv2SMI].

The Internet uses the following convention to uniquely identify an Internet object instance:

```
{internet object name}::= {<internetEntityId>(a)
                           <internet instanceId>}
```

3 Elements of CMIS Service Emulation

The following sections describe the conceptual process for performing CMIS service emulation. In an actual implementation, it should be possible to combine some of the processing. It is highly recommended that the implementors of the ISO/Internet proxy combine the processes where possible to optimize the implementation.

3.1 Association Service

The proxy should provide the association service as defined in section 8.1 of [ISO9596-1]. This service includes association establishment and association release.

In ISO/CCITT systems management, management entities may exchange initialization information during the association establishment phase. Such information is used only by the proxy for its own configuration and is not conveyed to the communicating Internet agents.

The negotiation of application context and functional units between the ISO/CCITT manager and the ISO/Internet proxy is optional. This document does not define any application context; however, a proxy may be required to support the following application contexts as defined in the ISO standards and CCITT recommendations:

ISO Systems Management application context; or

CCITT TMN application context one.

CMIP and SMASE functional units may be negotiated between the ISO/CCITT manager and the ISO/Internet proxy. Once a set of functional units is agreed, the proxy will ensure only the agreed services are accepted over the association.

Editor's Note: [The above discussion requires review, since exchange of application context and CMIP functional units is mandated by the ISO/CCITT standard. Refer also to the discussion of conformance requirements in section 8.]

The CMIP protocol used between the ISO/CCITT manager and the ISO/Internet proxy is a connection-oriented protocol which requires an association be maintained throughout the management exchange(s). The protocol between the proxy and the Internet agent, however, may be a connection-less protocol which does not require the existence of an association. Upon receiving an association request from the ISO/CCITT manager, the proxy needs to determine whether connectivity to the agent is possible so that it may accept or reject the association request accordingly. The mechanism used by a proxy to detect the "reachability" of an Internet agent is implementation-dependent, and is not within the scope of the document.

For example, if the reliability of the association is not essential to its management applications, a proxy may assume its Internet agents are always reachable, and may accept association requests on that basis. In this approach, the proxy would terminate the association only when it detects the Internet agent is not reachable.

3.2 Object Selection - Scoping and Filtering

Managed object selection is used to identify a set of managed object instances in the management information tree (MIT) to which a CMIS request applies. Managed object selection is performed in two phases: scoping; and then, filtering. Scoping is used to select candidate object instances in the MIT to which operations may apply. A filter is then applied to attributes of the previously scoped object instances in order to identify the subset of object instances on which the CMIS operation is to be performed.

If no filter is specified, the CMIS request will be performed for all object instances identified by the scope parameter. If no scope parameter is specified, the default is the base managed object instance only.

There are different ways of performing the scoping operation, depending on the implementation. This document specifies one possible way of providing the managed object selection service.

The proxy has no direct knowledge of current object instances that exist in the Internet agent. Therefore, it must first determine the existence of an object before it knows whether it is within the scope. Obviously, all objects in the scope must be instances of object classes that are within the scope. Thus, the proxy should first determine the set of object classes within the scope, and then discover what instances of those object classes actually exist in the Internet agent. This set of object classes that are within the scope are called the "object class group" (OCG).

The following pseudo code algorithm specifies a generic method of determining the members of an "object class group".

Define the set of object classes at the current level in the naming tree that are being processed as the "class level group" (CLG).

Define the set of object classes at the next level in the naming tree that are to being processed after the CLG as the "next level group" (NLG).

Define the managed object class named by the incoming request as the "base managed object" (BMO).

Minimum Level and Maximum Level are derived from the CMIS scope parameter.

```
CLG = {BMO};
OCG = {}; /* empty set */
currentScope  = 0;

WHILE ( currentScope <= Maximum Level )
     NLG = {children of objects in CLG};
     IF (currentScope > Minimum Level)
               THEN OCG = {union of OCG and CLG};
     CLG = NLG;
     currentScope = currentScope + 1;
     ENDWHILE;
```

The determination of the set NLG as {children of objects in CLG} may be done using implementation-dependent internal data structures of the proxy.

An alternative method to determine NLG is to use the name binding templates directly. The following algorithm could then be used to determine the NLG.

```
WHILE (CLG not equal to {})
     Remove an object from CLG;
     FOR (all name bindings in this proxy's associated
          Capability object)
          IF object == SUPERIOR
               THEN NLG = {union of the
                         SUBORDINATE object and NLG};
               ENDFOR;
     ENDWHILE;
```

3.3 Management Operation Services

If the specified instances (i.e., those selected by scoping process) are "local objects", the proxy performs the services using local means.

If the specified instances are "remote objects", then the following steps apply. Any objects that physically reside in the Internet agents are considered "remote objects". For example, Internet MIB II objects like system, tcp, and udp are considered "remote objects".

1) Determine if the attributes specified by the filter expression (if any) belong to the object class. If not, then remove the object class from the object class group.

2) If the object is a pseudo object (table object), then there is only one possible instance and the values of the attributes are known locally to the proxy. The pseudo object exists if and only if there exists a non-pseudo subordinate object (table entries). The proxy shall attempt to determine if there exists a non-pseudo subordinate object by issuing an SNMP GetNext Request using as an argument the Internet object name for the pseudo object, <internetEntityId>(c). If the SNMP response contains an Internet object that translates to an attribute of a child of the pseudo object, then the pseudo object exists. If the pseudo object does not exist, then remove it from the object class group.

3) If the object is not a pseudo object, then determine if an instance of the object exists by attempting to retrieve, for the first instance, all of the attributes specified in the CMIS filter and the attributeId list or attribute list.

   i) For single instance objects, use an SNMP Get Request or GetNext Request, with the parameters translated according to section 4.

ii) For multiple instance objects, use an SNMP GetNext Request, with the parameters translated as shown in section 4, and with the <internet instanceId> set to zero for all Internet object names. This will result in the retrieval of the first instance of the translated Internet objects (i.e., a table entry). If the resulting table entry has been deleted, or is otherwise unavailable for retrieval, then go to step 5.

4) Apply the filter to the attributes of the object instance identified in steps 2 and 3.

i) If the filter evaluates to FALSE, then perform no further processing on the object. However, for multiple instance objects, save the results of step 3 part (ii).

ii) If the filter evaluates to TRUE, then attempt to perform the operation on the object, as follows.

- If the CMIS operation was M-GET, then the M-GET has been completed from the Internet agent. Formulate the appropriate CMIS M-GET response and send it to the manager.

- If the CMIS operation was M-SET, then perform the corresponding SNMP Set Request on the Internet objects. When the SNMP Response returns, formulate the appropriate CMIS M-SET response and send it to the manager.

- If the CMIS operation was M-CREATE, then perform the create on the Internet objects (conceptual row elements) using the algorithm appropriate to the object. When the create process is finished, formulate the appropriate CMIS M-CREATE response and send it to the manager.

- If the CMIS operation was M-DELETE, then perform the delete on the Internet objects. When the SNMP Response returns, formulate the appropriate CMIS M-DELETE response and send it to the manager.

5) For multiple instance objects, use an SNMP GetNext Request, with the parameters translated according to section 4, and with the <internet instanceId> set to the values retrieved for the previous instances for all Internet object names. This will result in the retrieval of the next instance of the translated Internet objects. If the Internet object instances are not of the same type as those requested, then all instances of the

multiple instance object class have been processed; go to step 6. If the Internet object instances are of the same type as those requested then retain the results of the GetNext Request for the next iteration and repeat steps 4 and 5.

6) Attempt to select another object class from the object class group. If one exists then go to step 1; otherwise, return an appropriate final CMIP PDU (e.g., empty M-GET or M-SET response) and quit processing the request.

3.4 Synchronization

If the ISO/Internet proxy receives a CMIS "atomic" request, but cannot perform the operation atomically, the "synchronization not supported" CMIS error response should be returned to the ISO/CCITT manager.

The types of atomic requests that the ISO/Internet proxy can perform are as follows.

1) If all the instances selected by a scoped CMIS request are "local object instances", then the ISO/Internet proxy can perform the CMIS request locally (and atomically); and

2) If the CMIS request can be performed by the ISO/Internet proxy using a single SNMP request, then the operation can also be performed atomically.

For a "best effort" request, the ISO/Internet proxy should try to perform the request on all the instances specified by the request. Since the SNMP protocol supports only "atomic" operations, an operation (especially an SNMP Set Request operation) on multiple variables may be rejected if the operation on any one of the selected variables failed. Upon receiving such an error, the proxy should retry the request by sending multiple requests with each request containing only a single variable. In the time window in which these SNMP requests are being processed, another SNMP Set Request could be issued which could modify the value of a selected variable. Because of this, the complete integrity of a CMIS scoped request cannot be guaranteed. A proxy which complies with this document is not required to detect or avoid this situation, and will not usually report any error if this situation occurs.

3.5 M-GET Service

The following sub-sections describe how the M-GET service may be emulated. Upon receiving a CMIS M-GET request, the proxy first verifies the existence of the based managed object. The procedures for verifying the existence of a managed object is described in section 4.1.

3.5.1 Form The Request

If the CMIS request's attributeIdList parameter is empty (selects all attributes), the proxy shall query the schema information to find out what attributes are specified for the requested object class.

If the CMIS request's attribute specifies a non-null attributeIdList, the proxy shall verify that the identified attribute(s) are specified for the object class. If the identified attribute is not specified in the object class, the proxy shall return a noSuchAttribute CMIS error without sending SNMP requests to the Internet agent.

For all attributes that are specified in the object, an SNMP Get or SNMP GetNext Request shall be formed, based on the mapping specified in section 4. Use SNMP Get or GetNext is an implementation issue; however, SNMP GetNext is recommended for performance reasons. Since some non-conforming agents may not implement all the object types in an object group, SNMPv1 Get would return a noSuchName error in this case, and the proxy will need to remove the non-implemented variable binding and resend the SNMP Request. If SNMP GetNext is used instead, the proxy would either discard the non-implemented attribute or translate the SNMP Response to appropriate CMIS getListError.

3.5.2 Form The Response(s)

The proxy shall form the CMIS response according to the mappings specified in section 4.

If the CMIS request's attributeIdList is null (selects all attributes), the proxy shall never return the CMIS getListError. If the Internet agent does not implement all the variables in an object (which violates conformance to the SNMP specification), the proxy shall form the CMIS M-GET response with all the attributes implemented by that Internet agent.

If the CMIS request's attributeIdList selects all attributes, the proxy shall supply in all the attributes that are inherited from the ISO/CCITT Top object in the CMIS response.

3.6 M-CANCEL-GET Service

The M-CANCEL-GET operation shall be performed as described in [ISO9596-1]. The ISO/Internet proxy does not need to generate any SNMP Requests in order to emulate the CMIS M-CANCEL-GET request. However, upon receiving an M-CANCEL-GET request, the ISO/Internet proxy shall stop sending further CMIS M-GET responses to the ISO/CCITT manager for the canceled M-GET request. Furthermore, the proxy shall not initiate further SNMP Requests to the Internet agent for the canceled M-GET request. If the Internet agent continues to return SNMP Get responses corresponding to the canceled M-GET request, they shall be discarded by the proxy.

Depending on when an M-CANCEL-GET request is received, the proxy may send out different responses for the canceled M-GET request and for the M-CANCEL-GET request.

If the Invoke Id of the M-GET request to be canceled is not recognized by the proxy, the proxy shall return a "no such invoked identifier" CMIS error to the ISO/CCITT manager. This can happen when the proxy has not received such an M-GET request, or when the proxy has completed the identified M-GET request.

An M-GET operation is considered completed if the corresponding M-GET response has been sent. For the single object M-GET case, this means the sending of a single M-GET response. For the scoped multiple object case, this means the sending of the final empty M-GET response for the linked replies.

If the identified M-GET request was received, but has not been completed, the proxy generates an "operation canceled error" to the ISO/CCITT manager as a response to the canceled M-GET request. In this case, the proxy will also acknowledge the successful completion of the M-CANCEL-GET request to the ISO/CCITT manager.

3.7 M-SET Service

The following sections describe how M-SET service may be emulated. Upon receiving a CMIS M-SET request, the proxy verifies the existence of the based managed object, according to the procedures defined in section 4.1.

3.7.1 Perform The Set Operation

For each selected ISO/CCITT object instance, the proxy would generate one or more SNMP Set Requests to modify the attributes identified by the CMIS modificationList parameter, according to the specified modify operator. Only the "replace" modify operator is supported by the ISO/Internet proxy. The modify operator is optional and if it is not specified in a CMIS request, the "replace" operator should be assumed.

The CMIS "add value" and "remove value" modify operators are not supported by SNMP protocol, and are not supported by the ISO/Internet proxy. Since SNMP uses default values only for initialization (i.e. at creation time), the "set to default" modify operator is not supported by the ISO/Internet proxy either. If the modify operator value included in an M-SET request is not supported, "invalid operator" should be reported in the CMIS setListError response.

3.7.2 Form The Response(s)

If the M-SET request is a "confirmed" request, the proxy shall construct an M-SET response. The CMIS M-SET response should contain the attribute values list or the appropriate setListError. Once the CMIS M-SET response has been constructed, it is passed to the CMIP service provider, which send the corresponding CMIP PDU to the ISO/CCITT manager.

If the CMIS M-SET request is a scoped request, attribute values of each ISO/CCITT object are reported as a linked reply.

3.8 M-ACTION Service

Since Internet MIBs do not have any actions defined, the translation of CMIS M-ACTION to corresponding SNMP operations is not needed. Any CMIS M-ACTION request which is received pertaining to a translated Internet MIB object will be rejected by the proxy with an "noSuchAction" error response. However, CMIS M-ACTION may be used by the proxy for other purposes.

3.9 M-CREATE Service

3.9.1 Request Validation

The ISO/Internet proxy is responsible for validating that incoming CMIS M-CREATE requests do not violate name binding and object class definitions.

3.9.2 Name Binding

The ISO/Internet proxy must determine if an instance may be created according to the CREATE clause of the NAME BINDING template specified for the object class. If the instance cannot be created, the CMIS error response "classInstanceConflict" is returned.

The ISO/Internet proxy must also determine from the NAME BINDING template if the instance specified in the request maybe created under the superior object instance identified in the M-CREATE request. If the NAME BINDING does not specify the identified containment relationship, an "invalidOperation" CMIS error response should be returned.

3.9.3 Check For Duplication

The proxy must determine if the instance already exists. If it does, a "duplicate managed object instance" CMIS error response should be returned.

3.9.4 With Referenced Object

If a CMIS M-CREATE request includes a reference object, the ISO/Internet proxy should retrieve the referenced object instance from the Internet agent.

The proxy uses an SNMP GetNext Request for retrieval, with the parameters translated according to section 4, and with the <internetinstanceId> set to the translated <internet instanceId> of the reference object for all Internet object names.

The proxy checks if the attribute used for SNMP row creation indicates that the row is not available for use (e.g., has been deleted or is in some other not ready condition). This attribute is the CREATEDELETEATT attribute indicated in the parsable portion of table entries translated according to [IIMCIMIBTRANS].

If the reference object instance does not exist, the proxy must send a "No such reference object" CMIS error response to the ISO/CCITT manager.

3.9.5 With Automatic Instance Naming

A CMIS M-CREATE request can use automatic instance naming to form a name for the object instance to be created. Automatic instance naming is used if: a) a CMIS M-CREATE request does not specify a distinguished name for the object instance to be created; and b) the request specifies an object class that has a name binding allowing automatic instance naming.

It is the responsibility of the ISO/Internet proxy to select an instance name based on the behavior of the object class and name binding. In some cases, the relative distinguished name (RDN) is formed using attributes provided in the CMIS M-CREATE request. For example, the RDN for the Internet MIB-II "atEntry" could be formed from the "atNetIfIndex" attribute and the "atNetAddress" attribute. In other cases, the RDN can be assigned by the ISO/Internet proxy.

If the superior object instance is not specified, the ISO/Internet proxy cannot create the object instance and a "processing failure" CMIS error should be returned.

3.9.6 Perform The Create Operation

The CMIS M-CREATE is realized by setting the status column of the corresponding Internet MIB table entry to a valid value

with all other columns of the table entry properly initialized. If the combination of the attributes specified in the CMIS M-CREATE request and the attributes obtained from the reference object do not provide a complete set of attribute values for all of the mandatory attributes for the entry specified by the object class being instantiated, then the ISO/Internet proxy should still try to create the object with all the available attributes.

If the actual creation process with the incomplete attribute list succeeds, the ISO/Internet proxy should retrieve all the attributes of the newly-created entry, including the attributes which have values supplied by the Internet agent with using default values. This complete list of attribute values is returned in the CMIS M-CREATE response.

If the actual creation process with this partial attribute list fails, the ISO/Internet proxy sends a "missing attribute value" CMIS error back to the ISO/CCITT manager.

3.9.7 Form The Response

The results from the Internet agent are used by the proxy to construct a CMIS M-CREATE response, which is then returned to the ISO/CCITT manager, using the mappings defined by section 4.

3.10 M-DELETE Service

3.10.1 Perform the Delete Operation

For all the selected ISO/CCITT object instances, the following procedures should be taken.

3.10.2 Name Binding

Determine from the NAME BINDING template if the instance specified in the request may be deleted. If the name binding does not allow the deletion of the identified object, a CMIS error response is returned.

3.10.3 Perform The Delete Operation

If the object instance identified in the CMIS M-DELETE request exists, the delete operation is performed. In SNMPv1, object deletion is achievable only if there is a columnar object representing the status of each conceptual row. Deleting an object instance is realized by setting the status columnar object to an invalid value. The value representing "invalid" is implementation-specific. The proxy therefore needs to be aware of the "invalid" value and the status columnar object in order to perform the deletion. For SNMPv2, the object deletion can be achieved by sending an SNMP Set Request to the Internet agent to change the Row Status value to "destroy."

3.10.4 Form The Response(s)

This process includes formatting the CMIP M-DELETE response with the appropriate attribute list or deleteListError parameters. Once the CMIS M-DELETE response has been constructed, it is returned to the ISO/CCITT manager.

3.11 Management Notification Services

Although SNMPv1 and SNMPv2 use different PDU structures to convey Traps, all SNMP Traps are mapped to the IIMC-defined internetAlarm notification and sent as CMIS M-EVENT-REPORTs. Since SNMP Traps are not confirmed, the translated CMIS M-EVENT-REPORTs are sent as "unconfirmed" event reports.

If the SNMPv2 manager-to-manager communication is supported between an Internet manager and an ISO/CCITT manager, it is possible for the proxy to receive an InformRequest from the Internet system. Like Traps, InformRequests are also mapped to CMIS M-EVENT-REPORTs. Unlike Rraps, the internetAlarm notifications resulting from InformRequests are sent as "confirmed" event reports.

If the translation of Traps to notifications fails, no CMIS M-EVENT-REPORT will be generated and the SNMP Traps are simply discarded.

The proxy shall expect a CMIS M-EVENT-REPORT response for all internetAlarm notifications sent in confirmed mode. The CMIS M-EVENT-REPORT response shall contain an empty event report argument. Upon receipt of the CMIS M-EVENT-REPORT response, the proxy shall return an SNMP Response PDU to the Internet agent that is in accordance with SNMPv2 protocol rules and contains an error code of "noError".

If the translation of an SNMPv2 InformRequest to a CMIS M-EVENT-REPORT fails, the proxy shall send an SNMP Response to the originator of the SNMP InformRequest with the error code of "genErr".

If the proxy cannot determine the Internet agent that initiated the SNMP Trap, then the CMIS M-EVENT-REPORT shall be sent as if it originated from the cmipsnmpProxyTable managed object class.

4 Common Procedures For CMISE Service Emulation

The procedures described in this section are used during CMISE service emulation defined in section 3. These procedures are collected together here for ease of specification, and to facilitate common implementation.

4.1 Verifying Existence Of An Object Instance

Since the proxy does not maintain a replicate copy of the MIB data maintained by the Internet agents, the existence of the a managed object, such as a based managed object specified in an incoming CMIS request, may need to be verified before further processing, such as scoping and filtering.

If the base object specified in the request does not exist in the Internet agent, then the proxy must send a "NoSuchObjectInstance" CMIS error response back to the ISO/CCITT manager.

If the base managed object is a pseudo object, the ISO/Internet proxy tries to determine if there exists a non-pseudo subordinate object. The base object exists if and only if there exists a non-pseudo subordinate object.

4.2 Translating Timestamps

This document does not specify a standard translation for the timestamp value in CMIS responses. However, the following paragraphs describe two potential implementations for this translation: ISO/Internet proxy's local time, and Internet agent's local time with fixed unknown delta.

4.2.1 ISO/Internet Proxy's Local Time

The timestamp value in the CMIS response can be set to the time provided by the ISO/Internet proxy's internal clock when the final SNMP Response is received to complete processing of a given CMIS request.

4.2.2 Internet Agent's Local Time

The ISO/Internet proxy can query the Internet agent for "sysUpTime", in addition to the original SNMP variable binding list in the first SNMP Request. Using this method, this value is recorded as the "agent's initial sysUpTime" and the ISO/Internet proxy's local time is recorded as "initial contact time".

Each CMIS request is then translated to one or more SNMP Requests by the ISO/Internet proxy to fulfill the CMIS request. If the last SNMP Request for the same CMIS request

is an SNMP Get Request, the "sysUpTime" is added into the SNMP variable binding list. Otherwise, an extra SNMP Get Request is issued with "sysUpTime" as the only element in the variable binding list. This new "sysUpTime" is called "agent's current sysUpTime".

The timestamp in the last CMIS response is then calculated as follows: initial contact time + (agent's current sysUpTime - agent's initial sysUpTime).

This approach eliminates the inaccuracy caused by network delay between the ISO/Internet proxy and Internet agent, and gives the ISO/CCITT manager a more accurate indication of when the operation was actually performed by the Internet agent (rather than the time that the response processed by the ISO/Internet proxy).

4.3 Derivation of SNMP Request Parameters

4.3.1 SNMPv2 Party and Context Parameters

The SNMPv2 source/destination party and context parameters shall be derived from the values in the privileged attribute certificate (PAC) passed in the access control parameter of the incoming ACSE or CMIS request.

4.3.2 SNMPv1 Community String Parameter

The SNMPv1 community string parameter shall be derived from the value in the privileged attribute certificate(PAC) passed in the access control parameter of the ACSE or CMIS request.

4.3.3 Internet Agent Transport Address

For SNMPv2, the proxy uses the value of the destination party identifier, derived according to the procedures in 4.3.1, to look up the transport address in an entry of the partyTable.

For SNMPv1, the Internet agent transport address shall be derived from the associated transport address in the table of cmipsnmpProxyAgent entries. The cmipsnmpProxyAgent is the one which has the same systemId as the attribute value within the RDN of the system object provided in the AETitle (if local name is used), or the CMIS managed object instance parameter.

4.3.4 SNMP Variable-Bindings Parameter

The SNMP variable-bindings list parameter contains a sequence of varBinds, each of which is an (Internet object name, value) pair.

For CMIS M-CREATE, M-SET, M-DELETE requests, the Internet object name shall be derived from the DN contained in the CMIP managed object instance parameter, and the attribute

identifier provided in the CMIS request attributeIdList or attributeList parameter, using the algorithm defined in [IIMCIMIBTRANS] clause 2.3.1.

For M-CREATE and M-SET requests, the Internet object value shall be assigned the attribute value associated with the attributeId from which the Internet object name was derived.

For M-GET requests, it is recommended the Internet object value is NULL.

For M-DELETE requests, the proxy shall use the delete information as described in the NAME BINDING template behavior defined for the object class. Within the BEHAVIOUR text, the CREATEDELETEATT specifies the Internet object name and CREATEDELETEVALUE specifies the Internet object value which signifies row deletion.

4.4 Derivation Of CMIS Parameters

Given the rules specified in this section, and knowledge of the IIMC containment hierarchy (name bindings), the ISO/CCITT {classOID}, {attributeOID}, and distinguished name can be derived from Internet names and the agent identifier.

The iimcAutoTrans OID is known to the proxy. It is defined in [IIMCIMIBTRANS].

4.4.1 Attribute Id Parameter

Using knowledge of the Internet name structure, and knowledge of valid <internetEntityId>(a) values known to the proxy, the <internetEntityId>(a) and <internet instanceId> may be extracted from the Internet object name.

The extraction process is not possible if the valid <internetEntityId>(a) value is not known to the proxy. In that case ,the translation process cannot be performed.

The ISO/CCITT attribute identifier is formed as:

{iimcAutoTrans <internetEntityId>(a)}

4.4.2 Managed Object Class Parameter

In CMIS response, the managed object class parameter can be derived from the proxy's retained information.

If actual class is used in the incoming CMIS request, the proxy must derive the object class parameter from the DN in the original CMIS request. The proxy shall compare the attribute value of the last RDN in the CMIS request with all the known ISO/CCITT object classes. The proxy shall assume that object class that has the longest match with the attribute value of the last RDN is the actual object class.

If the CMIS request is a scoped request, the object class shall be derived from the retained information.

4.4.3 Managed Object Instance Parameter

The managed object instance value (the base managed object's DN) is retained by the proxy during processing of the CMIS request. However, for DNs other than the base managed object instance, the following steps shall be taken to derive the subordinate RDNs.

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

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

ii) The internetClassId value, and the internetClass OID are used to form the final RDN for the object's DN. Assume that the object class was able to be determined using the procedures of 4.4.2. The sequence of other RDNs for the DN may be determined as follows.

Use knowledge of the containment hierarchy defined by name bindings, and the Internet agent's identifier. The object class's name binding may be identified as that name binding which contains the object class OID as its final component, in accordance with the name binding registration procedures defined in [IIMCIMIBTRANS] clause 2.1.3. Use the superior/subordinate relationships in the name bindings to build the DN in reverse, beginning with the final RDN and ending with the RDN for the ISO/CCITT system object. For superior classes that can have only a single instance, the internetClassId value for the object is created by appending the integer zero to the class OID. The agent's identifier is used as the value for the RDN of the ISO/CCITT system object.

One case exists for MIBs derived according to [IIMCIMIBTRANS] where it is possible for the superior object class to have multiple instances. This may occur when the subordinate object class was translated as the result of an SNMPv2 AUGMENTS clause and the superior object class is a table entry. In that case, the instance of the superior object class is identified by the same instanceId used to identify the subordinate object, prepended with the superior object's class OID.

If the Internet agent's address cannot be determined, then it may not be possible to associate a notification with a specific agent. This may be a problem if multiple Internet agents are associated with the same network address. In such cases, the DN for the cmipsnmpProxyTable object instance shall be used as the object instance.

4.4.4 EventId Parameter

The eventId parameter shall be the OID assigned the internetAlarm as defined in [IIMCIMIBTRANS].

4.4.5 InternetAlarm ProbableCause Parameter

The internetAlarm notification probableCause parameter shall be derived as defined in [IIMCIMIBTRANS] clause 3.2.5.

Internet traps/notifications are registered using the OID corresponding to the value of the Internet snmpTrapOID object defined in [SNMPv2MIB].

For SNMPv1 Trap PDUs, the snmpTrapOID is derived as stated in the SNMPv1/SNMPv2 Coexistence document [SNMPv2COEX] clause 4.1.2 (2). That definition is repeated below:

"... if the value of the generic-trap field is 'enterpriseSpecific' then the value used is the concatenation of the enterprise field from the trap PDU with additional sub-identifiers, '0', and the value of the specific-trap field."

For notifications defined according to the SNMPv2 SMI, the probableCause is determined by either the snmpTrapOID.0 or snmpEventID.i, which is contained in the second variable binding of the Trap or InformRequest, respectively.

Only the "globalValue" (i.e., OID form) of the probableCause syntax shall be used.

4.4.6 InternetAlarm Event Time Parameter

The event time parameter in the CMIS M-EVENT-REPORT should be calculated based on the timestamp field in the SNMP Trap. For SNMPv2, timestamp (sysUpTime) is provided in the first variable-bindings of the Trap or InformRequest. In order to convert the sysUpTime, which is the time ticks since the system was last re-initialized, to the CMIS event time, the proxy shall be made aware of every re-initialization of the Internet agents.

4.4.7 InternetAlarm AttributeIdentifier List

The following process shall be followed for each variable in the variable-bindings, excluding the first two variable-bindings.

The name portion of the variable binding shall be converted to an ISO/CCITT attributeId using the procedure specified in section 4.4.1. The converted ISO/CCITT attributeId shall be placed in the attributeIdList.

4.4.8 InternetAlarm ObjectInstanceList

The following process shall be followed for each variable in the variable-bindings, excluding the first two variable-bindings.

The name portion of the variable binding shall be converted to an ISO/CCITT object instance name using the procedures specified in section 4.4.3. The converted ISO/CCITT object instance name shall be placed in the object instance list.

If the proxy cannot determine the object instance name (e.g., because the Internet agent's identifier cannot be determined), then the objectInstanceList parameter shall not be included in the internetAlarm.

4.4.9 InternetAlarm InternetTrapInfo Parameter

The following process shall be followed for each variable in the variable-bindings.

The name portion of the variable binding shall be converted to an ISO/CCITT object instance name using the procedures specified in section 4.4.3. The converted ISO/CCITT object instance name shall be placed in the objectInstance field of the internetTrapInfo parameter.

If the Internet agent's identifier cannot be determined, or the <internetEntityId>(a) is unknown to the proxy, then the object instance name cannot be determined and the objectInstance field shall not be included in the internetTrapInfo parameter, but shall be included in the unknownVarBindList parameter.

4.4.10 InternetAlarm UnknownVarBindList Parameter

If the proxy cannot determine the attributeId for a variable binding (i.e., because the <internetEntityId>(a)portion of the Internet object name is not known to the proxy), then that variable binding shall be included in the unknownVarBindList parameter.

4.4.11 InternetAlarm PerceivedSeverity Parameter

The proxy cannot determine the perceivedSeverity for the translated SNMP Trap without specific knowledge of the Trap. Therefore, the proxy always assumes "indeterminate" for all the CMIS M-EVENT-REPORTs generated as a result of SNMP Traps.

However, a proxy can build in some local knowledge of the SNMP Traps and assign different perceivedSeverity values based on its local knowledge. Such local knowledge is not within the scope of this document.

4.4.12 InternetAlarm TransportDomain Parameter

For SNMPv2 Traps, the transportDomain parameter may be determined by using the one of the party identifier parameters associated with the Trap. The partyEntry object identified by the party identifier contains the partyDomain attribute.

For either SNMPv1, or SNMPv2 Traps, knowledge of the transport protocol used may be provided to the proxy. Alternatively, if the transport address can be determined, the proxy can determine the transport protocol from the format of the address. The proxy may then be able to determine the appropriate transportDomain value from local knowledge of the OIDs registered for different transport domains.

4.4.13 InternetAlarm TransportAddress Parameter

See section 4.4.12 for possible ways to determine the transport address.

4.4.14 InternetAlarm AccessControl Parameter

The access control parameter shall be assigned the community string or party identifiers associated with the SNMP Trap.

5 Error Message Translation

5.1 Translating SNMP Error Messages

SNMP error responses received by the ISO/Internet proxy are translated to CMIS error responses and sent back to the ISO/CCITT manager. The following sections provides a mapping for SNMP error messages to CMIS error responses.

5.1.1 tooBig

If the SNMP error "tooBig" is received, the ISO/Internet proxy should try to break the SNMP Request into smaller requests and resend the requests. If it is not feasible to break the request to any smaller request, and this error occurs, the CMIS error response "Complexity limitation" should be returned to the ISO/CCITT manager.

5.1.2 noSuchName

If the SNMP error "noSuchName" occurs when an attribute is queried as part of a CMIS Filter evaluation, then the filterItem will be evaluated as FALSE.

In order to check if an object exists, all the object class's attributes should be queried until at least one attribute's existence is verified. If none of the attributes exist, and the object is the base managed object, then a

"NoSuchObjectInstance" CMIS error response should be returned.

If the object exists and the SNMP "noSuchName" error occurs when attempting to read or modify an attribute, a CMIS "No Such Attribute" error response should be returned to the ISO/CCITT manager.

If the ISO/Internet proxy maintains correct schema information and the Internet agent is a conforming agent, an Internet object's attributes should either all exist or none exist. In order to make the ISO/Internet proxy a practical solution, the preceding error situation is included in order to deal with a non-conforming Internet agent.

5.1.3 badValue

If the SNMP error "badValue" is returned for an SNMP Get Request, then a "processing failure" CMIS error response should be returned to the ISO/CCITT manager. In the ProcessingFailure parameter of the CMIS error response, the errorId should be "snmpBadValue", and the errorInfo should be the variable binding identified by the error-index.

If the badValue error occurs during an SNMP Set Request to fulfill a CMIS M-DELETE request, a "processing failure" CMIS error response should be returned. In the ProcessingFailure parameter, the errorId should be " cannotDelete" and the errorInfo should be the variable binding that is identified by the error-index.

5.1.4 readOnly

The proxy should never receive an SNMP readOnly error from an SNMPv1 agent. If this error is received, a "processing failure" CMIS error response should be returned to the ISO/CCITT manager. In the processingFailure parameter, the errorId should be "snmpReadOnly" and the errorInfo should be the variable binding that is identified by the error-index.

For SNMPv2, if the SNMP error "readOnly" occurs when checking for the existence of a base object, a "processingfailure" CMIS error response should be returned to the ISO/CCITT manager. In the ProcessingFailure parameter of the CMIS error response, the errorId should be "snmpReadOnly", and the errorInfo should be the variable binding identified by the error-index. If the error occurs when deleting the object, then the deleteErrorInfo field in the response shall be set to "accessDenied".

5.1.5 genErr

If the SNMP error "genErr" occurs, the "ProcessingFailure" CMIS error response should be returned to ISO/CCITT manager. If the entry exists, scoping continues; otherwise, scoping

terminates. In the ProcessingFailure parameter of the CMIS error response, the errorId should be "snmpGenErr".

There are additional error messages in SNMPv2. Most of the errors are defined for the Set Request. Since a Set Request may be originated when processing a CMIP M-SET request, an M-CREATE request or an M-DELETE request, the proxy must ensure each error code is translated to the one which is most compatible with the original CMIS request. In addition, the proxy must ensure the selected error value is compatible with the use of other parameters such as scoping, filtering, synchronization and multiple linked reply.

5.1.6 noAccess

This error indicates the variable binding's name specifies a variable which is not accessible by an SNMP Set Request. This error should be mapped to the CMIS "accessDenied" error.

5.1.7 wrongType

This error indicates the variable binding's value field of an SNMP Set Request specified a type which is inconsistent with that required for the variable. This error may be mapped to the CMIS "invalidAttributeValue" error.

5.1.8 wrongLength

This error indicates the variable binding's value field of an SNMP Set Request specifies, according to the ASN.1 language, a length which is inconsistent with that required for the variable. If the original CMIS request is M-CREATE or M-SET, the CMIS error "InvalidAttributeValue" shall be returned. If the original CMIS request is M-DELETE, the CMIS "processing failure" error shall be returned.

5.1.9 wrongEncoding

This error is used to indicate the variable binding's value field of an SNMP Set Request contains an ASN.1 encoding which is inconsistent with that field's ASN.1 tag. This error should be mapped to the CMIS "processingFailure" error.

5.1.10 wrongValue

This error indicates the variable binding's value field in an SNMP Set Request specifies a value which could under no circumstances be assigned to the variable. This error should be mapped to the CMIS "invalidAttributeValue" error.

5.1.11 noCreation

This error is generated when an SNMP Set Request variable binding name specified a variable which does not exist and

could not ever be created. This error should be mapped to the CMIS "invalidObjectInstance" error.

5.1.12 inconsistentValue

This error indicates that an SNMP Set Request variable binding value field specified a value that could under other circumstances be assigned to the variable, but is presently inconsistent. If the SNMP Set Request was generated as a result of a CMIS M-CREATE or M-SET operation, the error should be mapped to the CMIS "invalidAttributeValue" error.

If the SNMP Set Request was generated as a result of CMIS M-DELETE operation, the error may be mapped to the CMIS "processingfailure" error.

5.1.13 resourceUnavailable

This error indicates that the assignment of a value by an SNMP Set Request requires the allocation of a resource which is presently unavailable. This error may be mapped to the CMIS "resourceLimitation" error.

5.1.14 commitFailed

When performing an SNMP Set Request, the Internet agent must ensure all variable assignments occur atomically. If any of the assignments fail, an SNMP "commitFailed" error is returned. If the original CMIS request is a "best effort" request, the proxy should either retry the failed variable assignments by sending multiple SNMP Set Requests, or return a CMIS setListError with a "processingfailure" error.

5.1.15 undoFailed

When performing an SNMP Set Request, the Internet agent must ensure all variable assignments occur atomically. If any of the assignments fail, the agent should undo all the assignments. An SNMP "undoFailed" error is returned when the agent cannot undo all the assignments. CMIS does not have any error value equivalent to this. The CMIS "processing failure" error must be returned.

5.1.16 authorizationError

This error indicates that an SNMP Request has been discarded because the authorization context used in the request does not allow the PDU type. This error is mapped to the CMIS "accessDenied" error.

5.1.17 notWritable

The "notWritable" error is used to indicate that an SNMP Set Request is trying to modify the value of a variable which is

not modifiable, no matter what new value is specified. This error shall be mapped to the CMIS "invalidOperation" error.

5.2 CMIS Processing Failure

There are many error scenarios in which the error cannot be mapped to a specific CMIS error. In this case, the "processing failure" CMIS error response should be reported back to the ISO/CCITT manager. In order to provide the ISO/CCITT manager with a better description of the error, the specificErrorInfo field in ProcessingFailure is used to record the cause of the problem.

The following object identifiers are defined:

```
errors          OBJECT IDENTIFIER :: { iimcProxyMisc 4 }

snmpTooBig      OBJECT IDENTIFIER :: { errors 0 }
snmpBadValue    OBJECT IDENTIFIER :: { errors 1 }
snmpReadOnly    OBJECT IDENTIFIER :: { errors 2 }
snmpGenErr      OBJECT IDENTIFIER :: { errors 3 }
noResponse      OBJECT IDENTIFIER :: { errors 4 }
 cannotDelete   OBJECT IDENTIFIER :: { errors 5 }
notImplemented  OBJECT IDENTIFIER :: { errors 6 }
wrongLength     OBJECT IDENTIFIER :: { errors 7 }
wrongEncoding   OBJECT IDENTIFIER :: { errors 8 }
```

where the errInfo parameter depends on the value of errorId:

```
errorId         errInfo
-------         -------
snmpTooBig      VarBindList
snmpBadValue    VarBind
snmpReadOnly    OBJECT IDENTIFIER
cannotDelete    VarBind
notImplemented  INTEGER {
                filter(0),
                scope(1),
                transport(2),
                authenticationProtocol(6),
                privacyProtocol(7)
                }
```

6 ISO/CCITT Systems Management Functions

ISO/CCITT systems management standards include a set of Systems Management Function specifications. An ISO/Internet proxy may choose to support some or all of these systems management functions. This section provides some of the modeling approaches which may be used in supporting ISO/CCITT systems management functions.

6.1 Object Management Function

The ISO/CCITT Object Management Function [ISO10164-1] defines a set of pass-through services and a set of notification services. Since the pass-through services are mapped directly to the CMIS services, the service emulation for pass-through services is the same as the emulation of corresponding CMIS services.

The notification services in the Object Management Function are the object creation reporting service, the object deletion reporting service, and the attribute value change reporting service. This proxy does not specify any additional service emulation for these notifications, beyond that already specified for CMIS M-EVENT-REPORT.

6.2 State management function

The ISO/CCITT State Management Function [ISO10164-2] specifies a set of state attributes which may be imported by a managed object class to represent the status of the underlying managed resources. SNMP also defines a set of management states, such as linkUp and linkDown for an interface, and the Row Status for a conceptual row. Automatic translation of SNMP states to ISO/CCITT equivalent states by this ISO/Internet proxy is not specified. However, an implementation may manually map some or all of the SNMP states to ISO/CCITT states, if so desired.

The State Management Function also defines a state change notification. This notification may be imported by any managed object class to report state attribute changes. This proxy does not specify any additional service emulation for this notification, beyond that already specified for CMIS M-EVENT-REPORT.

6.3 Attributes For Representing Relationships

The ISO/CCITT Attributes for Representing Relationships [ISO10164-3] defines a set of relationship attributes which maybe imported by any managed object to represent relationships among managed objects. Relationship variables in SNMP MIBs are not automatically mapped to ISO/CCITT relationship attributes during the MIB translation specified in [IIMCIMIBTRANS]. However, an implementation may manually map SNMP relationships to ISO/CCITT relationships when applicable and so desired.

[ISO10164-3] also defines a relationship change notification which may be imported by a managed object class to report any relationship attribute changes. This proxy does not specify any additional service emulation for this notification, beyond that already specified for CMIS M-EVENT-REPORT.

6.4 Alarm Reporting Function

The ISO/CCITT Alarm Reporting Function [ISO10164-4] defines a set of alarm notifications which may be supported by managed object classes to report detected possible faults. These notifications are not automatically generated by [IIMCIMIBTRANS] during MIB translation. Instead, Internet Traps and Inform Requests are mapped to a special-purpose internetAlarm as defined by this document.

6.5 Event Report Management Function

The ISO/CCITT Event Report Management Function [ISO10164-5] defines an Event Forwarding Discriminator (EFD) managed object which allows an ISO/CCITT manager to control the forwarding and processing of potential event reports by an ISO/CCITT agent. The Event Report Management Function maybe supported by an ISO/Internet proxy to allow the ISO/CCITT manager to control where and how Internet Traps and Inform Requests may be forwarded.

Since all Internet Traps and Inform Requests are translated by the proxy and are forwarded to their destinations by the proxy, EFD managed objects are best supported by the proxy as local objects. Upon receiving a CMIS M-CREATE request for an EFD, the proxy creates the EFD object instance according to the specified name binding. Once created, the EFD is used by the proxy to determine which CMIS M-EVENT-REPORTs are to be forwarded to a particular destination during a specified time period.

Since a proxy may provide proxy services to multiple Internet agents, each created EFD managed object shall be associated with a respective Internet agent, so that only the traps or inform requests originating from the associated Internet agent are processed according to the criteria specified in the EFD. The binding of an EFD and the system object representing the Internet agent is represented by the name binding of the EFD. Since EFD objects are treated as local objects of the proxy, any CMIS control and monitoring requests on the EFD managed objects are handled at the proxy.

Editor's Note: [The above discussion requires review, especially as regards mapping EFDs to individual Internet agents.]

6.6 Log Control Function

The ISO/CCITT Log Control Function [ISO10164-6] defines a Log managed object which allows control and monitoring of a log and the retrieval of its log records. If the Log managed object is support, Internet Traps and Inform Requests may be logged according to a predefined criteria.

Since only notifications are logged and these are constructed by the proxy, the Log managed object can be defined as a local object of a proxy. Upon receiving a CMIS M-CREATE request for a Log object, the proxy creates the Log instance according to the specified name binding. Once created, the Log is used by the proxy to process the received Traps and Inform Requests (and local notifications) for logging.

Since a proxy may provide proxy services to multiple Internet agents, each created Log managed object shall be associated with its respective Internet agent, so that only the Internet Traps or Inform Requests originating from or to the associated Internet agent are processed by the Log. The binding of a Log and its agent is reflected in the name binding of the Log object. Since log objects are treated as local objects in the proxy, any CMIS control and monitoring requests on the Log managed objects are terminated at the proxy.

6.7 Security Alarm Reporting Function

Support of the ISO/CCITT Security Alarm Reporting Function [ISO10164-7] is similar to support of the Alarm Reporting Function [ISO10164-4], as described in section 6.5.

6.8 Security Audit Trail Function

Support of the ISO/CCITT Security Audit Trail Reporting Function [ISO10164-8] is similar to support of the Alarm Reporting Function [ISO10164-4], as described in section 6.5.

7 ISO/CCITT-Internet Proxy MIB

The ISO/CCITT-Internet Proxy MIB defines a set of objects for specifying the information that is needed for both community-based and party-based SNMP management on a per Internet agent basis.

The containment hierarchy and other introductory information regarding the IIMC Proxy MIB can be found in section 2.3.

ISO/CCITT-Internet proxy MIB object classes and attributes are assigned OIDs under the {iimcManagementProxy} arc, as allocated in [IIMCIMIBTRANS].

7.1 Proxy MIB Managed Object Class Definitions

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

DEFINED AS
!This managed object class is used to represent an Internet agent in the proxy MIB. Each agent that the proxy manages is represented by an instance of this object class.

The cmipsnmpProxyAgentId attribute contains the administratively-assigned name of the managed system that contains the Internet agent. Usually this is an Internet Domain Name. This attribute value shall be determined by the manager when the object is created.

The managementProtocol attribute specifies the Internet management protocol used by the proxy to manage devices. It shall be the OID indicating SNMPv1, SNMPv2, or some other protocol. This attribute is assigned a value (an OID) by the manager that is appropriate for the Internet agent.

The supportedMIBs attribute identifies the set of GDMO documents that describe the MIBs that the Internet agent supports. The ISO/CCITT manager may add elements to or remove elements from this attribute.

Two object instances shall be created by the proxy automatically when an instance of the cmipsnmpProxyAgent class is created. One is the logical system object that represents the Internet agent. The other is the "OP1 Library Vol. 4":capabilityObject as defined by [OP1LIBV4].

The following text describes one possible implementation of gathering information defined in the Capability object's supportedObjectClassList. When the cmipsnmpProxyAgent is created, or when the supportedObjectClassList attribute changes, the proxy shall find out all the object classes defined in all the GDMO documents described in the supportedMIBs attribute. The proxy then forms an SNMP Get Next Request with all the object classes (translated to the OID used by the Internet agent) in a variable binding list to find out whether a particular object class is supported by the Internet agent. The proxy then fills the supportedNameBindingList by finding out all the name bindings used by the object classes in the supportedObjectClassList.

Editor's Note: [Reviewers, the above proposal responds to Action #5 assigned at the Red Bank meeting. Please comment on this proposal.]

The accessControlEnforcement attribute indicates where access control is applied: at the Internet agent, the ISO/Internet proxy, or both.

The accessControlMechanism attribute indicates whether no access control, Internet access control as specified in [SNMPv2SEC], or ISO/CCITT access control as specified in [ISO10164-9] is to be used. The default is no access control.

The opState attribute indicates the perceived state of the Internet agent. It is the same as the operationalState attribute defined in [ISO10165-2]. It is reregistered here to facilitate the application of Internet Access control mechanisms.

The "enabled" state means that the Internet agent is operational, as perceived by the proxy, i.e., it can be reached.

The "disabled" state means that the Internet agent is not operational, as perceived by the proxy, i.e., it cannot be reached.

The validity of the opState attribute is dependent on the mechanisms used by the proxy to determine reachability, and the frequency with which it is invoked. For connectionless environments (e.g., UDP), polling will have to be performed by the proxy. For connection oriented environments (e.g., TCP), loss of connectivity as indicated by lack of "keep alive" messages can be used to provide this information.

The adminState attribute is used to suspend and resume the proxy activity relative to the Internet agent. It is the same as the administrativeState attribute defined in [ISO10165-2]. It is reregistered here to facilitate the application of Internet Access control mechanisms.

The "unlocked" state means that proxy must continue to perform, or resume performing, proxy activities on behalf of the Internet agent.

The "locked" state means that the proxy must not perform, or suspend performing, proxy activities on behalf of the Internet agent.

Editor's Note: [Reviewers, the above proposal responds to Action #4 assigned at the Red Bank meeting. Please review and comment. Also, please consider if attributes should be defined to control a polling interval and maximum number of retries.]
!;;;

```
     ATTRIBUTES
          systemTitle                GET,
          cmipsnmpProxyAgentId       GET,
          transportAddress                GET-REPLACE,
          managementProtocol              GET
                                          REPLACE-WITH-DEFAULT,
          supportedMIs0              GET-REPLACE ADD-REMOVE,
          accessControlEnforcement        GET-REPLACE,
          accessControlMechanism          GET-REPLACE
                                     DEFAULT VALUE
               IimcProxyASN1.c_noAccessControl;
          adminState                      GET-REPLACE,
          opState                         GET;
     NOTIFICATIONS
          {iimcManagementDocMan 1}:internetAlarm;;;;
REGISTERED AS  { iimcManagementProxy 1 1};


cmipsnmpProxyTable MANAGED OBJECT CLASS
     DERIVED FROM
          "Rec. X.721 | ISO/IEC 10165-2 : 1992" :top;
     CHARACTERIZED BY
          cmipsnmpProxyTablePkg PACKAGE
          BEHAVIOUR
          cmipsnmpProxyTablePkgBehaviour BEHAVIOUR
          DEFINED AS
          !This managed object class is used to contain
          objects that represent an Internet agent in the
          proxy MIB.

          The internetAlarm shall be emitted by this object
          class when the application level source of the SNMP
          Trap or Inform Request cannot be determined. The
          address field of the internetAlarm shall be set to
          the network address associated with the SNMP Trap
          or Inform Request.!;;
     ATTRIBUTES
          {iimcManagementDocMan 1}:internetClassId GET,
     NOTIFICATIONS
          {iimcManagementDocMan 1}:internetAlarm;;;
REGISTERED AS  {iimcManagementProxy 1};
```

7.2 Proxy MIB Attribute Definitions

```
accessControlEnforcement ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.AccessControlEnforcement;
     MATCHES FOR EQUALITY, ORDERING;
     BEHAVIOUR accessControlEnforcementBehaviour BEHAVIOUR
          DEFINED AS
          !The accessControlEnforcement attribute indicates
          where access control is applied: Internet agent,
          proxy, or both.!;;
REGISTERED AS {iimcManagementProxy 1 1 1};


accessControlMechanism ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.AccessControlMechanism;
     MATCHES FOR EQUALITY, ORDERING;
     BEHAVIOUR accessControlMechanismBehaviour BEHAVIOUR
     DEFINED AS
          !The accessControlMechanism attribute indicates
          which access control is to be applied at the proxy
          device.  The mechanism may be no access control,
          the internet access control as defined in
          [SNMPv2SEC] or one of the ISO access control
          mechanisms as defined in [ISO10164-9].!;;
REGISTERED AS {iimcManagementProxy 1 1 2};


adminState ATTRIBUTE
     DERIVED FROM
          "Rec. X.721 | ISO/IEC 10165-2 : 1992":
          administrativeState;
     BEHAVIOUR adminStateBehaviour BEHAVIOUR
     DEFINED AS
          !This attributes is the same as the
          administrativeState registered by ISO in  [10165-
          2]. It is reregistered here to allow for access
          control via Internet mechanisms.!;;
REGISTERED AS {iimcManagementProxy 1 1 3};


cmipsnmpProxyAgentId ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.CmipsnmpProxyAgentId;
     MATCHES FOR EQUALITY, SUBSTRINGS;
     BEHAVIOUR cmipsnmpProxyAgentIdBehaviour BEHAVIOUR
     DEFINED AS
          !This is the naming attribute for the
          cmipsnmpProxyAgent object class. It contains
          the Internet Domain Name of the managed system
          that contains the SNMPv1/SNMPv2 agent.  The value
          is at the time the object is created.!;;
REGISTERED AS {iimcManagementProxy 1 1 4};
```

```
managementProtocol ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.ObjectIdentifier;
     MATCHES FOR EQUALITY, ORDERING;
     BEHAVIOUR managementProtocolBehaviour BEHAVIOUR
     DEFINED AS
          !This attributes specifies the Internet management
          protocol used for proxy to managed devices.  It
          shall have a value of either SNMPv1 or SNMPv2.!;;
REGISTERED AS {iimcManagementProxy 1 1 9};


opState ATTRIBUTE
     DERIVED FROM
          "Rec. X.721 | ISO/IEC 10165-2 : 1992":
          operationalState;
     BEHAVIOUR opStateBehaviour BEHAVIOUR
     DEFINED AS
          !This attributes is the same as the
          operationalState registered by ISO in [10165-2].
          It is reregistered here to allow for access control
          via Internet mechanisms.!;;
REGISTERED AS {iimcManagementProxy 1 1 10};


supportedMIBs ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.SupportedMIBs;
     MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
     BEHAVIOUR supportedMIBsBehaviour BEHAVIOUR
     DEFINED AS
          !This attributes specifies the set of Internet OIDs
          of registered MIBs that the agent supports.!;;
REGISTERED AS {iimcManagementProxy 1 1 11};


transportAddress ATTRIBUTE
     WITH ATTRIBUTE SYNTAX
          IimcProxyASN1.OctetString;
     MATCHES FOR EQUALITY, SUBSTRINGS;
     BEHAVIOUR transportAddressBehaviour BEHAVIOUR
     DEFINED AS
          !This attributes specifies the transport address of
          the Internet agent.  The format shall be as defined
          by relevant internet documents.!;;
REGISTERED AS {iimcManagementProxy 1 1 12};
```

7.3 Proxy MIB Name Bindings

ISO/CCITT-Internet proxy name bindings are registered under the {iimcProxyNB} arc, which is the {iimcManagementProxy 2} arc specified by [IIMCIMIBTRANS]. The cmipsnmpProxyTable

object is bound to the ISO system managed object particular to the proxy system.

```
cmipsnmpProxyAgent-cmipsnmpProxyTableNB  NAME BINDING
     SUBORDINATE OBJECT CLASS cmipsnmpProxyAgent
          AND SUBCLASSES;
     NAMED BY SUPERIOR OBJECT CLASS cmipsnmpProxyTable
          AND SUBCLASSES;
     WITH ATTRIBUTE cmipsnmpProxyAgentId;
     BEHAVIOUR
     cmipsnmpProxyAgent-cmipsnmpProxyTableNBBehaviour
     BEHAVIOUR DEFINED AS
          !This managed object may be created and deleted
          either by management action, or by local action, of
          the proxy.

          A side effect of the creation of this object shall
          be the creation of the ISO system managed object
          associated with the Internet agent.  The
          systemTitle attribute of the ISO system object thus
          created shall be assigned the value of the
          cmipsnmpProxyAgentId
          attribute.!;;
     CREATE;
     DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS {iimcManagementProxyNB 2};


cmipsnmpProxyTable-systemNB  NAME BINDING
     SUBORDINATE OBJECT CLASS cmipsnmpProxyTable
          AND SUBCLASSES;
     NAMED BY SUPERIOR OBJECT CLASS
          "Rec. X.721 | ISO/IEC 10165-2 :1992": system;
     WITH ATTRIBUTE {iimcManagementDocMan 1}:internetClassId;
REGISTERED AS {iimcManagementProxyNB 1};
```

7.4 Proxy MIB ASN.1 Modules

```
IimcProxyASN1 {iimcManagementModMan 3}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS
     iimcManagementProxy, iimcManagementMOC,
     iimcManagementAtt
          FROM IimcAssignedOIDs {iimcManagementModMan 1}
     ObjectIdentifier
          FROM IimcCommonDef (iimcManagementModMan 2};

     iimcManagementProxyNB OBJECT IDENTIFIER
          ::={iimcManagementProxy 1}

     -- for ISO/CCITT-Internet proxy name bindings
     iimcProxyMisc OBJECT IDENTIFIER
          ::={iimcManagementProxy 2}
```

```
     -- for miscellaneous assignments
     CmipsnmpProxyAgentId     ::= GraphicString

     AccessControlUsed        ::= INTEGER {
                          noAccessControl (0),
                               internet (1),
                               so (2)}

     AccessControlEnforcement ::= INTEGER {
                               agent   (1),
                               proxy   (2),
                               both    (3)}

     SupportedMIBs            ::= SET OF CHOICE {
                               oid OBJECT IDENTIFIER,
                               ps  PrintableString
                               }

     c_noAccessControl INTEGER ::= 0

     snmpv1 OBJECT IDENTIFIER ::= {iimcProxyMisc 1}
        -- the OID for SNMPv1

     snmpv2 OBJECT IDENTIFIER ::= {iimcProxyMisc 2}
        -- the OID for SNMPv1

     snmpv1CommString OBJECT IDENTIFIER ::= {iimcProxyMisc 3}
        -- the OID for SNMPv1 community string authentication


     errors         OBJECT IDENTIFIER :: {iimcProxyMisc 4 }

     snmpTooBig     OBJECT IDENTIFIER :: { errors 0 }
     snmpBadValue   OBJECT IDENTIFIER :: { errors 1 }
     snmpReadOnly   OBJECT IDENTIFIER :: { errors 2 }
     snmpGenErr     OBJECT IDENTIFIER :: { errors 3 }
     noResponse     OBJECT IDENTIFIER :: { errors 4 }
     cannotDelete   OBJECT IDENTIFIER :: { errors 5 }
     notImplemented OBJECT IDENTIFIER :: { errors 6 }
     wrongLength    OBJECT IDENTIFIER :: { errors 7 }
     wrongEncoding  OBJECT IDENTIFIER :: { errors 8 }

     ErrInfo ::= INTEGER {
               filter(0),
               scope(1),
               transport(2),
               authenticationProtocol(6),
               privacyProtocol(7)
          }

END
```

7.5 Party MIB MOCS

This section provides managed object conformance statements (MOCS) for the IIMC Party MIB.

Editor's Note: [This section to be provided once the Party MIB definition becomes stable.]

8 Conformance Requirements

8.1 Management Communication Requirements

An implementation of the ISO/Internet proxy shall satisfy the following management communication requirements.

- The ISO/Internet proxy shall conform to ISO/CCITT CMIP in the agent role, as specified by [ISO9596-1] and [ISO9595], and profiled by either [AOM11] or [AOM12].

- The ISO/Internet proxy shall conform to either Internet SNMPv1 or SNMPv2 in the manager role, as specified by [RFC1157] or [SNMPv2PROT].

- The ISO/Internet proxy shall conform to all mandatory security requirements specified by [IIMCSEC].

Editor's Note: [See 8.4 for discussion of alternative proposals.]

8.2 Management Function Requirements

An implementation of the ISO/Internet proxy shall satisfy the following management function requirements.

- The ISO/Internet proxy may optionally conform to ISO/CCITT system management functions in the agent role, as specified by [ISO10164-1,2,3,4,5,6,7,8], and profiled by AOM211, AOM221, or AOM231.

Editor's Note: [Should the above be handled differently? Should AOM221 be called out separately to encourage support for this profile and the EFD? What is the purpose of calling out conformance to functions not used by the proxy (such as 10164-3,7,8)?]

8.3 Management Information Requirements

An implementation of the ISO/Internet proxy shall satisfy the following management information requirements.

- The ISO/Internet proxy shall support the IIMC Proxy MIB definitions contained in section 7 of this document, in the agent role.

- The ISO/Internet proxy shall support the IIMC Party MIB definitions contained in [IIMCSEC], in the agent role. The ISO/Internet proxy shall also support the Internet Party MIB definitions contained in [SNMPv2PARTY], in the manager role.

- The ISO/Internet proxy shall support one or more translated MIBs which have been derived in accordance with the procedures specified in [IIMCIMIBTRANS]. For each supported MIB, the ISO/CCITT GDMO translation shall be supported in the agent role, and the Internet MIB shall be supported in the manager role.

- The ISO/Internet proxy shall comply with the information models specified by [ISO10165-1,4] and either [RFC1155],[RFC1212], or [SNMPv2SMI].

Editor's Note: [There should probably be a requirement to support the ISO System object. Should there be a requirement to support the Internet MIB-II? Should there be any optional requirement to support OMNIPoint Capability and/or Discovery objects?]

8.4 Service Emulation Requirements

An ISO/Internet proxy implementation that claims conformance to this specification must state the level of CMIS service emulation that it supports. Two levels are defined:

a) a basic proxy which emulates CMIS kernel services, without any support for scoping and filtering; and

b) an enhanced proxy which emulates all CMIS services, including support for scoping and filtering on all applicable CMIS services.

The basic proxy emulates CMIS kernel services as supported by the Basic Management Communications Profile [AOM11]. At this level, the proxy is not required to support CMIS multiple object selection, filtering, or linked reply functional units.

The enhanced proxy requires support for additional CMIS functional units as supported by the Enhanced Management Communications Profile [AOM12]. At this level, the proxy is required to support CMIS multiple object selection, filtering, and linked reply functional units.

Editor's Note: [The above levels of conformance represent those possible using the existing AOM1n profiles and CMIP functional unit negotiation mechanisms. However, there has been considerable discussion of this issue, and two alternative definitions have also been proposed for the basic level (a):

1) a basic proxy that requires support for the kernel only, plus support for scoped CMIS Get; or

2) a basic proxy that requires support for the kernel only, plus support for the OMNIPoint 1 Discovery managed object which allows retrieval of all managed object class and instance names.

These proposals argue that AOM11 alone is insufficient, because it requires the ISO/CCITT manager to know a priori all of the managed objects existing at the Internet agent. This has been generally agreed to be a problem during IIMC discussion. The issue is therefore how to provide additional functionality without substantial increase in implementation complexity.

Alternative 1) allows support for CMIS scoping, but restricts its use to the CMIS Get service. This usage restriction simplifies proxy implementation considerably, but cannot be legitimately expressed using existing AOM1n profiles and CMIP/SMASE functional unit mechanisms (i.e., there is no known way to negotiate support for CMIS multiple object selection or linked reply, but restrict its usage to specific CMIS services). This alternative must be further explored, considering various mechanisms (e.g., SMASE functional units, application contexts).

Alternative 2) allows very limited dynamic discovery capabilities through support of a special-purpose managed object known as the OMNIPoint 1 Discovery object. This object allows a manager to retrieve the agent's containment tree (class and instance information) through use of a normal (non-scoped) CMIS Action. Again, this greatly simplifies proxy implementation by eliminating support requirements for CMIS multiple object selection or linked reply, but requires processing similar to multiple object selection when emulating the Discovery object's Action. However, the Discovery object, while internationally-endorsed as part of OMNIPoint 1, is a pre-standard intercept which will eventually be replaced by ISO/CCITT management knowledge standards.

Finally, consideration has also been given to simply requiring full support of AOM12. This solution would impose much greater implementation complexity than either of the above alternatives.

COMMENTS ON THESE PROPOSALS ARE SOLICITED DURING REVIEW. IN PARTICULAR, COMMENTS REGARDING MECHANISMS FOR ALTERNATIVE 1 AND THE ACCEPTABILITY OF ALTERNATIVE 2 ARE INVITED.]

9 Abbreviations

CMIP:	Common Management Information Protocol
CMIS:	Common Management Information Service
CMISE:	Common Management Information Service Element
DN:	Distinguished Name
MIB:	Management Information Base
MOC:	Managed Object Class (CMIP)
MOI:	Managed Object Instance (CMIP)
MIT:	Management Information Tree (naming tree)
OID:	ASN.1 Object Identifier
PDU:	Protocol Data Unit
RDN	Relative Distinguished Name
SNMP:	Simple Network Management Information Protocol
SNMPv1:	Simple Network Management Information Protocol version 1 [RFC1157]
SNMPv2:	Simple Network Management Information Protocol version 2 [SNMPv2PROT]

10 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
Lee LaBarre	- The MITRE Corporation
David Liu	- Northern Telecom
Owen Newnan	- U S West
Steve Ng	- MPR Teltech
Yasuhiro Ohara	- NTT
George Pavlou	- University College of London
Lisa Phifer	- Bellcore
Tom Rutt	- AT&T
Mark Smith	- Hewlett-Packard
Einar Stefferud	- Network Management Associates
Huy Truong	- Tandem
Dean Voiss	- NetLabs
David Waitzman	- BBN
Graham Wisdom	- Timeplex
Yoshi Yamashita	- NKK Corporation

Appendix A Example Operation

```
Operation: CMIS M-GET request
Object class: Internet MIB-II ip
Object instance:{systemTitle = "NetLabs" }
                {internetClassId = ip }
Scope: 2nd level down
Filter: ipRouteType == indirect
Attribute id list: ipRouteDest
```

Example Internet MIB-II ipRouteEntries:

ipRouteDest	ipRouteType	Other object types
192.95.93.1	direct	
192.95.93.2	indirect	
192.95.93.3	invalid	
192.95.93.4	other	
192.95.93.5	indirect	

(end of the table)

The following sections show an example of how the ISO/Internet proxy fulfills the above CMIS M-GET request based on the example Internet MIB-II ipRouteEntries.

1. Check the existence of the base object

The ISO/Internet proxy issues an SNMP GetNext Request.

```
GetNextRequest { ip }
GetResponse { ipForwarding = 1 }
```

If the get is successful, the ISO/Internet proxy would have verified that the base object exists.

2. Managed object selection

Based on the name binding definition, the following object classes are found in the "object class group":

a) ipAddrEntry
b) ipRouteEntry
c) ipNetToMediaEntry

For each object in the "object class group", the object instances may be retrieved via SNMP GetNext Requests.

a) ipAddrEntry: The definition for this object class does not contain the attribute specified in the filter (ipRouteType), therefore the ISO/Internet proxy concludes that there are no object instances with ipAddrEntryobject class in the scope.

b) ipRouteEntry: The definition for this object class contains the attribute specified in the filter (ipRouteType),therefore the ISO/Internet proxy issues

SNMP GetNext Requests to retrieve the object instances. The SNMP requests issued and the responses received would be as follows:

```
GetNextRequest  {ipRouteDest, ipRouteType}
GetResponse     {ipRouteDest.192.94.93.1 = 192.94.93.1,
                 ipRouteType.192.94.93.1 = direct}

GetNextRequest  {ipRouteDest.192.94.93.1,
                ipRouteType.192.94.93.1}
GetResponse     {ipRouteDest.192.94.93.2 = 192.94.93.2,
                 ipRouteType.192.94.93.2 = indirect}

GetNextRequest  {ipRouteDest.192.94.93.2,
                 ipRouteType.192.94.93.2}
GetResponse     {ipRouteDest.192.94.93.3 = 192.94.93.3,
                 ipRouteType.192.94.93.3 = invalid}

GetNextRequest  {ipRouteDest.192.94.93.3,
                 ipRouteType.192.94.93.3}
GetResponse     {ipRouteDest.192.94.93.4 = 192.94.93.4,
                 ipRouteType.192.94.93.4 = other}

GetNextRequest  {ipRouteDest.192.94.93.4,
                ipRouteType.192.94.93.4}
GetResponse     {ipRouteDest.192.94.93.5 = 192.94.93.5,
                 ipRouteType.192.94.93.5 = indirect}

GetNextRequest  {ipRouteDest.192.94.93.5,
                ipRouteType.192.94.93.5}
GetResponse     {ipRouteIfIndex = 5,
                 ipRouteProto = 1}
```

c) ipNetToMediaEntry: The definition for this object class does not contain the attribute specified in the filter (ipRouteType), therefore the ISO/Internet proxy concludes that there are no object instances with ipNetToMediaEntry object class in the scope.

There are a set of five object instances in the scope. After the filter is applied to these five object instances, there are only two object instances left on which the CMIS M-GET operation is performed.

3. Form the response

The following CMIS responses should be formed and sent back to the ISO/CCITT manager

```
CMIS M-GET Response (first linked reply PDU):
     Object class: ipRouteEntry
     Object instance:{systemTitle = "NetLabs" }
          {internetClassId = ip }
          {internetClassId = ipRouteTable }
          {internetClassId = ipRouteEntry.192.94.93.2 }
```

```
          Attribute list: ipRouteDest = 192.4.93.2

     CMIS M-GET Response (second linked reply PDU):
          Object class: ipRouteEntry
          Object instance:{cmipsnmpProxyId = "NetLabs" }
               {internetClassId = ip }
               {internetClassId = ipRouteTable }
               {internetClassId = ipRouteEntry.192.94.93.5 }
          Attribute list: ipRouteDest = 192.4.93.5

     CMIS M-GET Response (final empty response PDU)
```

Appendix B Translated MIB Naming Proposals

There are at least three alternative proposals which have been put forward to resolve the naming issue. These are outlined below. There are many other possible proposals.

Editor's Note: [PLEASE COMMENT ON THESE PROPOSALS DURING REVIEW.]

A) As described in the current document (Draft 1)

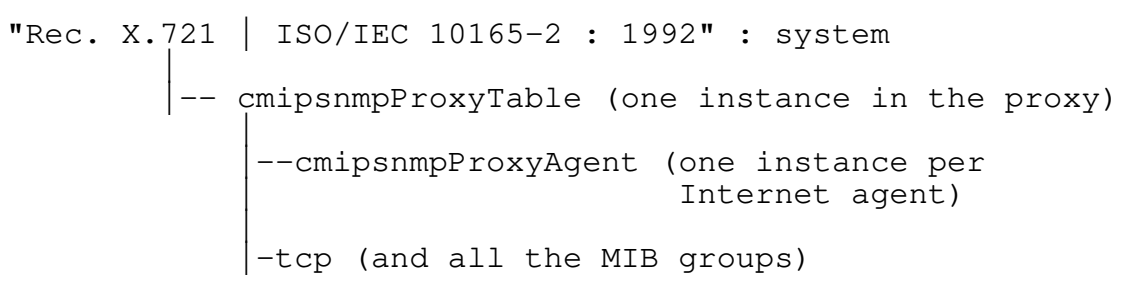
Pro's:
From the perspective of the ISO manager, all agents are alike in their containment hierarchy; whether via proxy or native CMIP implementation.

Con's:
Scoping across all the internet agents is difficult.

To use local name to access the MIBs implemented in the internet agents, the ISO manager needs to establish a CMIP association per internet agent.

If EFDs and Logs are instantiated in the proxy applies to all internet agent, there is no way of creating EFDs and Logs associated with the resources implemented by the CMIP agent but not the resources implemented by the internet agents.

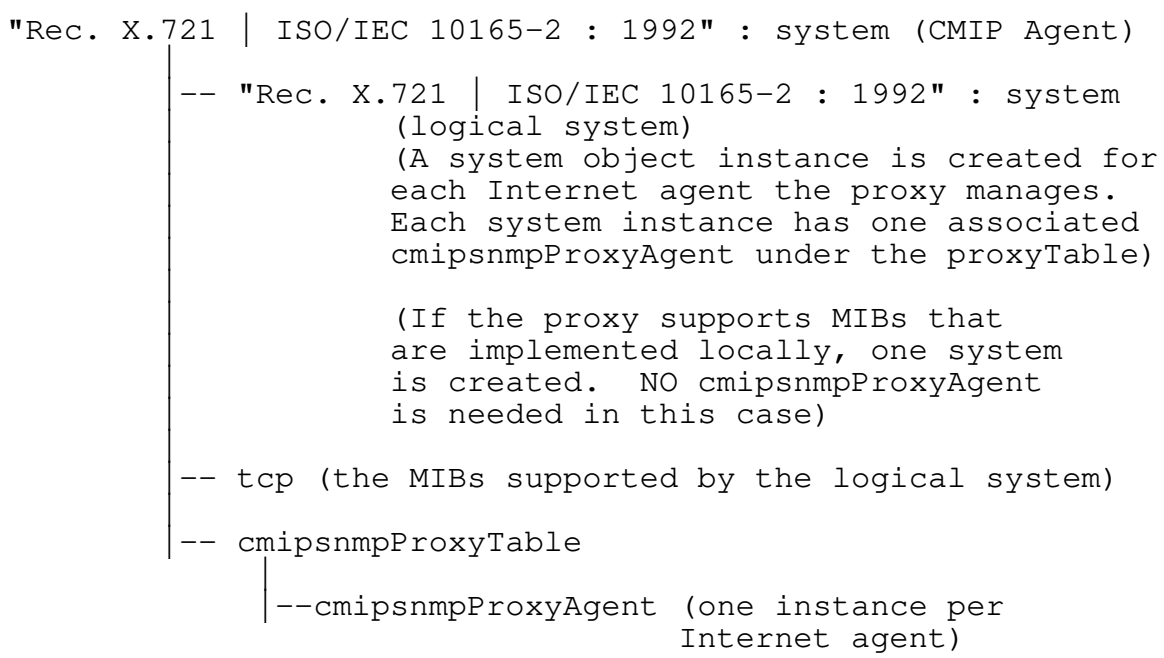
B) As described in the previous document (Draft 0)



Pro's:
Best effort scoping can easily be supported across all the Internet agents.

Con's:
From the perspective of the ISO manager, different containment hierarchy is used by the proxy and native CMIP

C) Alternative Proposal



Pro's:

From the perspective of the ISO/CCITT manager, all agents are alike in their containment hierarchy; whether via proxy or native CMIP agents.

Best effort scoping can be supported across all the Internet agents.

If the ISO/CCITT manager wants to manage all the MIBs implemented in the Internet agents and the MIBs supported by the proxy, the ISO/CCITT manager can establish ONE connection (only) to the PROXY SYSTEM.

If the ISO/CCITT manager needs to get information from the tcp object from a native CMIP agent, the ISO/CCITT manager can use a management operation with local name {tcp}. If the ISO/CCITT manager wants to get information from the tcp object implemented by an Internet agent, then the ISO/CCITT manager can use management operation with local name {CMIP Agent system, tcp}.

If the ISO/CCITT manager only needs to set up one EFD to manage the same condition across all the MIBs implemented in the Internet agents and the MIBs implemented locally. Only one CMIP association needs to be established between the ISO/CCITT manager and the proxy.

If the ISO/CCITT manager wants to manage a particular "logical system", an association can be established

between the ISO/CCITT manager and the "logical system". The local name can be used relative to the "logical system" and EFDs can be set up for this particular "logical system".

Con's:

Some CMIP-based agent implementations may not have the capability of implementing multiple system objects.

References

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

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

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

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

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

[ISO10164-1] ISO/IEC IS 10164-1: Information Technology - Open Systems Interconnection - Systems Management - Part 1: Object Management Function, 1991.

[ISO10164-2] ISO/IEC IS 10164-2: Information Technology - Open Systems Interconnection - Systems Management - Part 2: State Management Function, 1991.

[ISO10164-3] ISO/IEC IS 10164-3: Information Technology - Open Systems Interconnection - Systems Management - Part 3: Attributes For Representing Relationships, 1991.

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

[ISO10164-5] ISO/IEC IS 10164-5: Information Technology - Open Systems Interconnection - Systems Management - Part 5: Event Report Management Function, 1991.

[ISO10164-6] ISO/IEC IS 10164-6: Information Technology - Open Systems Interconnection - Systems Management - Part 6: Log Control Function, 1991.

[ISO10164-7] ISO/IEC IS 10164-7: Information Technology - Open Systems Interconnection - Systems Management - Part 7: Security Alarm Reporting Function, 1991.

[ISO10164-8] ISO/IEC IS 10164-8: Information Technology - Open Systems Interconnection - Systems Management - Part 8: Security Audit Trail Function, 1992.

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

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

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

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

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

[RFC1157] RFC 1157, J.D. Case, M.S. Fedor, M.L.Schoffstall, C. Davin, Simple Network Management Protocol (SNMP), May 1990.

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

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

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

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

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

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

[SNMPv2SMI] J.D. Case, K. McCloghrie, M.T. Rose, S.L.Waldbusser, Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2), Internet-draft, December 1992.

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

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

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

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

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

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

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

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

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

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

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

[AOM1UL] ISO/IEC ISP 11183-1, Information Technology - International Standardized Profiles AOM1n OSI Management - Management Communications Protocols - Part 1: Specification of ACSE, Presentation, and Session Protocols for the use by ROSE and CMISE, May 1992.

[AOM12] ISO/IEC ISP 11183-2, Information Technology - International Standardized Profiles AOM1n OSI Management -

Management Communications Protocols - Part 2: AOM12 - Enhanced Management Communications, June 1992.

[AOM11] ISO/IEC ISP 11183-3, Information Technology - International Standardized Profiles AOM1n OSI Management - Management Communications Protocols - Part 3: AOM11 - Basic Management Communications, May 1992.

[OP1LIBV4] Network Management Forum: Forum 006, OMNIPoint 1 Library Volume 4, Issue 1.0, August, 1992.