Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 18 - Network Management

Output from the September 1993 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair **Paul Brusil, The Mitre Corporation**
SIG Editor **Robert Aronoff, NIST**

Foreword

This part of the Stable Implementation Agreements was prepared by the Network Management Special Interest Group (NMSIG) of the Open Systems Environment Implementors' Workshop (OIW). See Procedures Manual for Workshop charter.

Text in this part has been approved by the Plenary of the above–mentioned Workshop. This part replaces the previously existing chapter on this subject.

To highlight textual changes since the last Workshop output, additions to the text in this part are marked with shading; deleted text is left in but marked with strikeouts.

Table of Contents

List of Tables

18 Network Management

0 Introduction

Within the community of OSI researchers, users, and vendors, there is a recognized need to address the problems of initiating, terminating, monitoring, and controlling communication activities and assisting in their harmonious operation, as well as handling abnormal conditions. The activities that address these problems are collectively called network management.

Network management can be viewed as the set of operational and administrative mechanisms necessary to:

a. bring up, enroll, and/or alter network resources;

b. keep network resources operational;

c. fine tune these resources and/or plan for their expansion;

d. manage the accounting of their usage;

e. manage their protection from unauthorized use/tampering.

As such, network management is typically concerned with management activities in at least the following five functional areas: configuration management, fault management, performance management, accounting management, and security management. In order to accomplish these management activities, information must be exchanged among open systems.

In Part 18, there are Implementation Agreements (IA's) for providing interoperable OSI management information communication services among OSI systems. Also contained here are agreements on management information. These agreements pertain to the exchange of management information and management commands between open systems operating in a multivendor environment. For example, one goal is to ensure that a management system built by one vendor can manage objects built by another vendor.

1 Scope

The purpose of this Part (Part 18), is to provide implementation agreements that will enable independent vendors to supply customers with a diverse set of networking products that can be managed as part of an integrated environment. Where possible, these agreements are based upon OSI Systems Management standards.

1.1 Phased Approach

Because of the broad scope of the subject, and given that OSI Systems Management standards are still evolving, it is reasonable to assume that a comprehensive set of network management implementation agreements will take a number of years to develop. To arrive at an initial set of implementation agreements in a timely fashion, a phased approach has been adopted.

This phased work approach will result in a series of implementation agreements based on the expanding scope of the OSI Systems Management standards. It is the intention of the NMSIG to define the content of each phase as a compatible superset of the previous Phases to ensure that Phase N products can interact with products based on the implementation agreements of earlier phases.

1.1.1 Alignment With Evolving Standards

In some cases, these phased implementation agreements may be based on DIS standards. As the relevant standards progress from DIS to IS, the agreements will be aligned in future phases.

When a defect is found in any of the management related standards, the reported defect may be technically resolved by the appropriate international technical committee with likely approval by the voting members pending for several months. Since relevant defects can't be ignored in an implementation, these agreements will note defect resolutions which have the tentative approval of the appropriate standards committee. These interim resolutions will be recorded in clause 4.

Once a defect resolution has been completed by the appropriate standards body, the agreed upon resolution will be incorporated into the next phase of these implementors agreements. If appropriate, a previous phase that relied on an interim resolution will be examined to determine whether errata should be issued to bring the original phase into line with the final resolution.

1.1.2 Definition of Phase 1

As a first step in this phased approach, the NMSIG has targeted an initial set of agreements that provide limited interoperable management in a heterogeneous vendor environment. They are the beginning of a comprehensive set of implementation agreements based on the emerging OSI Systems Management standards. Furthermore, these initial agreements allow the community to gain experience with OSI management standards as they emerge.

The focus of the Phase 1 agreements is to enable a managing process provided by one vendor to interoperate with an agent process provided by a different vendor to perform limited management on a set of managed objects.

The scope of Phase 1 implementation agreements is the following:

Management Functions:

Object Management Function [OMF],
State Management Function [STMF],
Attributes For Representing Relationships [ARR],
Alarm Reporting Function [ARF],
Event Report Management Function [ERMF].

Management Information:

Information Model, Naming, Guidelines and Templates for Defining Managed Objects

Management Communication:

CMIS/P, Association Policies, and Upper Layer Services Required

Management Objects:

Support Objects required for the above.

Editor's Note: **[**The relation of the MIL definitions in Annex A of the Working Document to Phase 1 IA's needs to be clarified.**]**

Conformance Criteria:

Conformance Criteria for the above functionality.

To accomplish these goals in a timely fashion, the following simplifying constraints have been reflected in the Phase 1 agreements:

1. No agreements are provided regarding management domains;

2. These agreements require only the following application service elements: the Association Control Service Element (ACSE), the Common Management Information Service Element (CMISE), Remote Operations Service Element (ROSE), and the System Management Application Service Element (SMASE);

3. These agreements do not require implementation of services defined by the Directory standards;

4. No agreements regarding the security of management are provided.

1.1.3 Future Phases

It is the intention of the NMSIG to freeze the content of Phase 1 when these agreements are progressed to Stable status. Alignment changes required as the standards progress from DIS to IS will be made in future phases.

As standards defining new functionality are progressed, the NMSIG will define future phases incorporating the new functionality as a compatible superset of previous phases.

2 Normative References

The following documents are referenced in the statements of the agreements relating to OSI systems management.

Editor's Note: [Items marked with an asterisk, "*", are ones which, while not cited in the text of this part of the IAs, are included here, nevertheless, to indicate where useful background information can be found.]

[ACSEP] ISO 8650, Information Processing Systems – Open Systems Interconnection – Protocol Specification for the Association Control Service Element (Revised Final Text of DIS 8650), ISO/IEC JTC1/SC21 N2327, 21 April 1988.

[ACSES]* ISO 8649, Information Processing Systems – Open Systems Interconnection – Service Definition for the Association Control Service Element (Revised Final Text of DIS 8649), ISO/IEC JTC1/SC21 N2326, 21 April 1988.

[ADDRMVP] ISO/IEC 9596/DAD 2, Common Management Information Protocol Specification: Addendum 2 (Add/Remove Protocol), ISO/IEC JTC1/SC21, 1 February 1990.

[ADDRMVS] ISO/IEC 9595/DAD 2, Common Management Information Service Definition: Addendum 2 (Add/Remove Service), ISO/IEC JTC1/SC21, 1 February 1990.

[ALS]* ISO/IEC DIS 9545, Information Processing Systems – Open Systems Interconnection – Application Layer Structure, 15 March 1989.

[AOM1PT1] ISO/IEC ISP 11183–1, Information Technology – International Standardized Profiles AOM1n OSI Management – Management Communications – Part 1: Specification of ACSE, Presentation and Session Protocols for the use by ROSE and CMISE, May 1992.

[AOM1PT3] ISO/IEC ISP 11183–3, Information Technology – International Standardized Profiles AOM1n OSI Management – Management Communications – Part 3: CMISE/ROSE for AOM11 – Basic Management Communications, May 1992.

[AOM1PT2] ISO/IEC ISP 11183–2, Information Technology – International Standardized Profiles AOM1n OSI Management – Management Communications – Part 2: CMISE/ROSE for AOM12 – Enhanced Management Communications, May 1992.

[AOM211] pDISP 12060–1, Information Technology – International Standardized Profiles – AOM2n OSI Management – Management Functions – Part 1: AOM211 – General Management Capabilities, July 1992.

[AOM212] pDISP 12060–2, Information Technology – International Standardized Profiles – AOM2n OSI Management – Management Functions – Part 2: AOM212 – Alarm Reporting and State Management Capabilities, July 1992.

[AOM213] pDISP 12060–3, Information Technology – International Standardized Profiles – AOM2n OSI Management – Management Functions – Part 3: AOM213 – Alarm Reporting Capabilities, July 1992.

[AOM221] pDISP 12060–4, Information Technology – International Standardized Profiles – AOM2nn OSI Management – Management Functions – Part 4: AOM221 – General Event Report Management, July 1992.

[AOM231] pDISP 12060–5, Information Technology – International Standardized Profiles – AOM2n OSI Management – Management Functions – Part 5: AOM231 – General Log Control Profile, July 1992.

[ARF] ISO/IEC IS 10164–4, Information Technology – Open Systems Interconnection – Systems Management – Part 4: Alarm Reporting Function, ISO/IEC JTC1/SC21 N6359, August 19, 1991.

[ARR] ISO/IEC IS 10164–3, Information Technology – Open Systems Interconnection – Systems Management – Part 3: Attributes for Representing Relationships, ISO/IEC JTC1/SC21 N5186, September 1991.

[ASN1]* ISO/IEC 8824, Information Technology – Open System Interconnection – Specification of Abstract Syntax Notation One (ASN.1), ISO/IEC JTC1/SC21 N4720, 30 April 1990.

[BER] ISO/IEC 8825, Information Technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), ISO/IEC JTC1/SC21 N4721, 30 April 1990.

[CANGETP] ISO/IEC 9596/DAD 1, Common Management Information Protocol Specification: Addendum 1 (CancelGet Protocol), ISO/IEC JTC1/SC21, 1 February 1990.

[CANGETS] ISO/IEC 9595/DAD 1, Common Management Information Service Definition: Addendum 1 (CancelGet Service), ISO/IEC JTC1/SC21, 1 February 1990.

[CMIP] ISO/IEC 9596–1, Information Technology – Open Systems Interconnection – Common Management Information Protocol Specification – Part 1: Specification, 24 November I990.

[CMIS] ISO/IEC 9595, Information Technology – Open Systems Interconnection – Common Management Information Service Definition, Common Management Information Service, 24 November 1990.

[DIR]* ISO 9594 – Information Processing Systems – Open Systems Interconnection – The Directory, 1988.

[DMI] ISO/IEC IS 10165–2, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 2: Definition of Management Information, ISO/IEC JTC1/SC21 N6363, August 1991.

[ENSCON]* Forum 025, The "Ensemble" Concepts and Format, Issue 1.0, Network Management Forum, July 1992.

[ERMF] ISO/IEC IS 10164–5, Information Technology – Open Systems Interconnection – Systems Management – Part 5: Event Report Management Function, ISO/IEC JTC1/SC21 N6360, August 1991.

[FRMWK]* ISO 7498–4, Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management Framework, 1989.

[GDMO] ISO/IEC IS 10165–4, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 4: Guidelines for the Definition of Managed Objects, ISO/IEC JTC1/SC21 N6309, July 30, 1991.

[ISPARR3] pDISP 12059–3, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 3: Attributes for Representing Relationships, July 1992.

[ISPAR4] pDISP 12059–4, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 4: Alarm Reporting, July 1992.

[ISPCOM0] pDISP 12059–0, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 0: Common Definitions for Management Function Profiles, July 1992.

[ISPERM5] pDISP 12059–5, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 5: Event Report Management, July 1992.

[ISPFRM] ISO/IEC TR 10000–1, Information Technology – Framework and Taxonomy of International Standardized Profiles – Part 1: Framework, ISO/IEC JTC1/SGFS N184, 9 February 1990.

[ISPLC6] pDISP 12059–6, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 6: Log Control, July 1992.

[ISPOM1] pDISP 12059–1, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 1: Object Management, July 1992.

[ISPSRVC] ISO/IEC TR 8509, Information Processing Systems – Open Systems Interconnection – Service Conventions, TC97/SC16/1646.

[ISPSTM2] pDISP 12059–2, Information Technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 2: State Management, July 1992.

[LCF] ISO/IEC IS 10164–6, Information Technology – Open Systems Interconnection – Systems Management – Part 6: Log Control Function, ISO/IEC JTC1/SC21 N6361, June 1991.

[MGNM]* CCITT Recommendation M.gnm, Draft Recommendation (M.gnm) Generic Network Information Model, CCITT SGIV, December 3, 1991.

[MIM] ISO/IEC IS 10165–1, Information Technology – Open Systems Interconnection – Management Information Services – Structure of Management Information – Part 1: Management Information Model, ISO/IEC JTC1/SC21 N6351, June 1991.

[NMSIG1] OIW Endorsement/Comment on System Management Function Taxonomy (Including Proposed Function Taxonomy), NMSIG–91/164, September 1991.

[OMF] ISO/IEC IS 10164–1, Information Technology – Open Systems Interconnection – Systems Management – Part 1: Object Management Function, ISO/IEC JTC1/SC21 N5184, September 1991.

[OP1LIB]* Forum 006, Forum Library – Volume 4: OMNIPoint 1 Definitions, Issue 1.0, Network Management Forum, August 1992.

[PPS]* ISO/IEC DIS 8823, Information Processing Systems – Open Systems Interconnection – Connection Oriented Presentation Protocol Specification, ISO/IEC JTC1/SC21 N2336, 5 April 1988.

[PSD]* ISO/IEC Final Text of DIS 8822, Information Processing Systems – Open Systems Interconnection – Connection Oriented Presentation Service Definition, ISO/IEC JTC1/SC21 N2335, 5 April 1988.

[ROSEP]* ISO/IEC 9072–2 – Information Processing Systems – Text Communications – Remote Operations Part 2: Protocol Specification, 19 September 1989.

[ROSES]* ISO/IEC 9072–1, Information Processing Systems – Text Communications – Remote Operations Part 1: Model, Notation and Service Definition, 19 September 1989.

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

[SATF] ISO/IEC DIS 10164–8, Information Technology – Open Systems Interconnection – Systems Management – Part 8: Security Audit Trail Function, ISO/IEC JTC1/SC21 N7039, June 1992.

[SMO] ISO/IEC IS 10040, Information Technology – Open Systems Interconnection – Systems Management Overview, ISO/IEC JTC1/SC21 N6353, August 1991.

[STMF] ISO/IEC IS 10164–2, Information Technology – Open Systems Interconnection – Systems Management – Part 2: State Management Function, ISO/IEC JTC1/SC21 N5185, September 1991.

3 Status

As of September 1991, the Stable management communications agreements in clause 6 of part 18 and clause 13.7 of part 5 became technically equivalent to DISP 11183. The DISP, however, is a more rigorous statement of specifications. Therefore, it has been the stated intent of the NMSIG to directly reference the ISP 11183, Parts 1 through 3, and all the agreements therein, when the DISP reaches ISP status. Since the DISP has now progressed to ISP 11183 with no technical changes, the NMSIG Stable management communications agreements in clause 6 of part 18 have now been changed to point directly to ISP 11183–1 through –3 [AOM1PT1, AOM1PT2, and AOM1PT2].

(Refer to the Working Implementation Agreements Document for additional status information.)

4 Errata

Editor's Note: ["Defect Report" material (including applicability) may be included here.]

The following table indicates the clause, type, and reference document of technical errata to this part.

Erratum No.	**Type & Date Entered**	Referenced Document	Clause	Comment
1	Technical 6/91	NMSIG-91/08	6.4.5	This clause, previously clause 6.2.6, was modified and moved to clause 6.4.5 to clarify that it is intended as a support agreement for CMIP rather than a usage agreement for CMIS.
2	Alignment 9/91	NMSIG-91/110 NMSIG-91/113	5	This clause has been updated to reflect alignment changes to the relevant base standards which have just progressed to IS as of August, 1991.
3	Technical 9/91	NMSIG-91/161	6.2.2.2	Move text from clause 5.1.2.1 to more appropriate clause 6.2.2.2 and clarify required support for minimal filter complexity to align with the DISP 11183.
4	Technical 9/91	NMSIG-91/161	6.2.3	Remove unnecessary restrictions on sending CMIP time parameters.
5	Alignment 9/91	NMSIG-91/114	6.1.1	Change reference to required application context support to align with IS version of [SMO].
6	Technical 9/91	NMSIG-91/161	6.3.3.1	Remove clause requiring mandatory attribute list in successful set response because considered redundant information.

Erratum No.	**Type & Date Entered**	Referenced Document	Clause	Comment
7	Alignment 9/91	NMSIG-91/120	7	Update clause to reflect alignment changes to the relevant base standards which have progressed to IS as of August, 1991.
8	Editorial 9/91	NMSIG-91/161	6.3.6.1	Move clause 6.3.6.1 to more appropriate location at clause 7.1.5.
9	Alignment 3/92	NMSIG-92/066	6.1.3	Update reference because number of clause in other part of OIW Stable Agreements changed.
10	Alignment 6/92	NMSIG-92/093	5.2 - 5.7	Update text to reference appropriate AOM2x pDISPs because have equivalent agreements, but are more rigorous.
11	Alignment 6/92	NMSIG - 92/200	6	Update text to reference ISP 11183 which is technically equivalent with IA text but is more rigorous.
12	Technical 12/92	NMSIG-92/409	A.5.1.2	Modify package name, transportConnectionRetransmissionIV MO-Package, which was incorrectly specified in the CHARACTERIZED BY clause of the MO class definition.
13	Technical 12/92	NMSIG-92/409	A.5.1.2	Modify object ID in REGISTERED AS clause of the MO class definition to register the newly modified MO (see erratum 12).
14	Technical 12/92	NMSIG-92/409	B.3.1	Modify object identifier value in TABLE B.10 to reflect changes in MO definition (see errata 12 and 13).
15	Technical 12/92	NMSIG-92/409	C.4.9	Modify object identifier value in the table to reflect changes in MO definition (see errata 12 and 13).

Erratum No.	**Type & Date Entered**	Referenced Document	Clause	Comment
16	Editorial 3/93	NMSIG-93/078	A.4.7	Added IMPORT statement for NameType and correctored ASN.1 syntax for AddressingSizeRange, EquipmentIdRange, MemorySizeRange, and OsInfoRange. Previous syntax was incorrect and would not work.
17	Editorial 3/93	NMSIG-93/086	A.4.4.24	Changed text to reflect singular rather than plural "manufacturer".
18	Editorial 3/93	NMSIG-93/086	A.4.4.31	Added text to clarify the specific object class name cited.
19	Editorial 3/93	NMSIG-93/086	A.4.4.54	Corrected grammar to have singular rather than plural verb.
20	Alignment 9/93	NMSIG-93/333R3	5.8.2	Change table note [1], for the Security Alarm Detector, to align with Draft Amendment 10164-7.
21	Alignment	NMSIG-93/333R3	5.11	Change function name to align with function name change in standard.
22	Alignment	NMSIG-93/333R3	5.12	Change function name to align with function name change in standard.

5 Management Functions and Services

5.1 General Agreements

5.1.1 Conventions Used In SMF Agreements

Each System Management Function defines a set of services referred to in this document as "SMF services." Agreements pertinent to SMF services are provided in the following subclauses. Each subclause contains a series of tables, as follows.

For each SMF service, a normative table references text agreements which constrain the usage and/or value of the associated service parameters. Text agreements defined elsewhere in this document are referenced by clause number. The lack of a row or reference signifies no agreement beyond the base standard.

These tables include codes which specify parameter usage for request, indication, response, and confirmation service primitives. These codes, defined in subclause 1.8.3 of these agreements (Classification of Conformance), in ISO/IEC TR 10000-1 (Framework and Taxonomy of ISPs) [ISPFRM], and in ISO/IEC TR 8509 (Service Conventions) [ISPSRVC], are repeated here for reader convenience:

M	Mandatory
O	Optional
C(p)	If Condition p exists, then parameter is mandatory; otherwise, the parameter is not applicable.
X	Excluded
I	Out Of Scope In these agreements, this means that, for the corresponding element, * implementations may use it outside the scope of these agreements, * conformance tests shall not be provided for it, * implementations may conform to other agreements where it is required, * no requirements are placed on either transmitter or receiver to support it, * receiver actions are unspecified when present.
-	Not Applicable
(=)	The value of the parameter is identical to the corresponding parameter in the interaction described by the preceding related service primitive.
U	The use of the parameter is a service-user option.
P	The parameter is mapped directly onto the corresponding parameters of the CMIS service primitive; refer to subclause 6 for agreements regarding this pass-through parameter.

In addition, the convention "A>B" is used in normative tables to indicate both the usage specified by the base standard (A) and the additional constraint imposed by these agreements (B). This convention is intended to call attention to agreements which modify the usage of a service parameter.

Unless otherwise noted, conditional parameters (C) shall be present according to the conditions defined in [CMIS] and the referenced System Management Function base standard.

5.1.2 General Agreements Referenced By Many SMF Services

The following general agreements pertain to some or all of the System Management Function services defined throughout clause 5. Normative tables for each SMF service reference these general agreements where applicable. These agreements do not apply to SMF services and parameters which do not reference them.

5.1.2.1 Maximum Length of Notification Identifier

To limit implementation complexity, the maximum length of the Notification Identifier parameter shall be 32 bits.

5.1.2.2 Maximum Number of SET Items

To limit implementation complexity, the maximum number of SET items contained within specified SMF service parameters that recipients must be able to process shall be 64.

5.1.2.3 Maximum Length of Additional Text

To limit implementation complexity, the maximum length of the Additional Text parameter which recipients must be able to process shall be 256 octets.

5.1.2.4 Use of Additional Info

Editor's Note: [The Additional Information parameter, described in [ARF] clause 8.1.2.14, includes a "significance indicator." It requires that "[e]ven if the Additional Information parameter is not fully understood, an event report indication shall be issued to the user. Indication that the Additional Information parameter is not fully understood is a local matter."]

5.2 Object Management Function Agreements

5.2.1 General Agreements

These agreements require support for the SMF services defined by the object management standard [OMF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the object management standard [OMF] and specified in [DMI], with the exception of event record subclasses. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [OMF] event record subclasses shall also be required by these agreements. These agreements permit optional negotiation of the system management functional units specified in clause 10 of [OMF].

5.2.2 Specific Agreements

See [ISPOM1] for specification of agreements for the Object Management Function.

5.3 State Management Function Agreements

5.3.1 General Agreements

These agreements require support for the SMF services defined by the state management standard [STMF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the state management standard [STMF] and specified in [DMI], with the exception of event record subclass. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [STMF] event record subclasses shall also be required by these agreements. These agreements permit optional negotiation of the State Change Reporting functional unit specified in clause 10 of [STMF].

5.3.2 Specific Agreements

See [ISPSTM2] for specification of agreements for the State Management Function.

5.4 Attributes For Representing Relationships Agreements

5.4.1 General Agreements

These agreements require support for the SMF services defined by the Attributes For Representing Relationships standard [ARR].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the attributes for representing relationships standard [ARR] and specified in [DMI], with the exception of event record subclass. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [ARR] event record subclasses shall also be required by these agreements. These agreements permit optional negotiation of the Relationship Change Reporting functional unit specified in clause 10 of [ARR].

5.4.2 Specific Agreements

See [ISPARR3] for specification of agreements for Attributes for Representing Relationships.

5.5 Alarm Reporting Function Agreements

5.5.1 General Agreements

These agreements require support for the SMF services defined by the alarm reporting standard [ARF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the alarm reporting standard [ARF] and specified in [DMI], with the exception of event record subclass. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [ARF] event record subclasses shall also be required by these agreements. These agreements permit optional negotiation of the Alarm Reporting functional unit specified in clause 10 of [ARF].

5.5.2 Specific Agreements

See [ISPAR4] for specification of agreements for the Alarm Reporting Function.

5.6 Event Report Management Function Agreements

5.6.1 General Agreements

These agreements require support for the SMF services defined by the event report management standard [ERMF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the event report management standard [ERMF] and specified in [DMI]. These agreements permit optional negotiation of the Monitor Event Report Management and Event Report Management functional units specified in clause 10 of [ERMF].

5.6.2 Specific Agreements

See [ISPERM5] for specification of agreements for the Event Report Management Function.

5.7 Log Control Function Agreements

5.7.1 General Agreements

These agreements require the SMF services defined by the log control standard [LCF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the log control standard [LCF] and specified in [DMI].

If any other function defined in clause 5 that supports notifications is supported, then any event record subclass defined by that function is required for the log control function.

These agreements permit optional negotiation for log control and monitor log control functional units specified in section 10 of [LCF].

The appropriate CMIS error (i.e., invalidAttributeValue) shall be returned for any attempt to set Max log size less than the value of Current log size, except if setting the Max log size to zero. When the Max log size is set to zero, then the maximum log size is unlimited.

5.7.2 Specific Agreements

See [ISPLC6] for specification of agreements for the Log Control Function.

5.8 Security Alarm Reporting Function Agreements

(Refer to the Working Implementation Agreements Document.)

5.8.1 General Agreements

These agreements require support for the SMF services defined by the security alarm reporting standard [SARF].

These agreements also require conformance to the abstract syntaxes identified in clause 11 of the alarm reporting standard [SARF] and specified in [DMI], with the exception of event record subclass. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [SARF] event record subclasses shall also be required by these agreements. These agreements permit optional negotiation of the security alarm reporting function as specified in section 10 of [SARF].

5.8.2 Security Alarm Reporting

This subclause provides agreements pertinent to the Security Alarm Reporting SMF service defined by section 9.2 of [SARF]. Subclause 6 provides agreements pertinent to CMIS services and pass–through parameters used by this SMF service.

Table 1 - Agreements on parameter usage pertinent to the Security Alarm Reporting SMF service

SMF Security Alarm Reporting parameter	Req	Rsp	SMF agreements
Event Type	M	C(=)	
Event Information			
Security Alarm Cause	M	-	
Security Alarm Severity	M	-	
Security Alarm Detector	M	-	[1]
Service User	M	-	
Service Provider	M	-	

SMF Security Alarm Reporting parameter	Req	Rsp	SMF agreements
Notification Identifier	U	-	5.1.2.1
Correlated Notifications	U	-	5.1.2.2
Additional Text	U	-	5.1.2.3
Additional Info	U	-	5.1.2.2, 5.1.2.4

[1] ~~To avoid ambiguity, the Distinguished Name form of this parameter shall be implemented and may be used.~~ In the case of manager receiving, the Distinguished Name, Local Distinguished Name and Non–Specific forms shall be implemented and may be used. In the case of agent sending, the Distinguished Name form of this parameter shall be implemented and may be used. Use of Local Distinguished Name and Non–Specific forms are beyond the scope of these agreements. If an implementation is unable to decode or understand the semantics of this parameter, an appropriate CMIS error (i.e., Invalid Attribute Value) shall be returned.

5.9 Security Audit Trail Function Agreements

5.9.1 General Agreements

These agreements require support for the SMF services defined by the security audit trail standard [SATF].

These agreements also require conformance to the abstract syntaxes identified in clause 11.2 of the security audit trail standard [SATF] and specified in [DMI], with the exception of event log record subclass. If support for the log control standard [LCF] as described in clause 5.7 is claimed, then all [SATF] event log record subclasses shall also be required by these agreements.

5.9.2 Security Audit Trail Reporting SMF Service

This subclause provides agreements pertinent to the Security Audit Trail Reporting SMF service defined by section 9.2 of [SATF]. Clause 6 provides agreements pertinent to CMIS services and pass–through parameters used by this SMF service.

Table 2 - Agreements on parameter usage pertinent to the Security Audit Trail Reporting SMF service

SMF Security Audit Trail parameter	Req	Rsp	SMF agreements

Event Type	M	C(=)	5.9.2.1
Event Information			
Service Report Cause	C(1)	-	
Notification Identifier	U	-	5.1.2.3
Correlated Notifications	U	-	5.1.2.4
Additional Text	U	-	5.1.2.5
Additional Info	U>I	-	5.1.2.6

C(1): Manadatory (M) for serviceReport

5.9.2.1 Notifications

These Implementors' Agreements require support for both the seviceReport and usageReport notification types.

5.9.3 Security Audit Trail Record

This subclause is a placeholder for agreements pertaining to the Security Audit Trail Record (SATR) managed object class.

5.10 Objects and Attributes for Access Control Agreements

(Refer to the Working Implementation Agreements Document.)

5.11 ~~Accounting Meter~~Usage Metering Function Agreements

(Refer to the Working Implementation Agreements Document.)

5.12 ~~Workload Monitoring Function~~Metric Objects and Attributes Agreements

(Refer to the Working Implementation Agreements Document.)

5.13 Summarization Function Agreements

(Refer to the Working Implementation Agreements Document.)

5.14 Test Management Function Agreements

(Refer to the Working Implementation Agreements Document.)

5.15 Confidence and Diagnostic Test Classes Agreements

(Refer to the Working Implementation Agreements Document.)

6 Management Communications

This clause covers the agreements pertaining to the use of associations over which to conduct management communications, and agreements for management communication, itself, by reference to ISP 11183 [AOM1PT1, AOM1PT2, and AOM1PT3]. ISP 11183 defines two profiles, AOM11 (Basic Management Communications) [AOM1PT3] and AOM12 (Enhanced Management Communications) [AOM1PT2], and defines upper layer requirements [AOM1PT1] for each of these profiles.

For rigorous specification of the agreements relevant to clause 6, Management Communications, see ISP 11183 [AOM1PT1, AOM1PT2, and AOM1PT3].

6.1 Association Policies

Associations are established using the procedures described in [ACSEP].

6.1.1 Application Context Negotiation

These IAs specify the negotiation of the Systems management application context specified in [SMO]. Other application contexts are outside the scope of these agreements.

6.1.2 Functional Unit Negotiation

These IAs specify that System Management Functional Units are negotiated as specified in [SMO].

6.1.3 Security Aspects of Associations

The ACSE authentication mechanisms and associated data types shall be as defined in clause 9 (Upper Layers Security) of part 12 of the OIW Working Agreements.

Support of ACSE authentication is optional.

7 Management Information

This clause, which is based on ISO standards' documents [MIM] and [GDMO], contains agreements regarding basic concepts and modelling techniques related to management information. It enumerates agreements on (i) the information model (subclause 7.1) and (ii) guidelines for defining management information (subclause 7.2). These agreements apply to developers of contributions to the Management Information Library (MIL). They form a normative part of the standard; hence they must be strictly followed while defining management information. It is not within the scope of this clause to make agreements about specific elements of management information or to define such specific elements of management information. Such definitions and/or agreements can be obtained via the Management Information Library.

7.1 The Information Model

When modelling management information, these agreements require use of [MIM] with the following additional constraints.

7.1.1 Inheritance

The following constraint related to inheritance is enforced in order to remove potential ambiguities:

> During the lifetime of a managed object instance, each of its attributes must have a value that is valid for the attribute syntax of that attribute.

7.1.2 Interoperability

7.1.2.1 Interoperability Provided By The Agent System

Allomorphism, as specified in clause 5.2.3.1 of [MIM], is out of scope. Any other specification within the [MIM] or [GDMO] that refers to allomorphism is also out of scope.

7.1.2.2 Interoperability Provided By The Manager System

The semantics of clause 5.2.3.2 of [MIM] are supported. A manager system can supply the object identifier as specified in clause 7.4.5 of [GDMO] to specify that a managed object should perform an operation as a member of its actual class. The object identifier is intended to be used in requests only, and shall be interpreted by the responder as a requirement to return its real object class value in the response. Agent systems shall support this object identifier as defined in [MIM] 5.2.3.2 and [GDMO] 7.4.5.

7.1.3 Filter

The concept of filter is supported as specified in clause 6. Restrictions on its usage are specified in subclause 6.2.2.2 of these agreements.

7.1.4 Management Operations

An implementation that complies with these agreements shall support management operations as defined in clause 5.3.4 of [MIM] with the following additional clarification.

> [MIM] clause 5.3.4.1 (2), [DMI] clause 6.14, and [GDMO] clause 6.1.4 imply that the object class attribute shall not be included in the create request Attribute List parameter. [MIM] states that any conflicting duplicate specifications cause the request to fail.

7.1.5 Deletion of Objects Containing Objects

The error "Processing Failure" shall be returned if a managed object has existing contained objects and the behavior defined for that object prohibits its deletion unless all contained objects have been deleted.

7.2 Guidelines for the Definition of Management Information

This subclause contains agreements about guidelines for the definition of management information, as specified in [GDMO].

7.2.1 Syntactical Definitions of Management Information

7.2.1.1 Attribute Template

The following constraint applies to the Attribute Template specified in clause 9.7.2 of [GDMO]:

> The BEHAVIOUR construct may be omitted only if a behaviour definition has been inherited from the parent attribute, i.e., the attribute is derived from another attribute whose definition contains a BEHAVIOUR construct.

7.2.2 Guidelines For Defining Behaviour

The following details should be provided in the set of specifications defining a managed object class:

a) a textual description of the network/system resource(s) the managed object class represents, including their functional role;

b) a description of the relationships that can occur between different instances of the managed object class being defined, as well as those that can occur between instances of the managed object class being defined and instances of other managed object classes;

c) a description of the operations that are supported by the managed object class, with precise definition of the effects, side effects if any, constraints, response notifications, failure modes;

d) specification of how instances of this managed object class are created and deleted, particularly whether they can be created/deleted via the management CREATE/DELETE operations;

e) a description of notifications that can be generated, the conditions that generate them (e.g., crossing of a threshold), their contents and side–effects, if any. In particular, identify all the attributes that are subject to the AttributeChange and StateChange notifications, if these notifications are supported;

f) other constraints, including those involving other managed object classes.

7.2.3 Other Guidelines

The Systems Management functions have defined various attributes and events, as indicated in clause 5 of these agreements. Object definers should make use of these attributes and events wherever applicable.

8 Conformance

8.1 Introduction

Clause 8 specifies the conformance requirements for the OIW Network Management Implementation Agreements (IAs). Implementors of products will provide claims of conformance to these requirements. These claims will be in the form of Protocol Implementation Conformance Statements (PICS) and Managed Object Conformance Statements (MOCS). These requirements will also be used to develop test cases which will be used to validate claims of conformance. This clause defines the general conformance requirements and criteria which shall be used as a basis for tests of implementations claiming conformance to these agreements. Dependent conformance requirements are defined in the context in which they are used (e.g., SMF general conformance requirements include CMIP dependent conformance requirements for CMIS services used).

Editor's Note: [The use of the two terms, "general conformance class" and "dependent conformance class", is under review. When a final answer to Question Q1/49.9 (on the long term solution to general and dependent conformance) has been approved, it is intended to clarify and/or correct this conformance section.]

(Refer to the Working Implementation Agreements Document for additional introductory text.)

8.2 General Requirements of Conformance

Conformance for these agreements is designed to specify a well–defined set of management capabilities and features. For the purposes of organization and clarity of these agreements, management has been divided into three categories. Clauses 5 (Management Functions and Services), 6 (Management Communications) and 7 (Management Information) state the agreements which respectively comprise three conformance categories. Within these categories, particular conformance categories are specified which delineate conformance requirements for a well–defined and bounded set of management capabilities and features (e.g., within the Management Functions and Services conformance categories, a conformance category is specified which defines conformance to Alarm Reporting and State Management Services). Once a conformance category is delineated which specifies the set of requirements for that category, tests can be developed to evaluate conformance of products to that conformance category. And finally, for some conformance categories, roles (Manager, Agent, or both) are specified. One or more roles may be supported for those conformance categories to which an implementation is conformant.

The development of conformance categories will enable:

a) users to define procurement specifications;

b) vendors to define management capabilities and features;

c) conformance test houses and others to define test cases.

To be conformant to the IAs, an implementation shall be conformant to at least one of the following categories:

a) Management Communication;

b) Management Functions and Services;

c) Management Information.

Implementations which are conformant to these categories shall comply with the requirements stated in the following clause.

8.3 Specific Conformance Categories

8.3.1 Management Communication Categories

To be conformant to the Management Communication categories, an implementation shall conform to agreements in clause 6. Conformance to management communication also requires an implementor to state which of the management communication profiles specified in clause 6 are supported in the implementation. The implementor's statement of which profile is supported shall be indicated in a CMIP PICS as follows. The implementor shall complete the PICS proforma as specified by one of the profiles specified in clause 6.

Note: **[**Conformance requirements for these IAs, relating to services required of the upper layers and other ASEs, are discussed in part 5, clause 13.7**]**

8.3.2 Management Functions and Services Conformance Categories

To be conformant to the Management Functions and Services categories, an implementation shall conform to the agreements in clause 5 on at least one of the categories defined below in either a manager role, an agent role or both roles. [Note: These categories are aligned with the proposed AOM2x Profiles for Systems Management Functions.] [NMSIG1] Conformance to agreements in clause 5 requires conformance to referenced ISO standards/CCITT Recommendations and to all other clauses referenced in 5, including dependent conformance to the underlying services required by the SMFs.

The implementor shall state which of the following conformance categories are supported. For each category, the implementor shall complete the related PICS and MOCS proformas to indicate which functional unit(s) and role(s) are supported.

8.3.2.1 General Management Capabilities Conformance Category

Note: [This category corresponds to proposed profile AOM211 [AOM211].**]**

Conformance to the General Management Capabilities Conformance Category requires general conformance to the Object Management Function [OMF], general conformance to the State Management Function [STMF], general conformance to the Attributes for Representing Relationships Function [ARR], and general conformance to the Alarm Reporting Function [ARF]. To be conformant to the Object Management Function, an implementation shall conform to the requirements stated in [OMF]. In addition, an implementation shall conform to clause 5.2 of these agreements and all other clauses referenced in 5.2. To be conformant to the State Management Function, an implementation shall conform to the requirements stated in [STMF]. In addition, an implementation shall conform to clause 5.3 of these agreements and all other clauses referenced in 5.3. To be conformant with the Attributes for Representing Relationships Function, an implementation shall conform to the requirements stated in [ARR], In addition, an implementation shall conform to clause 5.4 of these agreements and all other clauses referenced in 5.4. To be conformant to the Alarm Reporting Function, an implementation shall conform to the requirements stated in [ARF]. In addition, an implementation shall conform to clause 5.5 of these agreements and all clauses referenced in 5.5.

8.3.2.2 Alarm Reporting and State Management Capabilities Conformance Category

Note: [This category corresponds to proposed profile AOM212 [AOM212].**]**

Conformance to the Alarm Reporting and State Management Capabilities Conformance Category requires general conformance to the State Management Function [STMF], general conformance to the Alarm Reporting Function [ARF], and dependent conformance to the Object Management Function [OMF]. To be conformant to the State Management Function, an implementation shall conform to the requirements stated in [STMF]. In addition, an implementation shall conform to clause 5.3 of these agreements and all other clauses referenced in 5.3. To be conformant to the Alarm Reporting Function, an implementation shall conform to the requirements stated in [ARF]. In addition, an implementation shall conform to clause 5.5 of these agreements and all clauses referenced in 5.5.

Dependent conformance to the Object Management Function required by the Alarm Reporting and State Management Capabilities Conformance Category requires support for the PT–SET and PT–GET elements of procedure in clauses 11.1.6 and 11.1.7 of [OMF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.2.7 and clause 5.2.9 of these agreements and all clauses referenced in 5.2.7 and 5.2.9. The implementation need only support the PT–SET and PT–GET elements of procedure as applied to the State Management Attributes identified in [STMF] and specified in [DMI]. An implementation shall also conform to the notifications identified in [STMF] and specified in [DMI].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the PT–GET and PT–SET elements of procedure as defined in clauses 11.1.6 and 11.1.7 of [OMF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.6 of [STMF].

8.3.2.3 Alarm Reporting Capabilities Conformance Category

Note: [This category corresponds to proposed profile AOM213 [AOM213].**]**

Conformance to the Alarm Reporting Capabilities Conformance Category requires general conformance to the Alarm Reporting Function [ARF]. To be conformant to the Alarm Reporting Function, an implementation shall conform to the requirements stated in [ARF]. In addition, an implementation shall conform to clause 5.5 of these agreements and all clauses referenced in 5.5.

8.3.2.4 General Event Report Management Conformance Category

Note: [This category corresponds to proposed profile AOM221 [AOM221].**]**

Conformance to the General Event Report Management Conformance Category requires general conformance to the Event Report Management Function [ERMF], dependent conformance to the Object Management Function [OMF], and dependent conformance to the State Management Function [STMF]. To be conformant to the Event Report Management Function, an implementation shall conform to the requirements stated in [ERMF]. In addition, an implementation shall conform to clause 5.6 of these agreements and all clauses referenced in 5.6.

Dependent conformance to the Object Management Function required by the General Event Report Management Conformance Category requires support for the PT–SET, PT–GET, PT–CREATE, PT–DELETE, object creation reporting, object deletion reporting, and attribute value change reporting elements of procedure in clauses 11.1.1 through 11.1.7 of [OMF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.2.2 through clause 5.2.7, and clause 5.2.9 of these agreements and all clauses referenced in these clauses. An implementation shall also conform to the notifications identified in [OMF] and specified in [DMI].

Dependent conformance to the State Management Function required by the General Event Report Management Conformance Category requires support for the state change reporting elements of procedure in clause 11.1 of [STMF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.3.2 of these agreements and all clauses referenced by clause 5.3.2. An implementation shall also conform to the notifications identified in [STMF] and specified in [DMI].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the PT–SET, PT–GET, PT–CREATE, PT–DELETE, object creation reporting, object deletion reporting, and attribute value change reporting elements of procedure as defined in clauses 11.1.1 through 11.1.7 of [OMF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.5 of [OMF].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the state change reporting elements of procedure as defined in clause 11.1 of [STMF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.6 of [STMF].

8.3.2.5 General Log Control Conformance Category

Note: [This category corresponds to proposed profile AOM231 [AOM231].**]**

Conformance to the Log Control Conformance Category requires general conformance to the Log Control Function [LCF], dependent conformance to the Object Management Function [OMF], dependent conformance to the State Management Function [STMF] and dependent conformance to the Alarm Reporting Function [ARF]. To be conformant to the Log Control Function, an implementation shall conform to the requirements stated in [LCF]. In addition, an implementation shall conform to clause 5.7 of these agreements and all clauses referenced in 5.7.

Dependent conformance to the Object Management Function required by the General Log Control Conformance Category requires support for the PT–SET, PT–GET, PT–CREATE, PT–DELETE, object creation reporting, object deletion reporting, and attribute value change reporting elements of procedure in clauses 11.1.1 through 11.1.7 of [OMF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.2.2 through clause 5.2.7, and clause 5.2.9 of these agreements and all clauses referenced in these clauses. An implementation shall also conform to the notifications identified in [OMF] and specified in [DMI].

Dependent conformance to the State Management Function required by the General Log Control Conformance Category requires support for the state change reporting elements of procedure in clause 11.1 of [STMF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.3.2 of these agreements and all clauses referenced by clause 5.3.2. An implementation shall also conform to the notifications identified in [STMF] and specified in [DMI].

Dependent conformance to the Alarm Reporting Function required by the General Log Control Conformance Category requires support for the alarm reporting elements of procedure in clause 11.1 of [ARF] in either the agent role, the manager role, or both roles as specified by the implementor in the PICS. In addition, an implementation shall conform to clause 5.5.2 of these agreements and all clauses referenced by clause 5.5.2. An implementation shall also conform to the notifications identified in [ARF] and specified in [DMI].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the PT–SET, PT–GET, PT–CREATE, PT–DELETE, object creation reporting, object deletion reporting, and attribute value change reporting elements of procedure as defined in clauses 11.1.1 through 11.1.7 of [OMF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.5 of [OMF].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the state change reporting elements of procedure as defined in clause 11.1 of [STMF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.6 of [STMF].

For each role claimed to be supported in the PICS, an implementation shall support the transfer syntax derived from the encoding rules defined in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs required to support that portion of the "CMIP–PCI" abstract syntax defined in [CMIP] required to support the alarm reporting elements of procedure as defined in clause 11.1 of [ARF].

The implementation shall support the transfer syntax derived from the encoding rules specified in [BER] named [joint–iso–ccitt asn1(1) basic encoding(1)], for the purpose of generating and interpreting the MAPDUs defined by the abstract data types referenced in 11.2.5 of [ARF].

8.3.3 Management Information Conformance Category

To be conformant to the Management Information Conformance Category, an implementation shall include at least one managed object defined as specified by clause 7. The requirements for managing this managed object shall not conflict with the specifications in clauses 5 and 6. Managed object class definitions shall be provided either in full or by reference. Registered object identifiers shall be associated with any such managed object class definition and supporting definitions (e.g., attributes, name bindings). All mandatory abstract syntaxes and semantics associated with those identifiers shall be used. Note that all managed objects and supporting definitions in Annex A satisfy these conformance requirements.

An implementation is conformant to a managed object class definition if it supports all the mandatory packages specified in the managed object class as well as all associated information (e.g., attributes, notifications, actions, parameters) referenced in these packages and at least one name binding that may be used to support the naming of instances of this managed object class. Although it is not necessary to be conformant to all superior object classes in the containment tree of an instance of a conformant managed object class, all name bindings and naming attributes necessary to access that object instance shall be publicly available.

8.3.3.1 MOCS Proforma

The implementor shall provide a statement specifying which managed object classes are supported. A MOCS proforma shall be completed by the implementor for each managed object class supported.

Editor's Note: [The CD Version of ISO/IEC 10165–6 (Requirements and Guidelines for Implementation Conformance Statement Proformas Associated with Management Information) is now available. MOCS Proformas for each managed object class supported should be developed consistent with 10165–6 [MOCS].]

For each managed object class supported, the following shall be supplied:

a) a statement of pragmatic constraints (e.g., attribute values/ranges, initial values) supported, unless such constraints are defined in the managed object class definition;

b) a statement of conditional packages supported;

c) a statement of role(s) (manager, agent, or both) in which the object class definition is supported.

Editor's Note: [CD 10165–6 does not currently distinguish roles.]

8.3.4 Management Application Contexts

The implementation shall support at least the application context for systems management defined in ISO/IEC 10040 [SMO].

Note: **[**Such a statement is required by [SMO] clause 7.2.**]**

Note: **[**Such a statement is required by part 5, clause 13.7, which discusses conformance requirements for these IAs, as related to services required of the upper layers and other ASEs.**]**

8.4 Demonstration of Conformance

(Refer to the Working Implementation Agreements Document.)

The purpose of this clause is to establish requirements for environments needed to demonstrate conformance. In general, to test management implementations, a combination of management communication, management functions and services and management information must be installed in a system under test. For example, to demonstrate managed object class definition conformance, management communications must be supported. Likewise, to demonstrate communications conformance, a MIB configuration must be supported.

8.4.1 Management Communication

(Refer to the Working Implementation Agreements Document.)

To demonstrate conformance to the Management Communication General Conformance Category claimed to satisfy clause 8.3.1, the system must demonstrate conformance to either AOM11 or AOM12. To demonstrate conformance to AOM11, a system shall contain object(s) that can be addressed in such a way that all CMIP kernel functional unit capability can be demonstrated. To demonstrate conformance to AOM12, a system shall contain a MIB configuration that has some type of tree hierarchy to demonstrate scoping and filtering capabilities. An additional requirement for demonstrating conformance to AOM12 is that an implementation of the managed objects must support the capabilities to exercise the full functionality of AOM12 (i.e., kernel, multiple object selection, multiple reply, filter and cancel GET).

8.4.2 Management Functions and Services

(Refer to the Working Implementation Agreements Document.)

Conformance to the Management Information Conformance Category is provided through conformance to managed objects. To demonstrate conformance to the supported managed objects, the system shall support the conditions in clause 8.4.1 (Management Communication).

For conformance to an object supported in the Agent role, the implementation shall demonstrate that all appropriate CMIS operations and modify operations for the defined objects and attributes which are claimed to be supported in the MOCS, are, in fact, supported.

For conformance to an object supported in the Manager role, the implementation shall demonstrate the ability to receive PDUs from and transmit PDUs to an object instantiation for all PDUs, attributes and functions claimed to be supported in the MOCS.

8.4.3 Management Information

(Refer to the Working Implementation Agreements Document.)

To demonstrate conformance to the Management Functions and Services Categories claimed to be supported in clause 8.3.2, the system must support the co–conditions in clauses 8.4.1 and 8.4.2. A system must also conform to the elements of procedure for the systems management services defined by the particular System Management Function (SMF) and the managed objects, attributes, and notifications defined by the SMF. An additional requirement for the demonstration of conformance to the Management Function and Services Conformance Category is the implementation of a managed object supporting the services claimed to be supported.

9 Management Ensembles

(Refer to the Working Implementation Agreements Document.)

9.1 Management Ensemble Concepts

(Refer to the Working Implementation Agreements Document.)

9.2 Management Ensemble Format

(Refer to the Working Implementation Agreements Document.)

9.2.1 Use of Boiler Plate Text

(Refer to the Working Implementation Agreements Document.)

10 Management Coexistence and Interworking

(Refer to the Working Implementation Agreements Document.)

10.1 Internet MIB Translation

(Refer to the Working Implementation Agreements Document.)

10.2 ISO/CCITT MIB Translation

(Refer to the Working Implementation Agreements Document.)

10.3 ISO/CCITT to Internet Management Proxy

(Refer to the Working Implementation Agreements Document.)

Annex A (informative)

Management Information Library (MIL)

(Refer to the Working Implementation Agreements Document for additional information.)

A.1 Introduction

(Refer to the Working Implementation Agreements Document.)

A.2 Rules and Procedures

(Refer to the Working Implementation Agreements Document.)

A.3 General Guidelines

(Refer to the Working Implementation Agreements Document.)

A.4 Harmonized Library

The definitions specified in this clause can be referenced by using the label "OP1 Library Vol. 1" (e.g., "OP1 Library Vol. 1":computerSystem).

By inclusion of the managed object (MO) definitions and the object identifiers in Annex A and Annex B, respectively, of the Stable Implementors' Agreements (SIAs), these managed object (MO) definitions have become formally registered. Implementors of part 18 of the SIAs do not have to support any of these MOs. However, even though Annex A and Annex B are informative annexes, any implementation that claims to conform to these definitions must treat these definitions as normative and comply with the relevant portions of Annex A.4 and A.5, and Annex B.

A.4.1 Managed Object Classes and Mandatory Packages

A.4.1.1 Computer System

```
computerSystem   MANAGED OBJECT CLASS

 DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;

 CHARACTERIZED BY  computerSystemPkg;

 CONDITIONAL PACKAGES
    peripheralNamePkg        PRESENT IF !an instance supports it and the
                                        peripheralListPkg is NOT present!,
    peripheralListPkg        PRESENT IF !an instance supports it and the
                                        peripheralNamePkg is NOT present!,
    processingEntityNamePkg  PRESENT IF !an instance supports it and the
                                        processingEntityListPkg is NOT present!,
    processingEntityListPkg  PRESENT IF !an instance supports it and the
                                        processingEntityNamePkg is NOT present!,
    systemTimePkg            PRESENT IF !an instance supports it!,
    upTimePkg                PRESENT IF !an instance supports it!,
    "Rec. M.3100 : 1992":userLabelPackage
                             PRESENT IF !an instance supports it!,
```

```
    usageStatePkg            PRESENT IF !resource can detect usage!;

REGISTERED AS  {x-objectClass 1};


computerSystemPkg              PACKAGE
     BEHAVIOUR computerSystemPkgDefinition,
               computerSystemPkgBehaviour;
     ATTRIBUTES
         computerSystemId GET,
         "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState  GET,
         "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState GET-REPLACE,
         "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus GET-REPLACE ADD-REMOVE,
         "Rec. X.721 | ISO/IEC 10165-2 : 1992":availabilityStatus GET;

     ATTRIBUTE GROUPS
         "Rec. X.721 | ISO/IEC 10165-2 : 1992":state
               "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState
               "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState
               "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus
               "Rec. X.721 | ISO/IEC 10165-2 : 1992":availabilityStatus;

     NOTIFICATIONS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":stateChange,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":processingErrorAlarm,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":environmentalAlarm,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":equipmentAlarm;;

computerSystemPkgDefinition       BEHAVIOUR

DEFINED AS
        !The Computer System managed object class represents the aggregate of components which, when
        considered as a whole, is capable of performing data processing, storage, and retrieval functions.
        In order to perform its function, the computer system may have a variety of components including
        processing entities, terminals, disk drives, printers, etc.

        The Computer System is intended to represent an aggregation of other objects, and can model either
        self-contained computer systems or computer systems which are physically distributed, possibly over
        a wide geographical area.  An instance of the Computer System managed object class may have
        subordinate managed objects representing the individual entities within the computer system.
        Examples are entities such as disks, operating systems and processing entities.

        Since the Computer System may be physically distributed, it is not appropriate to model the computer
        system managed object class as a subclass of the Equipment managed object class (since Equipment
        implies a single physical location through its location attribute).  However, there can be cases
        where the Computer System is not physically distributed, in which case a Name Binding allowing
        Computer System to be named by OMNIPoint Equipment is permissible.

        It is not appropriate to model Computer System as a subclass of the DMI System managed object class.
        Unlike Computer System, the DMI System is a "container" object class which is instantiated in
        managed systems and exists mainly to name the managed and support objects it makes visible.!;


computerSystemPkgBehaviour  BEHAVIOUR

DEFINED AS
        !A value for the computerSystemId attribute can only be provided when the object is created.
        Furthermore, this attribute value may not change once the managed object has been instantiated.
        Thus, this attribute is never the subject of an AttributeValueChange Notification.

        Conditions under which an AttributeValueChange Notification is emitted are stated in the behaviour
        of the appropriate package or attribute.  In the absence of such a statement in the behaviour, the
        attribute does not cause an AttributeValueChange notification to be emitted.
        All attributeValueChange notifications shall include the Attribute Identifier List parameter.

        The stateChange notification is emitted when any of the following attributes change in value:
        administrativeState, operationalState, and availability status.

        The stateChange notification is not emitted when the alarmStatus attribute changes value.  (This is
        to avoid duplication of notifications.)

        Since every combination of state attribute values may not be appropriate for particular kinds of
        computer systems, only appropriate combinations need be supported.
```

```
    The processingErrorAlarm notification is emitted when the computerSystem resource experiences any of
    the alarm conditions defined by ISO/IEC 10164-4 (e.g., storage capacity problem, version mismatch,
    corrupt data, software error, underlying resource unavailable).!;
```

A.4.1.2 Connection Oriented Transport Protocol Layer Entity

```
coTransportProtocolLayerEntity    MANAGED OBJECT CLASS

       DERIVED FROM      "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;

       CHARACTERIZED BY  coTransportProtocolLayerEntityPkg;

       CONDITIONAL PACKAGES
                         manufacturerListPkg        PRESENT IF !an instance supports it and the
                                                                manufacturerNamePkg is NOT present!,
                         manufacturerNamePkg        PRESENT IF !an instance supports it and the
                                                                manufacturerListPkg is NOT present!,
                         productLabelPkg            PRESENT IF !an instance supports it!,
                         opVersionPkg               PRESENT IF !an instance supports it!,
                         serialNumberPkg            PRESENT IF !an instance supports it!,
                         typeTextPkg                PRESENT IF !an instance supports it!,
                         upTimePkg                  PRESENT IF !an instance supports it!,
                         incomingProtocolErrorPkg   PRESENT IF !an instance supports it!,
                         outgoingProtocolErrorPkg   PRESENT IF !an instance supports it!,
                         checksumPDUsDiscardedPkg   PRESENT IF !an instance supports it!,
                         maxPDUSizeIVPkg            PRESENT IF !the "OP1 Library Vol. 2 : 1992":transport
                                                    ConnectionIVMO object class is not used to
                                                    provide this initial value!,
                         usageStatePkg              PRESENT IF !resource can detect usage!;

REGISTERED AS {x-objectClass 2};


coTransportProtocolLayerEntityPkg   PACKAGE

       BEHAVIOUR  coTransportProtocolLayerEntityPkgDefinition,
                  coTransportProtocolLayerEntityPkgBehaviour;

       ATTRIBUTES
           coTransportProtocolLayerEntityId  PERMITTED VALUES SYNTAX-1.GraphicString64 GET,
           transportEntityType GET,
           localTransportAddresses  GET,
           activeConnections  PERMITTED VALUES SYNTAX-1.Integer32 GET,
           maxConnections   PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState  GET-REPLACE,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState  GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus GET-REPLACE ADD-REMOVE,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":outgoingConnectionRequestsCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingConnectionRequestsCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":outgoingConnectionRejectErrorCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingConnectionRejectErrorCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":outgoingDisconnectErrorCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingDisconnectErrorCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingDisconnectCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":outgoingDisconnectCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsSentCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsReceivedCounter
                                                PERMITTED VALUES SYNTAX-1.Integer32 GET;

       ATTRIBUTE GROUPS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":state
                       "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState
                       "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState
                       "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus;

       ACTIONS    activate, deactivate;
```

```
       NOTIFICATIONS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":stateChange,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":processingErrorAlarm;;


coTransportProtocolLayerEntityPkgDefinition  BEHAVIOUR

DEFINED AS
        !The coTransportProtocolLayerEntity managed object class represents an instantiation  of any
        connection-oriented transport layer protocol (e.g., the ISO Transport Protocol   layer or the
        Internet Transmission Control Protocol (TCP) Layer).  The transport   protocol layer is layer four
        of the OSI Reference model.  It provides for the   transparent transference of data between two peer
        entities.  It relieves its users   from any concerns about the detailed way in which supporting
        communication media are   utilized to achieve this transfer.  The connection-oriented transport
        protocol layer   entity makes use of a transport connection for the purpose of transferring data.

        This is a generally applicable managed object class, in that it does not represent any specific
        connection-oriented transport protocol - rather it contains characteristics common across various
        different connection-oriented transport layer protocols.  This managed object class is not intended
        to override any transport layer managed object classes defined in ISO.  It provides a high level
        view of a connection-oriented transport layer protocol and complements the protocol-specific views
        being defined in the standards.!;


coTransportProtocolLayerEntityPkgBehaviour  BEHAVIOUR

DEFINED AS
        !Conditions under which an attributeValueChange notification is emitted are stated in the behaviour
        of the appropriate package or attribute.  In the absence of such a statement, in the behaviour, the
        attribute does not cause an attributeValueChange to Be emitted.

        The attributeValueChange notification is emitted when any of the following attributes change in
        value: localTransportAddresses, maxConnections, transportEntityType, and all counter attributes
        (only when they wrap).  All attributeValueChange notifications shall include the Attribute
        Identifier List parameter.  All attributeValueChange notifications which report counter attribute
        wraps shall contain the maximum counter attribute value in the Old Attribute Value parameter.

        The stateChange notification is emitted when any of the following attributes change in value:
        administrativeState and operationalState.

        The processingErrorAlarm notification is emitted when the coTransportProtocolLayerEntity resource
        experiences any of the alarm conditions defined by ISO/IEC 10164-4 (e.g., storage capacity problem,
        version mismatch, corrupt data, software error, underlying resource unavailable).!;
```

This is a generally applicable managed object class, in that it does not represent any specific connection-oriented transport protocol. ISO/IEC 10733 [TLM] defines specific objects for managing OSI transport protocol layer entities.

A.4.1.3 Connectionless Network Protocol Layer Entity

```
clNetworkProtocolLayerEntity    MANAGED OBJECT CLASS

       DERIVED FROM      "Rec. X.721 | ISO/IEC 10165-2 : 1992": top;

       CHARACTERIZED BY  clNetworkProtocolLayerEntityPkg;

       CONDITIONAL PACKAGES
                         manufacturerListPkg  PRESENT IF !an instance supports it and the
                                                         manufacturerNamePkg is NOT present!,
                         manufacturerNamePkg  PRESENT IF !an instance supports it and the
                                                         manufacturerListPkg is NOT present!,
                         productLabelPkg      PRESENT IF !an instance supports it!,
                         opVersionPkg         PRESENT IF !an instance supports it!,
                         serialNumberPkg      PRESENT IF !an instance supports it!,
                         typeTextPkg          PRESENT IF !an instance supports it!,
                         upTimePkg            PRESENT IF !an instance supports it!;

REGISTERED AS  {x-objectClass 3};


clNetworkProtocolLayerEntityPkg   PACKAGE

        BEHAVIOUR  clNetworkProtocolLayerEntityPkgDefinition,
                   clNetworkProtocolLayerEntityPkgBehaviour;
```

```
   ATTRIBUTES
      clNetworkProtocolLayerEntityId  PERMITTED VALUES SYNTAX-1.GraphicString64 GET,
      networkEntityType GET,
      localNetworkAddresses  GET-REPLACE ADD-REMOVE,
      nPDUTimeToLive  PERMITTED VALUES SYNTAX-1.Integer32 GET-REPLACE,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState  GET-REPLACE,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState  GET,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus  GET-REPLACE ADD-REMOVE,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusSentCounter
                                      PERMITTED VALUES SYNTAX-1.Integer32 GET,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusReceivedCounter
                                      PERMITTED VALUES SYNTAX-1.Integer32 GET,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsSentCounter
                                      PERMITTED VALUES SYNTAX-1.Integer32 GET,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsReceivedCounter
                                      PERMITTED VALUES SYNTAX-1.Integer32 GET,
      pdusForwardedCounter  PERMITTED VALUES SYNTAX-1.Integer32 GET,
      pdusReasmbldOKCounter  PERMITTED VALUES SYNTAX-1.Integer32 GET,
      pdusReasmbldFailCounter  PERMITTED VALUES SYNTAX-1.Integer32 GET,
      pdusDiscardedCounter   PERMITTED VALUES SYNTAX-1.Integer32 GET;

  ATTRIBUTE GROUPS
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":state
                  "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState
                  "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState
                  "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmStatus;

  ACTIONS    activate, deactivate;

  NOTIFICATIONS
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":processingErrorAlarm,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":communicationsAlarm,
      "Rec. X.721 | ISO/IEC 10165-2 : 1992":stateChange;;


clNetworkProtocolLayerEntityPkgDefinition  BEHAVIOUR

DEFINED AS
        !The clNetworkProtocolLayerEntity managed object class represents an instantiation of a
        connectionless network protocol layer.  The network protocol layer provides network services for the
        transparent transfer of data between peer transport entities.  It relieves the transport protocol
        layer from the need to know anything about the underlying network technologies used to achieve data
        transfer.

        This is a generally applicable managed object class, in that it does not represent any specific
        connectionless network protocol; instead, it contains characteristics common across various
        different connectionless network layer protocols.  This managed object class is not intended to
        override any network layer managed object classes defined in ISO.  It provides a high level view of
        a connectionless network layer protocol and complements the protocol-specific views  being defined
        in the standards.

        An instance of this managed object class supports only one type of protocol.!;

clNetworkProtocolLayerEntityPkgBehaviour  BEHAVIOUR

DEFINED AS
        !Conditions under which an attributeValueChange notification is emitted are stated in the behaviour
        of the appropriate package or attribute.  In the absence of such a statement, in the behaviour, the
        attribute does not cause an attributeValueChange to be emitted.

        The attributeValueChange notification is emitted when any of the following attributes change in
        value: networkEntityType, localNetworkAddresses, nPDUTimeToLive, and all counter attributes (only
        when they wrap).  All attributeValueChange notifications shall include the Attribute Identifier List
        parameter.  All attributeValueChange notifications which report counter attribute wraps shall
        contain the maximum counter attribute value in the Old Attribute Value parameter.

        The stateChange notification is emitted when any of the following attributes change in value:
        administrativeState and operationalState.

        The communicationsAlarm notification is emitted when the clNetworkProtocolLayerEntity resource
        experiences any of the alarm conditions defined by ISO/IEC 10164-4 (e.g., loss of signal, local
        transmission error, remote transmission error).  In particular, this notification is used to report
        when a data NPDU is discarded for any reason other than network congestion.
```

```
        The processingErrorAlarm notification is emitted when the clNetworkProtocolLayerEntity resource
        experiences any of the alarm conditions defined by ISO/IEC 10164-4 (e.g., storage capacity problem,
        version mismatch, corrupt data, software error, underlying resource unavailable).!;
```

This is a generally applicable managed object class, in that it does not represent any specific connectionless network protocol. ISO/IEC 10737 [NLM] defines specific objects for managing OSI network protocol layer entities.

A.4.1.4 OMNIPoint Equipment

```
-- This definition is subclassed from CCITT M.3100 Equipment, adding the following items:
--
--      Mandatory AttributeChange, ObjectCreation, ObjectDeletion Notifications
--      Mandatory Environmental, Processing Error, and Equipment Alarm Notifications
--      Mandatory Administrative and Operational State Attributes and State Change Notification
--      CREATE/DELETE operations and behaviours (in name bindings)
--      Conditional Contact, Customer, Function, Manufacturer, OMNIPoint Network, Service, Software
--      and Vendor Name and List Packages
--      Conditional Product and Serial Number Packages
--      Conditional Type Text Package
--      Conditional Location Pointer Package
--
-- ANSI T1M1.5 concerns regarding physical vs. functional modelling were resolved by excluding the Forum
-- R1 Equipment Type attribute from the OMNIPoint definition.  The TypeText, FunctionName, and/or
-- FunctionList attributes may be used to carry (as graphic strings or pointers) information concerning
-- the function(s) supported by the physical Equipment.  It is expected that Forum R1 to OMNIPoint 1
-- mapping rules will define a translation between Forum R1 EquipmentType enumerations and these OMNIPoint
-- Equipment attributes.


opEquipment      MANAGED OBJECT CLASS

 DERIVED FROM  "Rec. M.3100 : 1992":equipment;

 CHARACTERIZED BY
         opEquipmentPkg,
         "Rec. M.3100 : 1992":createDeleteNotificationsPackage,
         "Rec. M.3100 : 1992":attributeValueChangeNotificationPackage,
         "Rec. M.3100 : 1992":stateChangeNotificationPackage,
         "Rec. M.3100 : 1992":administrativeOperationalStatesPackage,
         "Rec. M.3100 : 1992":environmentalAlarmPackage,
         "Rec. M.3100 : 1992":processingErrorAlarmPackage,
         "Rec. M.3100 : 1992":equipmentsEquipmentAlarmPackage;

 CONDITIONAL PACKAGES
         contactListPkg       PRESENT IF !an instance supports it and the
                                         contactNamePkg is NOT present!,
         contactNamePkg       PRESENT IF !an instance supports it and the
                                         contactListPkg is NOT present!,
         customerListPkg      PRESENT IF !an instance supports it and the
                                         customerNamePkg is NOT present!,
         customerNamePkg      PRESENT IF !an instance supports it and the
                                         customerListPkg is NOT present!,
         functionListPkg      PRESENT IF !an instance supports it and the
                                         functionNamePkg is NOT present!,
         functionNamePkg      PRESENT IF !an instance supports it and the
                                         functionListPkg is NOT present!,
         locationPointerPkg   PRESENT IF !an instance supports it and the
                                         "Rec. M.3100 : 1992":
                                         locationNamePackage is NOT present!,
         manufacturerListPkg  PRESENT IF !an instance supports it and the
                                         manufacturerNamePkg is NOT present!,
         manufacturerNamePkg  PRESENT IF !an instance supports it and the
                                         manufacturerListPkg is NOT present!,
         opNetworkListPkg     PRESENT IF !an instance supports it and the
                                         opNetworkNamePkg is NOT present!,
         opNetworkNamePkg     PRESENT IF !an instance supports it and the
                                         opNetworkListPkg is NOT present!,
         opVersionPkg         PRESENT IF !"Rec. M.3100 : 1992":
                                         versionPackage is also present!,
         productLabelPkg      PRESENT IF !an instance supports it!,
         serialNumberPkg      PRESENT IF !an instance supports it!,
         serviceListPkg       PRESENT IF !an instance supports it and the
                                         serviceNamePkg is NOT present!,
         serviceNamePkg       PRESENT IF !an instance supports it and the
                                         serviceListPkg is NOT present!,
         softwareListPkg      PRESENT IF !an instance supports it and the
                                         softwareNamePkg is NOT present!,
```

```
        softwareNamePkg      PRESENT IF !an instance supports it and the
                                        softwareListPkg is NOT present!,
        typeTextPkg          PRESENT IF !an instance supports it!,
        usageStatePkg        PRESENT IF !resource can detect usage!,
        vendorListPkg        PRESENT IF !an instance supports it and the
                                        "Rec. M.3100 : 1992":
                                        vendorNamePackage is NOT present!;

REGISTERED AS  {x-objectClass 4};

     opEquipmentPkg        PACKAGE

       BEHAVIOUR opEquipmentPkgBehaviour;
             --  opEquipmentPkgDefinition inherited from Rec. M.3100 Equipment

       ATTRIBUTES
           "Rec. M.3100 : 1992":equipmentId PERMITTED VALUES SYNTAX-1.EquipmentIdRange GET;

       ATTRIBUTE GROUPS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":state
                 "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeState
                 "Rec. X.721 | ISO/IEC 10165-2 : 1992":operationalState;;


opEquipmentPkgBehaviour   BEHAVIOUR

DEFINED AS -- inherited from Rec. M.3100 Equipment, with the following extensions:
        !A value for the "Rec. M.3100 : 1992":equipmentId attribute can only be provided when the object is
        created.  Furthermore, this attribute value may not change once the managed object has been
        instantiated.  Thus, this attribute is never the subject of an AttributeValueChange Notification.

        Conditions under  which an AttributeValueChange Notification is emitted are stated in the behaviour
        of the appropriate package or attribute.  In the absence of such a statement in the behaviour, the
        attribute does not cause an AttributeValueChange notification to be emitted.
        All attributeValueChange notifications shall include the Attribute Identifier List parameter.

        The processingErrorAlarm notification (if present) is emitted when the Equipment resource
        experiences any of the alarm conditions defined by ISO/IEC 10164-4 (e.g., storage capacity problem,
        version mismatch, corrupt data, software error, underlying resource unavailable).!;
```

A.4.1.5 OMNIPoint Network

```
-- This definition is subclassed from Rec. M.3100 Network, adding the following items:
--
--      Network Title and associated name binding to Root
--      AttributeChange, ObjectCreation, ObjectDeletion Notifications
--      CREATE/DELETE operations and behaviours (in name bindings)


opNetwork       MANAGED OBJECT CLASS

 DERIVED FROM  "Rec. M.3100 : 1992":network;

 CHARACTERIZED BY opNetworkPkg;

REGISTERED AS  {x-objectClass 5};


   opNetworkPkg          PACKAGE

       BEHAVIOUR opNetworkPkgBehaviour;
             -- opNetworkPkgDefinition inherited from Rec. M.3100 Network

       ATTRIBUTES
           networkTitle       GET;

       NOTIFICATIONS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange;;


opNetworkPkgBehaviour   BEHAVIOUR

DEFINED AS -- inherited from Rec. M.3100 Network, with the following extensions:
        !Values for the Network Identifier and Network Title attributes can only be provided when the object
        is created.  Furthermore, these attribute values may not change once the managed object has been
```

instantiated. Thus, they are never the subject of an AttributeValueChange Notification. When NetworkTitle is used for naming, the Network Identifier attribute has a NULL value.

Conditions under which an AttributeValueChange Notification is emitted are stated in the behaviour of the appropriate package or attribute. In the absence of such a statement in the behaviour, the attribute does not cause an AttributeValueChange notification to be emitted. All attributeValueChange notifications shall include the Attribute Identifier List parameter.!;

A.4.1.6 Processing Entity

```
processingEntity       MANAGED OBJECT CLASS

 DERIVED FROM  opEquipment;

 CHARACTERIZED BY  processingEntityPkg;

 CONDITIONAL PACKAGES
        addressingPkg        PRESENT IF !relevant to the underlying resource!,
        cpuUtilizationPkg    PRESENT IF !an instance supports it!,
        memorySizePkg        PRESENT IF !relevant to the underlying resource!,
        memoryUtilizationPkg PRESENT IF !an instance supports it!,
        upTimePkg            PRESENT IF !an instance supports it!;

REGISTERED AS  {x-objectClass 6};

processingEntityPkg PACKAGE

    BEHAVIOUR processingEntityPkgDefinition,
              processingEntityPkgBehaviour;

    ATTRIBUTES
        cpuType      PERMITTED VALUES SYNTAX-1.GraphicString16 GET,
        osInfo       PERMITTED VALUES SYNTAX-1.OsInfoRange GET;;

processingEntityPkgDefinition         BEHAVIOUR

DEFINED AS
```

!The processingEntity managed object class represents the physical portion of the computer system that performs a processing function, frequently called a Central Processing Unit (CPU). A Processing Entity may be composed of such components as arithmetical logical units (ALU), registers for processing memory, limited storage most often in the form of Random Access Memory (RAM), and various other types of memory used in the processing function. It does not include such components as disk drives, data bases, etc.

Some Processing Entities may have input/output channels, particularly when hardware is shared between elements of the Processing Entity. In other cases, the input/output must be seen as components of a superior managed object, for example a Computer System, or as OMNIPoint Equipment objects shared among several Computer Systems.

The cpuType attribute indicates the type of central processor unit found in the Processing Entity.

The osInfo attribute specifies the names and releases of the supported operating systems.!;

```
processingEntityPkgBehaviour          BEHAVIOUR

DEFINED AS
        !The AttributeValueChange notification is emitted when any of the following attributes change in
        value: cpuType or osInfo.!;
```

A.4.1.7 Transport Connection

```
transportConnection          MANAGED OBJECT CLASS

   DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992": top;

   CHARACTERIZED BY   transportConnectionPkg;

   CONDITIONAL PACKAGES
                  maxRetransmissionsPkg               PRESENT IF !an instance supports it!,
                  retransmissionTimePkg               PRESENT IF !an instance supports it!,
                  retransmissionTimerInitialValuePkg  PRESENT IF !an instance supports it!,
                  pdusRetransmittedCounterPkg         PRESENT IF !an instance supports it!,
                  octetsRetransmittedPkg              PRESENT IF !an instance supports it!,
                  pdusRetransmittedThresholdPkg       PRESENT IF !an instance supports it!,
```

```
                    outgoingProtocolErrorPkg           PRESENT IF !an instance supports it!,
                    checksumPDUsDiscardedPkg           PRESENT IF !an instance supports it!;
REGISTERED AS {x-objectClass 7};


transportConnectionPkg   PACKAGE

       BEHAVIOUR  transportConnectionPkgDefinition,
                  transportConnectionPkgBehaviour;

       ATTRIBUTES
           transportConnectionId  PERMITTED VALUES SYNTAX-1.GraphicString64 GET,
           localTransportConnectionEndpoint GET,
           remoteTransportConnectionEndpoint GET,
           transportConnectionReference  PERMITTED VALUES SYNTAX-1.Integer32 GET,
           localNetworkAddress  GET,
           remoteNetworkAddress  GET,
           inactivityTimeout  PERMITTED VALUES SYNTAX-1.Integer32 GET,
           inactivityTime PERMITTED VALUES SYNTAX-1.Integer32 GET,
           maxPDUSize  PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusSentCounter
                                             PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusReceivedCounter
                                             PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsSentCounter
                                             PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsReceivedCounter
                                             PERMITTED VALUES SYNTAX-1.Integer32 GET,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingProtocolErrorCounter
                                             PERMITTED VALUES SYNTAX-1.Integer32 GET;
       NOTIFICATIONS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion transportDisconnectCause,
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange;;


transportConnectionPkgDefinition  BEHAVIOUR

DEFINED AS
        !The transportConnection managed object class represents an active transport connection (e.g., an
        OSI transport connection or a TCP connection).  A transport connection is established and used by
        two peer connection oriented transport protocol layer entities for the purpose of transferring data.
        A connection oriented transport protocol layer entity may support multiple transport connections.

        This is a generally applicable managed object class, in that it does not represent any specific
        connection-oriented transport protocol; rather it contains characteristics common across various
        different connection-oriented transport layer protocols.  This managed object class is not intended
        to override any transport layer managed object classes defined in ISO.  It provides a high level
        view of a connection-oriented transport layer protocol and complements the protocol-specific views
        being defined in the standards.!;


transportConnectionPkgBehaviour  BEHAVIOUR

DEFINED AS
        !An instance of the Transport Connection managed object class is created automatically in response
        to normal operation of the network.  A prerequisite to the creation of a transport connection is the
        existence of a transport entity (e.g. an instance of the Connection Oriented Transport Protocol
        Layer Entity) on the open system. When a new Transport Connection instance is created, the "OP1
        Library Vol. 2 : 1992":transportConnectionIVMO instance with the same superior may be used to
        provide initial attribute values for the new instance.  Alternatively, the Maximum PDU Size
        attribute takes on the value of the Maximum PDU Size attribute specified in the superior Transport
        Protocol Layer Entity managed object instance. Subsequently the Maximum PDU Size attribute may take
        on another value which applies specifically to the connection represented by the instantiation of
        the transport connection.  This change may occur as the result of peer protocol negotiation.

        The Additional Information parameter of the objectDeletion notification may optionally contain a
        management extension (as defined in DMI) whose identifier is that of the "cause" attribute, whose
        significance is FALSE, and whose information is "cause" as defined in the associated PARAMETER
        template.

        Conditions under which an attributeValueChange notification is emitted are stated in the behaviour
        of the appropriate package or attribute.  In the absence of such a statement, in the behaviour, the
        attribute does not cause an attributeValueChange to be emitted.

        The attributeValueChange notification is emitted when any of the following attributes change in
        value: inactivityTimeout, maxPDUSize, and all counter attributes (only when they wrap).  All
        attributeValueChange notifications shall include the Attribute Identifier List parameter.  All
```

```
        attributeValueChange notifications which report counter attribute wraps shall contain the maximum
        counter attribute value in the Old Attribute Value parameter.

        Transport Connection will delete itself when the value of the inactivityTime attribute equals that
        of the inactivityTimeout attribute.!;
```

This is a generally applicable managed object class, in that it does not represent any specific connection-oriented transport protocol. ISO/IEC 10733 [TLM] defines specific objects for managing OSI transport protocol layer entities.

A.4.2 Conditional Packages

A.4.2.1 Addressing Package

```
addressingPkg     PACKAGE

  BEHAVIOUR    addressingPkgDefinition,
               addressingPkgBehaviour;

  ATTRIBUTES   addressingSize   PERMITTED VALUES SYNTAX-1.AddressingSizeRange GET,
               endianess        GET;
REGISTERED AS {x-package 1};

addressingPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package defines the addressing size and endianess which are characteristic of the underlying
        resource.!;

addressingPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the addressingSize or endianess attributes change
        value.!;
```

A.4.2.2 Checksum PDUs Discarded Package

```
checksumPDUsDiscardedPkg  PACKAGE

       BEHAVIOUR  checksumPDUsDiscardedPkgDefinition,
                  checksumPDUsDiscardedPkgBehaviour;
       ATTRIBUTES
                  checksumPDUsDiscardedCounter PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 2};


checksumPDUsDiscardedPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying resource to count the number of well-formed
        PDUs rejected by the peer entity due to a checksum error.!;


checksumPDUsDiscardedPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the checksumPDUsDiscarded attribute wraps.!;
```

A.4.2.3 Contact List Package

```
contactListPkg     PACKAGE

  BEHAVIOUR contactListPkgDefinition,
            contactListPkgBehaviour;

  ATTRIBUTES   contactList  PERMITTED VALUES SYNTAX-1.AnyNamesRange  GET-REPLACE  ADD-REMOVE;

REGISTERED AS {x-package 3};
```

```
contactListPkgDefinition  BEHAVIOUR

DEFINED AS
        !The Contact List Attribute identifies who (person or  organization) should be contacted about the
        resource.!;

contactListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the contactList attribute changes value.!;
```

A.4.2.4 Contact Name Package

```
contactNamePkg     PACKAGE

  BEHAVIOUR contactNamePkgDefinition,
            contactNamePkgBehaviour;

  ATTRIBUTES   contactName  PERMITTED VALUES SYNTAX-1.AnyNameRange  GET-REPLACE;

REGISTERED AS {x-package 4};


contactNamePkgDefinition  BEHAVIOUR

DEFINED AS
        !The Contact Name Attribute identifies who (person or  organization) should be contacted about the
        resource.!;

contactNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the contactName attribute changes value.!;
```

A.4.2.5 CPU Utilization Package

```
cpuUtilizationPkg     PACKAGE

  BEHAVIOUR    cpuUtilizationPkgBehaviour;

  ATTRIBUTES   cpuUtilization   PERMITTED VALUES SYNTAX-1.PercentageRange
                                GET; --  changed from GET-REPLACE (Forum)
REGISTERED AS {x-package 5};

cpuUtilizationPkgBehaviour  BEHAVIOUR

DEFINED AS
        !Even if the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is NOT emitted when the cpuUtilization attribute changes value.!;
```

A.4.2.6 Customer List Package

```
customerListPkg     PACKAGE

  BEHAVIOUR customerListPkgDefinition,
            customerListPkgBehaviour;

  ATTRIBUTES  customerList PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS {x-package 6};

customerListPkgDefinition  BEHAVIOUR

DEFINED AS
        !The Customer List attribute identifies any customers that  are users of the resource.!;

customerListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the customerList attribute changes value.!;
```

A.4.2.7 Customer Name Package

```
customerNamePkg     PACKAGE

  BEHAVIOUR customerNamePkgDefinition,
            customerNamePkgBehaviour;

  ATTRIBUTES  customerName PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS {x-package 7};

customerNamePkgDefinition  BEHAVIOUR

DEFINED AS
        !The Customer Name attribute identifies any customer that is a user of the resource.!;

customerNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the customerName attribute changes value.!;
```

A.4.2.8 Function List Package

```
functionListPkg    PACKAGE

  BEHAVIOUR functionListPkgDefinition,
            functionListPkgBehaviour;

  ATTRIBUTES functionList PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS {x-package 8};

functionListPkgDefinition  BEHAVIOUR

DEFINED AS
        !The functionList attribute identifies those functions  provided by this resource.!;

functionListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the functionList attribute changes value.!;
```

A.4.2.9 Function Name Package

```
functionNamePkg    PACKAGE

  BEHAVIOUR functionNamePkgDefinition,
            functionNamePkgBehaviour;

  ATTRIBUTES functionName PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS {x-package 9};

functionNamePkgDefinition  BEHAVIOUR

DEFINED AS
        !The functionName attribute identifies the function provided by this resource.!;

functionNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the functionName attribute changes value.!;
```

A.4.2.10 Incoming Protocol Error Package

```
incomingProtocolErrorPkg  PACKAGE

       BEHAVIOUR  incomingProtocolErrorPkgDefinition,
                  incomingProtocolErrorPkgBehaviour;

       ATTRIBUTES
```

```
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":incomingProtocolErrorCounter
                                                  PERMITTED VALUES SYNTAX-1.Integer32 GET;
REGISTERED AS  {x-package 10};


incomingProtocolErrorPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying resource to count the number of incoming
        protocol errors detected.!;


incomingProtocolErrorPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the incomingProtocolErrorCounter attribute wraps.!;
```

A.4.2.11 Location Pointer Package

```
locationPointerPkg   PACKAGE
       BEHAVIOUR   locationPointerPkgDefinition,
                   locationPointerPkgBehaviour;

       ATTRIBUTES
           locationPointer  GET-REPLACE;
REGISTERED AS      {x-package 11};


locationPointerPkgDefinition BEHAVIOUR

DEFINED AS
        !This package provides managed object instance information for a location (e.g., Hilo Hawaii USA).!;


locationPointerPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Location Pointer attribute changes value.!;
```

A.4.2.12 Manufacturer List Package

```
manufacturerListPkg   PACKAGE
       BEHAVIOUR   manufacturerListPkgDefinition,
                   manufacturerListPkgBehaviour;

       ATTRIBUTES
           manufacturerList  PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;
REGISTERED AS      {x-package 12};


manufacturerListPkgDefinition BEHAVIOUR

DEFINED AS
        !This package indicates information about the manufacturer(s) that manufactured the underlying
        resource!;


manufacturerListPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the ManufacturerList attribute changes value.!;
```

A.4.2.13 Manufacturer Name Package

```
manufacturerNamePkg   PACKAGE
       BEHAVIOUR   manufacturerNamePkgDefinition,
                   manufacturerNamePkgBehaviour;

       ATTRIBUTES
```

```
          manufacturerName  PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;
REGISTERED AS      {x-package 13};


manufacturerNamePkgDefinition BEHAVIOUR

DEFINED AS
        !This package indicates information about the manufacturer that manufactured the underlying
        resource!;


manufacturerNamePkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the ManufacturerName attribute changes value.!;
```

A.4.2.14 Max PDU Size IV Package

```
maxPDUSizeIVPkg  PACKAGE
       BEHAVIOUR   maxPDUSizeIVPkgDefinition,
                   maxPDUSizeIVPkgBehaviour;

       ATTRIBUTES
          maxPDUSize  PERMITTED VALUES SYNTAX-1.Integer32 GET-REPLACE;
REGISTERED AS      {x-package 14};


maxPDUSizeIVPkgDefinition BEHAVIOUR

DEFINED AS
        !This package provides the initial value for the maximum length of a PDU that can be supported by
        the local layer entity.!;

maxPDUSizeIVPkgBehaviour BEHAVIOUR

DEFINED AS
        !The Maximum TPDU Size attribute provides the initial value to be used by newly-instantiated
        subordinate Transport Connection managed object instances for the maximum TPDU size to be supported
        on that connection.!;
```

A.4.2.15 Max Retransmissions Package

```
maxRetransmissionsPkg  PACKAGE

       BEHAVIOUR  maxRetransmissionsPkgDefinition,
                  maxRetransmissionsPkgBehaviour;

       ATTRIBUTES
          maxRetransmissions  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 15};


maxRetransmissionsPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to count the
        maximum number of times a TPDU is to be retransmitted before the transport connection is aborted.!;


maxRetransmissionsPkgBehaviour  BEHAVIOUR

DEFINED AS
        !When a new Transport Connection instance is created containing this package, any "OP1 Library Vol.
        2 : 1992":transportConnectionRetransmissionIVMO instance with the same superior may be used to
        provide initial attribute values for the new instance.!;
```

A.4.2.16 Memory Size Package

```
memorySizePkg     PACKAGE
```

```
  BEHAVIOUR       memorySizePkgDefinition,
                  memorySizePkgBehaviour;

  ATTRIBUTES   memorySize   PERMITTED VALUES SYNTAX-1.MemorySizeRange  GET;

REGISTERED AS {x-package 16};

memorySizePkgDefinition  BEHAVIOUR

DEFINED AS
        !The memorySize attribute indicates, in kilobytes, the amount of memory available to a Processing
        Entity (irrespective of its current usage).!;
```

memorySizePkgBehaviour BEHAVIOUR

DEFINED AS
!If the AttributeValueChange notification is defined for the managed object class using this package, this notification is emitted when the memorySize attribute changes value.!;

A.4.2.17 Memory Utilization Package

```
memoryUtilizationPkg     PACKAGE

  BEHAVIOUR    memoryUtilizationPkgBehaviour;

  ATTRIBUTES   memoryUtilization   PERMITTED VALUES SYNTAX-1.PercentageRange
                                   GET; --  added in response to Bull comment
REGISTERED AS {x-package 17};

memoryUtilizationPkgBehaviour  BEHAVIOUR

DEFINED AS
        !Even if the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is NOT emitted when the memoryUtilization attribute changes value.!;
```

A.4.2.18 Octets Retransmitted Package

```
octetsRetransmittedPkg  PACKAGE

       BEHAVIOUR  octetsRetransmittedPkgDefinition,
                  octetsRetransmittedPkgBehaviour;

       ATTRIBUTES
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":octetsRetransmittedErrorCounter
                                                  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 18};


octetsRetransmittedPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to count the
        number of octets retransmitted.!;


octetsRetransmittedPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the octetsRetransmitted attribute wraps.!;
```

A.4.2.19 OMNIPoint Network List Package

```
opNetworkListPkg    PACKAGE

  BEHAVIOUR opNetworkListPkgDefinition,
            opNetworkListPkgBehaviour;
```

```
  ATTRIBUTES  opNetworkList  PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS {x-package 19};

opNetworkListPkgDefinition  BEHAVIOUR


DEFINED AS
        !The opNetworkList attribute indicates what networks use or are dependent on the resource.!;

opNetworkListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the opNetworkList attribute changes value.!;
```

A.4.2.20 OMNIPoint Network Name Package

```
opNetworkNamePkg    PACKAGE

  BEHAVIOUR opNetworkNamePkgDefinition,
            opNetworkNamePkgBehaviour;

  ATTRIBUTES  opNetworkName  PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS {x-package 20};

opNetworkNamePkgDefinition  BEHAVIOUR


DEFINED AS
        !The opNetworkName attribute indicates what network uses or is dependent on the resource.!;

opNetworkNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the opNetworkName attribute changes value.!;
```

A.4.2.21 OMNIPoint Version Package

```
opVersionPkg  PACKAGE  -- refinement of Rec. M.3100 versionPackage

       BEHAVIOUR   opVersionPkgDefinition,
                   opVersionPkgBehaviour;

       ATTRIBUTES
           "Rec. M.3100 : 1992":version
                                         PERMITTED VALUES SYNTAX-1.GraphicString16 GET-REPLACE;

REGISTERED AS  {x-package 21};



opVersionPkgDefinition BEHAVIOUR

DEFINED AS
        !This package reflects the release version of the underlying resource as an attribute, as defined by
        "Rec. M.3100 : 1992".!;


opVersionPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Version attribute changes value.!;
```

A.4.2.22 Outgoing Protocol Error Package

```
outgoingProtocolErrorPkg  PACKAGE
```

```
       BEHAVIOUR  outgoingProtocolErrorPkgDefinition,
                  outgoingProtocolErrorPkgBehaviour;

       ATTRIBUTES
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":outgoingProtocolErrorCounter
                                                  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 22};


outgoingProtocolErrorPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying resource to count the number of outgoing
        protocol errors detected.  Note that not all resources have this capability.!;


outgoingProtocolErrorPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the outgoingProtocolErrorCounter attribute wraps.!;
```

A.4.2.23 PDUs Retransmitted Counter Package

```
pdusRetransmittedCounterPkg  PACKAGE

       BEHAVIOUR  pdusRetransmittedCounterPkgDefinition,
                  pdusRetransmittedCounterPkgBehaviour;

       ATTRIBUTES
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusRetransmittedErrorCounter
                                                  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 23};


pdusRetransmittedCounterPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to count the
        number of PDUs retransmitted.!;


pdusRetransmittedCounterPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the PDUsRetransmittedCounter attribute wraps.!;
```

A.4.2.24 PDUs Retransmitted Threshold Package

```
pdusRetransmittedThresholdPkg  PACKAGE

       BEHAVIOUR  pdusRetransmittedThresholdPkgDefinition,
                  pdusRetransmittedThresholdPkgBehaviour;

       ATTRIBUTES
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":pdusRetransmittedErrorThreshold GET-REPLACE;
       NOTIFICATIONS
            "Rec. X.721 | ISO/IEC 10165-2 : 1992":communicationsAlarm;

REGISTERED AS  {x-package 24};


pdusRetransmittedThresholdPkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to threshold the
        number of PDUs retransmitted.!;


pdusRetransmittedThresholdPkgBehaviour  BEHAVIOUR

DEFINED AS
```

```
!When a new Transport Connection instance is created containing this package, any "OP1 Library Vol.
2 : 1992":transportConnectionRetransmissionIVMO instance with the same superior may be used to
provide initial attribute values for the new instance.

If the attributeValueChange notification is defined for the managed object class using this package,
this notification is emitted when the pdusRetransmittedThreshold attribute changes in value.!;
```

A.4.2.25 Peripheral List Package

```
peripheralListPkg  PACKAGE

       BEHAVIOUR   peripheralListPkgDefinition,
                   peripheralListPkgBehaviour;

       ATTRIBUTES
           peripheralList  PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS      {x-package 25};


peripheralListPkgDefinition BEHAVIOUR

DEFINED AS
        !The Peripheral List attribute identifies auxiliary devices that are used by the resource (e.g.,
        disk drives, tape drives, printers).!;


peripheralListPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Peripheral List attribute changes value.!;
```

A.4.2.26 Peripheral Name Package

```
peripheralNamePkg  PACKAGE

       BEHAVIOUR   peripheralNamePkgDefinition,
                   peripheralNamePkgBehaviour;

       ATTRIBUTES
           peripheralName  PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS      {x-package 26};


peripheralNamePkgDefinition BEHAVIOUR

DEFINED AS
        !The Peripheral Name attribute identifies an auxiliary device that is used by the resource (e.g.,
        disk drive, tape drive, printer).!;


peripheralNamePkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Peripheral Name attribute changes value.!;
```

A.4.2.27 Processing Entity List Package

```
processingEntityListPkg  PACKAGE

       BEHAVIOUR   processingEntityListPkgDefinition,
                   processingEntityListPkgBehaviour;

       ATTRIBUTES
           processingEntityList  PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS      {x-package 27};
```

```
processingEntityListPkgDefinition BEHAVIOUR

DEFINED AS
        !The Processing Entity List attribute identifies the processing entities which may be used by the
        containing object instance but which are not contained in it (i.e., processing entities which are
        shared among systems).!;


processingEntityListPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Processing Entity List attribute changes value.!;
```

A.4.2.28 Processing Entity Name Package

```
processingEntityNamePkg  PACKAGE

       BEHAVIOUR   processingEntityNamePkgDefinition,
                   processingEntityNamePkgBehaviour;

       ATTRIBUTES
           processingEntityName  PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS      {x-package 28};


processingEntityNamePkgDefinition BEHAVIOUR

DEFINED AS
        !The Processing Entity Name attribute identifies the processing entity which may be used by the
        containing object instance but which is not contained in it (i.e., processing entities which are
        shared among systems).!;


processingEntityNamePkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Processing Entity Name attribute changes value.!;
```

A.4.2.29 Product Label Package

```
productLabelPkg  PACKAGE

       BEHAVIOUR   productLabelPkgDefinition,
                   productLabelPkgBehaviour;

       ATTRIBUTES
           productLabel  PERMITTED VALUES SYNTAX-1.GraphicString32 GET-REPLACE;

REGISTERED AS      {x-package 29};


productLabelPkgDefinition BEHAVIOUR

DEFINED AS
        !This package allows the product number or identifying string (e.g., model number) of the underlying
        resource to be reflected as an attribute.!;


productLabelPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Product Label attribute changes value.!;
```

A.4.2.30 Retransmission Time Package

```
retransmissionTimePkg  PACKAGE
```

```
       BEHAVIOUR  retransmissionTimePkgDefinition,
                  retransmissionTimePkgBehaviour;

       ATTRIBUTES
           retransmissionTime PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 30};

retransmissionTimePkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to present its
        current retransmission timer value as an attribute.!;


retransmissionTimePkgBehaviour  BEHAVIOUR

DEFINED AS
        !When a new Transport Connection instance is created containing this package, the initial value of
        this attribute may be provided by the retransmissionTimerInitialValue attribute (if present in the
        new managed object instance).!;
```

A.4.2.31 Retransmission Timer Initial Value Package

```
retransmissionTimerInitialValuePkg  PACKAGE

       BEHAVIOUR  retransmissionTimerInitialValuePkgDefinition,
                  retransmissionTimerInitialValuePkgBehaviour;

       ATTRIBUTES
           retransmissionTimerInitialValue PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS  {x-package 31};

retransmissionTimerInitialValuePkgDefinition  BEHAVIOUR

DEFINED AS
        !This package reflects the capability of the underlying transport protocol resource to present its
        initial retransmission timer value as an attribute.!;


retransmissionTimerInitialValuePkgBehaviour  BEHAVIOUR

DEFINED AS
        !When a new Transport Connection instance is created containing this package, any "OP1 Library Vol.
        2 : 1992":transportConnectionRetransmissionIVMO instance with the same superior may be used to
        provide initial attribute values for the new instance.!;
```

A.4.2.32 Serial Number Package

```
serialNumberPkg  PACKAGE

       BEHAVIOUR   serialNumberPkgDefinition,
                   serialNumberPkgBehaviour;

       ATTRIBUTES
           serialNumber   PERMITTED VALUES SYNTAX-1.GraphicString32 GET-REPLACE;

REGISTERED AS      {x-package 32};



serialNumberPkgDefinition BEHAVIOUR

DEFINED AS
        !This package allows the serial number of the underlying resource to be reflected as an attribute.!;


serialNumberPkgBehaviour BEHAVIOUR

DEFINED AS
```

```
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Serial Number attribute changes value.!;
```

A.4.2.33 Service List Package

```
serviceListPkg    PACKAGE

  BEHAVIOUR serviceListPkgDefinition,
            serviceListPkgBehaviour;

  ATTRIBUTES   serviceList   PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE  ADD-REMOVE;

REGISTERED AS {x-package 33};

serviceListPkgDefinition  BEHAVIOUR

DEFINED AS
        !Service List attribute identifies any services that  are supported by the resource.!;

serviceListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the serviceList attribute changes value.!;
```

A.4.2.34 Service Name Package

```
serviceNamePkg    PACKAGE

  BEHAVIOUR serviceNamePkgDefinition,
            serviceNamePkgBehaviour;

  ATTRIBUTES   serviceName   PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS {x-package 34};

serviceNamePkgDefinition  BEHAVIOUR

DEFINED AS
        !Service Name attribute identifies any service that is supported by the resource.!;

serviceNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the serviceName attribute changes value.!;
```

A.4.2.35 Software List Package

```
softwareListPkg    PACKAGE

  BEHAVIOUR softwareListPkgDefinition,
            softwareListPkgBehaviour;

  ATTRIBUTES  softwareList PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS {x-package 35};

softwareListPkgDefinition  BEHAVIOUR

DEFINED AS
        !The Software List attribute identifies those software components that run on or are considered part
        of the resource.!;

softwareListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the softwareList attribute changes value.!;
```

A.4.2.36 Software Name Package

```
softwareNamePkg    PACKAGE
```

```
  BEHAVIOUR softwareNamePkgDefinition,
            softwareNamePkgBehaviour;

  ATTRIBUTES  softwareName PERMITTED VALUES SYNTAX-1.AnyNameRange GET-REPLACE;

REGISTERED AS {x-package 36};

softwareNamePkgDefinition  BEHAVIOUR

DEFINED AS
        !The Software Name attribute identifies the software component that runs on or are considered part
        of the resource.!;

softwareNamePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the softwareName attribute changes value.!;
```

A.4.2.37 System Time Package

```
systemTimePkg  PACKAGE

       BEHAVIOUR   systemTimePkgDefinition,
                   systemTimePkgBehaviour;

       ATTRIBUTES
           systemTime  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS      {x-package 37};


systemTimePkgDefinition BEHAVIOUR

DEFINED AS
        !This package records the current time clocked by the resource.!;


systemTimePkgBehaviour BEHAVIOUR

DEFINED AS
        !The attribute contained in this package is never the subject of an attribute value change
        notification.  Even if the AttributeValueChange notification is defined for the managed object class
        using this package, this notification is NOT emitted when the systemTime attribute changes value.!;
```

A.4.2.38 Type Text Package

```
typeTextPkg  PACKAGE

       BEHAVIOUR   typeTextPkgDefinition,
                   typeTextPkgBehaviour;

       ATTRIBUTES
           typeText   PERMITTED VALUES SYNTAX-1.GraphicString32 GET-REPLACE;

REGISTERED AS      {x-package 38};


typeTextPkgDefinition BEHAVIOUR

DEFINED AS
        !This package serves to supplement and refine individual managed object class attributes!;


typeTextPkgBehaviour BEHAVIOUR

DEFINED AS
        !If the attributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the Type Text attribute changes value.!;
```

A.4.2.39 Up Time Package

```
upTimePkg  PACKAGE

       BEHAVIOUR   upTimePkgDefinition,
                   upTimePkgBehaviour;

       ATTRIBUTES
           upTime  PERMITTED VALUES SYNTAX-1.Integer32 GET;

REGISTERED AS      {x-package 39};


upTimePkgDefinition BEHAVIOUR

DEFINED AS
        !This package records the elapsed time during which the underlying resource has been enabled.!;


upTimePkgBehaviour BEHAVIOUR

DEFINED AS
        !The attribute contained in this package is never the subject of an attribute value change
        notification.  Even if the AttributeValueChange notification is defined for the managed object class
        using this package, this notification is NOT emitted when the upTime attribute changes value.!;
```

A.4.2.40 Usage State Package

```
usageStatePkg   PACKAGE

       BEHAVIOUR   usageStatePkgDefinition,
                   usageStatePkgBehaviour;

       ATTRIBUTES
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":usageState  GET;

       ATTRIBUTE GROUPS
           "Rec. X.721 | ISO/IEC 10165-2 : 1992":state
                 "Rec. X.721 | ISO/IEC 10165-2 : 1992":usageState;

REGISTERED AS {x-package 40};


usageStatePkgDefinition  BEHAVIOUR

DEFINED AS
        !This package specifies the Usage State of the underlying resource, to be included in resources
        which are able to detect whether or not they are currently in use.!;


usageStatePkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the stateChange notification is defined for the managed object class using this package, this
        notification is emitted when the usageState attribute changes value.!;
```

A.4.2.41 Vendor List Package

```
vendorListPkg    PACKAGE

  BEHAVIOUR vendorListPkgDefinition,
            vendorListPkgBehaviour;

  ATTRIBUTES  vendorList PERMITTED VALUES SYNTAX-1.AnyNamesRange GET-REPLACE ADD-REMOVE;

REGISTERED AS {x-package 41};


vendorListPkgDefinition  BEHAVIOUR

DEFINED AS
        !The Vendor List attribute identifies the organization(s)  from which the resource was obtained
        (e.g., purchased,  leased, etc.)!;
```

```
vendorListPkgBehaviour  BEHAVIOUR

DEFINED AS
        !If the AttributeValueChange notification is defined for the managed object class using this
        package, this notification is emitted when the vendorList attribute changes value.!;
```

A.4.3 Name Bindings

A.4.3.1 Computer System Name Bindings

```
computerSystem-system       NAME BINDING
  SUBORDINATE OBJECT CLASS  computerSystem AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  "Rec. X.721 | ISO/IEC 10165-2 : 1992":system;
  WITH ATTRIBUTE  computerSystemId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 1};


computerSystem-opNetwork       NAME BINDING
  SUBORDINATE OBJECT CLASS  computerSystem AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS     opNetwork AND SUBCLASSES;
  WITH ATTRIBUTE  computerSystemId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 2};


computerSystem-computerSystem    NAME BINDING
  SUBORDINATE OBJECT CLASS  computerSystem AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS     computerSystem AND SUBCLASSES;
  WITH ATTRIBUTE  computerSystemId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 3};
```

A.4.3.2 CO Transport Protocol Layer Entity Name Bindings

```
coTransportProtocolLayerEntity-computerSystem  NAME BINDING
       SUBORDINATE OBJECT CLASS  coTransportProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS computerSystem AND SUBCLASSES;
       WITH ATTRIBUTE coTransportProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 4};

coTransportProtocolLayerEntity-system  NAME BINDING
       SUBORDINATE OBJECT CLASS  coTransportProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS
       "Rec. X.721 | ISO/IEC 10165-2 : 1992": system AND SUBCLASSES;
       WITH ATTRIBUTE coTransportProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 5};

coTransportProtocolLayerEntity-opEquipment  NAME BINDING
       SUBORDINATE OBJECT CLASS  coTransportProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS opEquipment AND SUBCLASSES;
       WITH ATTRIBUTE coTransportProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 6};
```

A.4.3.3 CL Network Protocol Layer Entity Name Bindings

```
clNetworkProtocolLayerEntity-computerSystem  NAME BINDING
       SUBORDINATE OBJECT CLASS clNetworkProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS computerSystem AND SUBCLASSES;
```

```
       WITH ATTRIBUTE clNetworkProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 7};

clNetworkProtocolLayerEntity-system  NAME BINDING
       SUBORDINATE OBJECT CLASS clNetworkProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS
       "Rec. X.721 | ISO/IEC 10165-2 : 1992": system AND SUBCLASSES;
       WITH ATTRIBUTE clNetworkProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 8};

clNetworkProtocolLayerEntity-opEquipment  NAME BINDING
       SUBORDINATE OBJECT CLASS clNetworkProtocolLayerEntity AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS opEquipment AND SUBCLASSES;
       WITH ATTRIBUTE clNetworkProtocolLayerEntityId;
REGISTERED AS  {x-nameBinding 9};
```

A.4.3.4 OMNIPoint Equipment Name Bindings

```
opEquipment-computerSystem       NAME BINDING
  SUBORDINATE OBJECT CLASS  opEquipment AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  computerSystem AND SUBCLASSES;
  WITH ATTRIBUTE  "Rec. M.3100 : 1992":equipmentId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 10};

opEquipment-system       NAME BINDING
  SUBORDINATE OBJECT CLASS  opEquipment AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  "Rec. X.721 | ISO/IEC 10165-2 : 1992":system;
  WITH ATTRIBUTE  "Rec. M.3100 : 1992":equipmentId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 11};

opEquipment-equipment       NAME BINDING
  SUBORDINATE OBJECT CLASS opEquipment AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  "Rec. M.3100 : 1992":equipment AND SUBCLASSES;
  WITH ATTRIBUTE  "Rec. M.3100 : 1992":equipmentId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 12};

opEquipment-opNetwork       NAME BINDING
  SUBORDINATE OBJECT CLASS  opEquipment AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS     opNetwork AND SUBCLASSES;
  WITH ATTRIBUTE  "Rec. M.3100 : 1992":equipmentId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 13};
```

A.4.3.5 OMNIPoint Network Name Bindings

```
-- The following name bindings are defined, in addition to those
-- inherited from Rec. M.3100 Network (which do not include CREATE/DELETE):

network-opNetwork-1       NAME BINDING
  SUBORDINATE OBJECT CLASS  opNetwork AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  "Rec. M.3100 : 1992":network AND SUBCLASSES;
  WITH ATTRIBUTE  "Rec. M.3100 : 1992":networkId;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 14};
```

```
network-opNetwork-2      NAME BINDING
  SUBORDINATE OBJECT CLASS  opNetwork AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS  "Rec. M.3100 : 1992":network AND SUBCLASSES;
  WITH ATTRIBUTE  networkTitle;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 15};


opNetwork-root       NAME BINDING
  SUBORDINATE OBJECT CLASS  opNetwork AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS  "Rec. X.660 | ISO/IEC 9834-1 : 1992":root;
  WITH ATTRIBUTE  networkTitle;
    CREATE  WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE  ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {x-nameBinding 16};
```

A.4.3.6 Processing Entity Name Bindings

```
-- processingEntity-opEquipment NAME BINDING
-- processingEntity-computerSystem NAME BINDING
-- both inherited from opEquipment, no additional bindings required.
```

A.4.3.7 Transport Connection Name Bindings

```
transportConnection-coTransportProtocolLayerEntity  NAME BINDING
       SUBORDINATE OBJECT CLASS transportConnection AND SUBCLASSES;
       NAMED BY
       SUPERIOR OBJECT CLASS coTransportProtocolLayerEntity AND SUBCLASSES;
       WITH ATTRIBUTE transportConnectionId;
       BEHAVIOUR transportConnectionNBBehaviour;
       DELETE  DELETES-CONTAINED-OBJECTS;
REGISTERED AS  {x-nameBinding 17};


transportConnectionNBBehaviour BEHAVIOUR

DEFINED AS
        !The expected real effect of the DELETE operation when applied to an instance of the transport
        connection managed object class is that the underlying transport connection resource is aborted.!;
```

A.4.4 Attributes

A.4.4.1 Active Connections

```
activeConnections          ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  activeConnectionsBehaviour;

REGISTERED AS     {x-attribute 1};

activeConnectionsBehaviour   BEHAVIOUR

DEFINED AS
        !The activeConnections attribute specifies the number of currently active transport connections
        (i.e., the number of transport connections which are in the open state [as defined for the
        underlying protocol machine], updated upon each connection establishment and release).!;
```

A.4.4.2 Addressing Size

```
addressingSize       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AddressingSizeBase;
```

```
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR  addressingSizeBehaviour;

REGISTERED AS {x-attribute 2};

addressingSizeBehaviour       BEHAVIOUR

DEFINED AS
        !The Addressing Size attribute indicates the number of bits which represent an address to the
        Processing Entity's central processing unit (CPU).!;
```

A.4.4.3 Checksum PDUs Discarded Counter

```
checksumPDUsDiscardedCounter   ATTRIBUTE

        DERIVED FROM
        "Rec. X.721 | ISO/IEC 10165-2 : 1992": counter;
        BEHAVIOUR  checksumPDUsDiscardedCounterBehaviour;

REGISTERED AS     {x-attribute 3};

checksumPDUsDiscardedCounterBehaviour   BEHAVIOUR

DEFINED AS
        !The attribute specifies the number of well-formed PDUs rejected by the peer entity due to a
        checksum error.!;
```

A.4.4.4 Computer System Id

```
computerSystemId  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.GraphicStringBase;
        MATCHES FOR  EQUALITY, SUBSTRINGS;
        BEHAVIOUR  computerSystemIdBehaviour;

REGISTERED AS     {x-attribute 4};

computerSystemIdBehaviour  BEHAVIOUR

DEFINED AS
        !The computerSystemId attribute is the distinguishing attribute for the computerSystem managed
        object class.!;
```

A.4.4.5 CL Network Protocol Layer Entity Id

```
clNetworkProtocolLayerEntityId  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.GraphicStringBase;
        MATCHES FOR  EQUALITY, SUBSTRINGS;
        BEHAVIOUR  clNetworkProtocolLayerEntityIdBehaviour;

REGISTERED AS     {x-attribute 5};

clNetworkProtocolLayerEntityIdBehaviour  BEHAVIOUR

DEFINED AS
        !The clNetworkProtocolLayerEntityId attribute is the distinguishing attribute for the
        clNetworkProtocolLayerEntity managed object class.!;
```

A.4.4.6 CO Transport Protocol Layer Entity Id

```
coTransportProtocolLayerEntityId  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.GraphicStringBase;
        MATCHES FOR  EQUALITY, SUBSTRINGS;
        BEHAVIOUR  coTransportProtocolLayerEntityIdBehaviour;

REGISTERED AS     {x-attribute 6};

coTransportProtocolLayerEntityIdBehaviour  BEHAVIOUR

DEFINED AS
        !The coTransportProtocolLayerEntityId attribute is the distinguishing attribute for the
        coTransportProtocolLayerEntity managed object class.!;
```

A.4.4.7 Contact List

```
contactList        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  contactListBehaviour;

REGISTERED AS {x-attribute 7};

contactListBehaviour        BEHAVIOUR

DEFINED AS
        !The Contact List attribute provides managed object instance information for one or more contacts.
        The following object classes (or any of their subclasses or allomorphic classes) are valid as
        contacts: "OP1 Library Vol. 4":Contact.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.8 Contact Name

```
contactName        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  contactNameBehaviour;

REGISTERED AS {x-attribute 8};

contactNameBehaviour        BEHAVIOUR

DEFINED AS
        !The Contact Name attribute provides information for one person or organization who can be
        contactacted about the resource.!;
```

A.4.4.9 CPU Type

```
cpuType       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.GraphicStringBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  cpuTypeBehaviour;

REGISTERED AS {x-attribute 9};

cpuTypeBehaviour       BEHAVIOUR

DEFINED AS
        !The Central Processor Unit (CPU) Type attribute indicates the type of central processor unit found
        in a Processing Entity.!;
```

A.4.4.10 CPU Utilization

```
cpuUtilization       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.IntegerBase;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR  cpuUtilizationBehaviour;

REGISTERED AS {x-attribute 10};

cpuUtilizationBehaviour       BEHAVIOUR

DEFINED AS
        !The cpuUtilization attribute specifies, as a percentage, the overall utilization of all central
        processor units found in a processing entity.  The percentage is expressed as an integer with
        permissible values in the range of 0 to 100.!;
```

A.4.4.11 Customer List

```
customerList        ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  customerListBehaviour;

REGISTERED AS {x-attribute 11};

customerListBehaviour        BEHAVIOUR

DEFINED AS
        !The Customer List attribute provides managed object instance information about one or more
        customers.  The following classes (or any of their subclasses or allomorphic classes) are valid as
        customers: "OP1 Library Vol. 4":Customer.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.12 Customer Name

```
customerName        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  customerNameBehaviour;

REGISTERED AS {x-attribute 12};

customerNameBehaviour        BEHAVIOUR

DEFINED AS
        !The Customer Name attribute provides information about one customer.!;
```

A.4.4.13 Endianess

```
endianess       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.Endianess;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR  endianessBehaviour;

REGISTERED AS {x-attribute 13};

endianessBehaviour       BEHAVIOUR

DEFINED AS
        !The Endianess attribute indicates the bit order (big endian, little endian) used by the Processing
        Entity's central processing unit (CPU).!;
```

A.4.4.14 Function List

```
functionList         ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  functionListBehaviour;

REGISTERED AS {x-attribute 14};

functionListBehaviour    BEHAVIOUR

DEFINED AS
        !The Function List attribute provides managed object instance information about one or more
        functions.  The following managed object classes (or any of their subclasses or allomorphic classes)
        are valid as functions:  "OP1 Library Vol. 4":Function.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.15 Function Name

```
functionName         ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  functionNameBehaviour;
```

```
REGISTERED AS {x-attribute 15};

functionNameBehaviour    BEHAVIOUR

DEFINED AS
        !The Function Name attribute provides information about one function.!;
```

A.4.4.16 Inactivity Time

```
inactivityTime           ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.HundredthsOfSec;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  inactivityTimeBehaviour;

REGISTERED AS      {x-attribute 16};

inactivityTimeBehaviour  BEHAVIOUR

DEFINED AS
        !This attribute specifies the amount of time (in 1/100ths of a second) that the transport connection
        has been inactive.!;
```

A.4.4.17 Inactivity Timeout

```
inactivityTimeout           ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.HundredthsOfSec;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  inactivityTimeoutBehaviour;

REGISTERED AS      {x-attribute 17};

inactivityTimeoutBehaviour  BEHAVIOUR

DEFINED AS
        !This attribute specifies the maximum amount of time (in 1/100ths of a second) that the transport
        connection can remain enabled when there is no activity (i.e., data flow ) on it.  A value of 0 for
        this attribute indicates that an inactivity timeout is not supported on the transport connection.!;
```

A.4.4.18 Local Network Address

```
localNetworkAddress       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.Address;
        MATCHES FOR    EQUALITY, SUBSTRINGS;
        BEHAVIOUR  localNetworkAddressBehaviour;

REGISTERED AS      {x-attribute 18};

localNetworkAddressBehaviour   BEHAVIOUR

DEFINED AS
        !The localNetworkAddress attribute identifies the local network address supported by a network
        protocol layer entity (e.g., local IP address for TCP or the local NSAP address for OSI).!;
```

A.4.4.19 Local Network Addresses

```
localNetworkAddresses       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX    SYNTAX-1.NetworkAddresses;
        MATCHES FOR   SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  localNetworkAddressesBehaviour;

REGISTERED AS      {x-attribute 19};

localNetworkAddressesBehaviour   BEHAVIOUR

DEFINED AS
        !The localNetworkAddresses attribute identifies the local network addresses supported by a network
        protocol layer entity (e.g., local IP address for TCP or the local NSAP address for OSI).
```

```
        Set comparison and/or set intersection matching rules may not be supported by some managed object
        instances which include this attribute.!;
```

A.4.4.20 Local Transport Addresses

```
localTransportAddresses    ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.TransportAddresses;
        MATCHES FOR   SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  localTransportAddressesBehaviour;

REGISTERED AS  {x-attribute 20};

localTransportAddressesBehaviour   BEHAVIOUR

DEFINED AS
        !The localTransportAddresses attribute specifies the set of local transport addresses (e.g, local
        TSAP identifiers) that a connection oriented transport protocol layer entity provides to its users.
        A transport address consists of a transport connection endpoint and a network address.

        Set comparison and/or set intersection matching rules may not be supported by some managed object
        instances which include this attribute.!;
```

A.4.4.21 Local Transport Connection Endpoint

```
localTransportConnectionEndpoint      ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.Address;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR   localTransportConnectionEndpointBehaviour;

REGISTERED AS      {x-attribute 21};

localTransportConnectionEndpointBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute identifies the local transport connection endpoint (e.g., the source port for TCP or
        the local t-selector for OSI Transport protocol).!;
```

A.4.4.22 Location Pointer

```
locationPointer       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.ObjectInstance;
        MATCHES FOR EQUALITY;
        BEHAVIOUR  locationPointerBehaviour;

REGISTERED AS {x-attribute 22};

locationPointerBehaviour       BEHAVIOUR

DEFINED AS
        !The Location Pointer attribute provides managed object instance information for a location (e.g.,
        Hilo Hawaii USA).  The following managed object classes (or any of their subclasses or allomorphic
        classes) are valid as locations:  "OP1 Library Vol. 4":Location.!;
```

A.4.4.23 Manufacturer List

```
manufacturerList  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR  SUBSTRINGS, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  manufacturerListBehaviour;

REGISTERED AS      {x-attribute 23};

manufacturerListBehaviour   BEHAVIOUR

DEFINED AS
        !The manufacturerList attribute indicates information about the manufacturer(s) that manufactured
        the underlying resource.  This attribute contains object instance name(s) for "OP1 Library Vol.
        4":manufacturer (or any subclass or allomorphic class).
```

```
        Set comparison and/or set intersection matching rules may not be supported by some managed object
        instances which include this attribute.!;
```

A.4.4.24 Manufacturer Name

```
manufacturerName  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR  EQUALITY, SUBSTRINGS;
        BEHAVIOUR  manufacturerNameBehaviour;

REGISTERED AS      {x-attribute 24};

manufacturerNameBehaviour   BEHAVIOUR

DEFINED AS
        !The manufacturerName attribute indicates information about the manufacturer that manufactured the
        underlying resource.  This attribute contains descriptive text.!;
```

A.4.4.25 Max Connections

```
maxConnections          ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  maxConnectionsBehaviour;

REGISTERED AS     {x-attribute 25};

maxConnectionsBehaviour   BEHAVIOUR

DEFINED AS
        !The maxConnections attribute specifies the maximum number of simultaneously active/open transport
        connections that can be supported by the transport protocol layer entity.!;
```

A.4.4.26 Max PDU Size

```
maxPDUSize          ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  maxPDUSizeBehaviour;

REGISTERED AS     {x-attribute 26};

maxPDUSizeBehaviour   BEHAVIOUR

DEFINED AS
        !The maxPDUSize attribute specifies the maximum length of a PDU that can be supported by the local
        layer entity.!;
```

A.4.4.27 Max Retransmissions

```
maxRetransmissions  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  maxRetransmissionsBehaviour;

REGISTERED AS     {x-attribute 27};

maxRetransmissionsBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the maximum number of times a TPDU is to be retransmitted before the
        transport connection is aborted.!;
```

A.4.4.28 Memory Size

```
memorySize       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.MemorySizeBase;
        MATCHES FOR EQUALITY, ORDERING;
```

```
        BEHAVIOUR  memorySizeBehaviour;

REGISTERED AS {x-attribute 28};

memorySizeBehaviour       BEHAVIOUR

DEFINED AS
        !The Memory Size attribute indicates, in kilobytes, the amount of memory available to a Processing
        Entity (irrespective of its current usage).!;
```

A.4.4.29 Memory Utilization

```
memoryUtilization       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.IntegerBase;

        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR  memoryUtilizationBehaviour;

REGISTERED AS {x-attribute 29};

memoryUtilizationBehaviour       BEHAVIOUR

DEFINED AS
        !The memoryUtilization attribute specifies, as a percentage, the overall utilization of amount of
        memory available to a processing entity.  The percentage is expressed as an integer with permissible
        values in the range of 0 to 100.!;
```

A.4.4.30 Network Entity Type

```
networkEntityType ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.NetworkEntityType;
        MATCHES FOR   EQUALITY;
        BEHAVIOUR  networkEntityTypeBehaviour;

REGISTERED AS      {x-attribute 30};

networkEntityTypeBehaviour   BEHAVIOUR

DEFINED AS
        !The networkEntityType attribute indicates the type of the network protocol layer entity.!;
```

A.4.4.31 Network Title

```
networkTitle       ATTRIBUTE

        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":systemTitle;
        BEHAVIOUR  networkTitleBehaviour;

REGISTERED AS {x-attribute 31};

networkTitleBehaviour       BEHAVIOUR

DEFINED AS
        !The Network Title is one of the distinguishing attributes of the OMNIPoint Network managed object
        class for use as described in clause 6.3 of [MIM]!;
```

A.4.4.32 NPDU Time To Live

```
nPDUTimeToLive          ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  nPDUTimeToLiveBehaviour;

REGISTERED AS      {x-attribute 32};

nPDUTimeToLiveBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the maximum amount of time (in units of 10 ms) that an NPDU can exist in
        the network.  This attribute is used to limit the lifetime of NPDUs during unstable network
        situations.!;
```

A.4.4.33 OMNIPoint Equipment List

```
opEquipmentList       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  opEquipmentListBehaviour;

REGISTERED AS {x-attribute 33};

opEquipmentListBehaviour       BEHAVIOUR

DEFINED AS
        !The OMNIPoint Equipment List attribute provides managed object instance information about one or
        more pieces of opEquipment.  The following classes (or any of their subclasses or allomorphic
        classes) are valid as equipment: OMNIPoint Equipment.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.34 OMNIPoint Network List

```
opNetworkList        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  opNetworkListBehaviour;

REGISTERED AS {x-attribute 34};

opNetworkListBehaviour       BEHAVIOUR

DEFINED AS
        !The OMNIPoint Network List attribute shall provide managed object instance information about a set
        of networks.  The following object classes (or any of their subclasses or allomorphic classes) are
        valid as networks: OMNIPoint Network.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.35 OMNIPoint Network Name

```
opNetworkName        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  opNetworkNameBehaviour;

REGISTERED AS {x-attribute 35};

opNetworkNameBehaviour       BEHAVIOUR

DEFINED AS
        !The OMNIPoint Network Name attribute shall provide information about a network.!;
```

A.4.4.36 Operating System Information

```
osInfo       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.OsInfoBase;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  osInfoBehaviour;

REGISTERED AS {x-attribute 36};

osInfoBehaviour       BEHAVIOUR

DEFINED AS
        !The Operating System Information attribute specifies the names and releases of the supported
        operating systems.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.37 PDUs Forwarded Counter

```
pdusForwardedCounter          ATTRIBUTE

        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992": counter;
        BEHAVIOUR    pdusForwardedCounterBehaviour;

REGISTERED AS     {x-attribute 37};

pdusForwardedCounterBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the number of valid incoming PDUs which were forwarded (transmitted as
        outgoing PDUs) to another destination.  This attribute does not count incoming PDUs which were
        delivered to a local service user.!;
```

A.4.4.38 PDUs Reassembled Ok Counter

```
pdusReasmbldOKCounter          ATTRIBUTE

        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992": counter;
        BEHAVIOUR  pdusReasmbldOKCounterBehaviour;

REGISTERED AS     {x-attribute 38};

pdusReasmbldOKCounterBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the number of PDUs that were reassembled successfully by a protocol layer
        entity.!;
```

A.4.4.39 PDUs Reassembled Fail Counter

```
pdusReasmbldFailCounter        ATTRIBUTE

        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992": counter;
        BEHAVIOUR  pdusReasmbldFailCounterBehaviour;

REGISTERED AS     {x-attribute 39};

pdusReasmbldFailCounterBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the number of valid PDUs received by a protocol layer entity but discarded
        due to reassembly failure.  This attribute counts only incoming PDUs which were recognized as valid
        segments of an SDU, but which were discarded during reassembly (for example, due to reassembly time
        expiration).!;
```

A.4.4.40 PDUs Discarded Counter

```
pdusDiscardedCounter        ATTRIBUTE

        DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992": counter;
        BEHAVIOUR  pdusDiscardedCounterBehaviour;

REGISTERED AS     {x-attribute 40};

pdusDiscardedCounterBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the number of invalid PDUs received and discarded by a protocol layer
        entity.!;
```

A.4.4.41 Peripheral List

```
peripheralList       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  peripheralListBehaviour;

REGISTERED AS {x-attribute 41};
```

```
peripheralListBehaviour       BEHAVIOUR

DEFINED AS
        !The Peripheral List attribute provides managed object instance information for peripheral devices
        accessible by a resource.

        The Peripheral List attribute identifies the auxiliary devices that are used by a resource.  This
        includes things such as disk drives, tape drives, printers, etc.

        The following object classes (or their subclasses or allomorphic classes) are valid processing
        entities: OMNIPoint Equipment.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.42 Peripheral Name

```
peripheralName       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  peripheralNameBehaviour;

REGISTERED AS {x-attribute 42};

peripheralNameBehaviour       BEHAVIOUR

DEFINED AS
        !The Peripheral Name attribute provides information for peripheral devices accessible by a resource.

        The Peripheral Name attribute identifies an auxiliary devices that is used by a resource.  This
        includes things such as disk drives, tape drives, printers, etc.!;
```

A.4.4.43 Processing Entity List

```
processingEntityList       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  processingEntityListBehaviour;

REGISTERED AS {x-attribute 43};

processingEntityListBehaviour       BEHAVIOUR

DEFINED AS
        !The Processing Entity List attribute specifies the processing entities which may be used by the
        containing object instance but which are not contained in (i.e., processing entities which are
        shared among systems).  The following object classes (or their subclasses or allomorphic classes)
        are valid processing entities: processingEntity.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.44 Processing Entity Name

```
processingEntityName       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  processingEntityNameBehaviour;

REGISTERED AS {x-attribute 44};

processingEntityNameBehaviour       BEHAVIOUR

DEFINED AS
        !The Processing Entity Name attribute specifies the processing entity which may be used by the
        containing object instance but which is not contained in (i.e., processing entities which are shared
        among systems).!;
```

A.4.4.45 Product Label

```
productLabel  ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX   SYNTAX-1.GraphicStringBase;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR  productLabelBehaviour;

REGISTERED AS      {x-attribute 45};

productLabelBehaviour   BEHAVIOUR

DEFINED AS
        !The productLabel attribute specifies the product number or identifying string (e.g., model number)
        of the underlying resource.!;
```

A.4.4.46 Remote Network Address

```
remoteNetworkAddress       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.Address;
        MATCHES FOR    EQUALITY, SUBSTRINGS;
        BEHAVIOUR  remoteNetworkAddressBehaviour;

REGISTERED AS      {x-attribute 46};

remoteNetworkAddressBehaviour   BEHAVIOUR

DEFINED AS
        !The remoteNetworkAddress attribute identifies the remote network address of a transport connection
        (e.g., remote IP address for TCP or the remote NSAP address for OSI).!;
```

A.4.4.47 Remote Transport Connection Endpoint

```
remoteTransportConnectionEndpoint      ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.Address;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR   remoteTransportConnectionEndpointBehaviour;

REGISTERED AS      {x-attribute 47};

remoteTransportConnectionEndpointBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute identifies the remote transport connection endpoint (e.g., the destination port for
        TCP or the remote t-selector for OSI Transport protocol).!;
```

A.4.4.48 Retransmission Time

```
retransmissionTime  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.HundredthsOfSec;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  retransmissionTimeBehaviour;

REGISTERED AS     {x-attribute 48};

retransmissionTimeBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute specifies the current value (in 1/100ths of a second) of the retransmission timer
        used by a transport connection.!;
```

A.4.4.49 Retransmission Timer Initial Value

```
retransmissionTimerInitialValue  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.HundredthsOfSec;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  retransmissionTimerInitialValueBehaviour;

REGISTERED AS     {x-attribute 49};

retransmissionTimerInitialValueBehaviour   BEHAVIOUR

DEFINED AS
```

```
        !This attribute specifies the initial value (in 1/100ths of a second) of the retransmission timer
        used by a transport connection.!;
```

A.4.4.50 Serial Number

```
serialNumber  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.GraphicStringBase;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR  serialNumberBehaviour;

REGISTERED AS      {x-attribute 50};

serialNumberBehaviour   BEHAVIOUR

DEFINED AS
        !The serialNumber attribute provides the serial number of the underlying resource.!;
```

A.4.4.51 Service List

```
serviceList        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  serviceListBehaviour;

REGISTERED AS {x-attribute 51};

serviceListBehaviour       BEHAVIOUR

DEFINED AS
        !The Service List attribute provides managed object instance information about one or more services.
        The following object classes (or any of their subclasses or allomorphic classes) are valid as
        services: "OP1 Library Vol. 4":Service.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.52 Service Name

```
serviceName        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  serviceNameBehaviour;

REGISTERED AS {x-attribute 52};

serviceNameBehaviour       BEHAVIOUR

DEFINED AS
        !The Service Name attribute provides information about one service.!;
```

A.4.4.53 Software List

```
softwareList        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  softwareListBehaviour;

REGISTERED AS {x-attribute 53};

softwareListBehaviour       BEHAVIOUR

DEFINED AS
        !The Software List attribute identifies those software components that run on or are considered part
        of the equipment.  (There is no corresponding managed object class at this time.)

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.4.54 Software Name

```
softwareName        ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNameBase;
        MATCHES FOR EQUALITY, SUBSTRINGS;
        BEHAVIOUR  softwareNameBehaviour;

REGISTERED AS {x-attribute 54};

softwareNameBehaviour       BEHAVIOUR

DEFINED AS
        !The Software Name attribute identifies the software component that runs on or is considered part of
        the equipment.!;
```

A.4.4.55 System Time

```
systemTime    ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.GeneralTime;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  systemTimeBehaviour;

REGISTERED AS     {x-attribute 55};

systemTimeBehaviour  BEHAVIOUR

DEFINED AS
        !The systemTime attribute specifies the current time clocked at the resource.!;
```

A.4.4.56 Transport Connection Id

```
transportConnectionId  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.GraphicStringBase;
        MATCHES FOR  EQUALITY, SUBSTRINGS;
        BEHAVIOUR  transportConnectionIdBehaviour;

REGISTERED AS     {x-attribute 56};

transportConnectionIdBehaviour  BEHAVIOUR

DEFINED AS
        !The transportConnectionId attribute is the distinguishing attribute for the transportConnection
        managed object class.!;
```

A.4.4.57 Transport Connection Reference

```
transportConnectionReference       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX         SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY;
        BEHAVIOUR  transportConnectionReferenceBehaviour;

REGISTERED AS      {x-attribute 57};

transportConnectionReferenceBehaviour   BEHAVIOUR

DEFINED AS
        !This attribute identifies the local transport connection reference that is established by the two
        transport connection endpoints (e.g., the local socket number for TCP or the local connection
        reference for OSI).!;
```

A.4.4.58 Transport Entity Type

```
transportEntityType ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.TransportEntityType;
        MATCHES FOR   EQUALITY;
        BEHAVIOUR  transportEntityTypeBehaviour;
```

```
REGISTERED AS      {x-attribute 58};

transportEntityTypeBehaviour   BEHAVIOUR

DEFINED AS
        !The transportEntityType attribute indicates the type of the transport protocol layer entity.!;
```

A.4.4.59 Type Text

```
typeText  ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.GraphicStringBase;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR  typeTextBehaviour;

REGISTERED AS      {x-attribute 59};

typeTextBehaviour   BEHAVIOUR

DEFINED AS
        !The typeText attribute serves to supplement and refine individual managed object class attributes.
        If none of the named items defined for the "type" attribute are appropriate, or the "type" attribute
        requires refinement, the typeText attribute contains supplemental information.!;
```

A.4.4.60 Up Time

```
upTime    ATTRIBUTE

        WITH ATTRIBUTE SYNTAX   SYNTAX-1.IntegerBase;
        MATCHES FOR   EQUALITY, ORDERING;
        BEHAVIOUR  upTimeBehaviour;

REGISTERED AS     {x-attribute 60};

upTimeBehaviour  BEHAVIOUR

DEFINED AS
        !The upTime attribute specifies the time interval (in seconds) that has elapsed since the entity's
        operational state changed to "enabled", or since the time that the entity was created in the
        "enabled" state.!;
```

A.4.4.61 Vendor List

```
vendorList       ATTRIBUTE

        WITH ATTRIBUTE SYNTAX  SYNTAX-1.AnyNamesBase;
        MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR  vendorListBehaviour;

REGISTERED AS {x-attribute 61};

vendorListBehaviour       BEHAVIOUR

DEFINED AS
        !The Vendor List attribute provides managed object instance information about a set of vendor
        organizations. The following classes (or any of their subclasses or allomorphic classes) are valid
        as vendors:  "OP1 Library Vol. 4":Vendor.

        The SET-COMPARISON and/or SET-INTERSECTION matching rules may not be supported by some managed
        object instances which include this attribute.!;
```

A.4.5 Actions

A.4.5.1 Activate

```
-- Copied from ISO/IEC DIS 10737, should be replaced by reference to standard
```

```
-- definition when/if this ACTION is registered in a final IS version.

activate ACTION

        BEHAVIOUR activateBehaviour;
        MODE CONFIRMED;
        WITH REPLY SYNTAX SYNTAX-1.ActivateActionReply;

REGISTERED AS { x-action 1 };


activateBehaviour BEHAVIOUR

DEFINED AS
        !This action initializes the operation of the resource.  As a result of the action, the sequence of
        operations necessary to cause the resource to enter its operational mode shall be initiated.  These
        may include, for example, checks against attribute constraint violation and checks on the validity
        of relationship attributes (cross-layer and other).  If these operations are successfully initiated,
        the administrative state (if present) shall be changed to "unlocked" and the value "successResponse"
        shall be returned in the responseCode parameter of the action reply.  If these operations cannot be
        successfully initiated, the value "failureResponse" shall be returned, together with a failure
        reason parameter describing the reason for the failure.  On successful completion of these
        operations, the operational state shall have the value "enabled".  Depending upon the current state
        of the resource when the action is attempted, some or all of the above operations may be
        unnecessary.!;
```

A.4.5.2 Deactivate

```
-- Copied from ISO/IEC DIS 10737, should be replaced by reference to standard
-- definition when/if this ACTION is registered in a final IS version.

deactivate ACTION

        BEHAVIOUR deactivateBehaviour;
        MODE CONFIRMED;
        WITH REPLY SYNTAX SYNTAX-1.ActivateActionReply;

REGISTERED AS { x-action 2 };


deactivateBehaviour BEHAVIOUR

DEFINED AS
        !This action terminates the operation of the resource.  As a result of the action, the sequence of
        operations necessary to cause the resource to cease operation shall be initiated.  If these
        operations are successfully initiated, the administrative state (if present) shall be changed to
        "locked" and the value "successResponse" shall be returned in the responseCode parameter of the
        action reply.  If these operations cannot be successfully initiated, the value "failureResponse"
        shall be returned, together with a failure reason parameter describing the reason for the failure.
        On successful completion of these operations, the operational state shall have the value "disabled".
        Depending upon the current state of the resource when the action is attempted, some or all of the
        above operations may be unnecessary.!;
```

A.4.6 Parameters

A.4.6.1 Transport Disconnect Cause

```
transportDisconnectCause PARAMETER

        CONTEXT   EVENT-INFO;
        WITH SYNTAX SYNTAX-1.Cause;
        BEHAVIOUR causeBehaviour;

REGISTERED AS { x-parameter 1 };


causeBehaviour BEHAVIOUR

DEFINED AS
        !This parameter specifies the reason why a transport connection was deleted.  It may be included in
        the Additional Information parameter of the objectDeletion notification.!;
```

A.4.7 Syntax Definitions

```
SYNTAX-1 { x-module 1 }
DEFINITIONS IMPLICIT TAGS ::= BEGIN
IMPORTS DistinguishedName FROM InformationFramework {joint-iso-ccitt ds(5) modules(1)
informationFramework(1)} ObjectInstance FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}
NameType FROM ASN1DefinedTypesModule {ccitt recommendation m gnm(3100) informationModel(0) asn1Modules(2)
asn1DefinedTypesModule(0)};


--   EXPORTS everything


--   The following OIDs are allocated from the OIW NMSIG registration arc,
--   for use in registering harmonized OIW/NMF definitions.


        nmsig            OBJECT IDENTIFIER ::=  { iso identified-organization(3) oiw(14) nmsig(2) }
        op1LibraryVol1   OBJECT IDENTIFIER ::=  { nmsig 2 }
        x-module         OBJECT IDENTIFIER ::=  { op1LibraryVol1 0 }
        x-objectClass    OBJECT IDENTIFIER ::=  { op1LibraryVol1 1 }
        x-package        OBJECT IDENTIFIER ::=  { op1LibraryVol1 2 }
        x-nameBinding    OBJECT IDENTIFIER ::=  { op1LibraryVol1 3 }
        x-attribute      OBJECT IDENTIFIER ::=  { op1LibraryVol1 4 }
        x-attributeGroup OBJECT IDENTIFIER ::=  { op1LibraryVol1 5 }
        x-parameter      OBJECT IDENTIFIER ::=  { op1LibraryVol1 6 }
        x-action         OBJECT IDENTIFIER ::=  { op1LibraryVol1 7 }
        x-notification   OBJECT IDENTIFIER ::=  { op1LibraryVol1 8 }
        x-responseCode   OBJECT IDENTIFIER ::=  { op1LibraryVol1 9 }


--   By convention, the postfix "base" is used when defining base types which appear
--   as syntax labels in ATTRIBUTE templates and the postfix "range" is used when defining
--   constrained types which appear as syntax labels in PERMITTED VALUES clauses.


        ActivateActionReply      ::=  SEQUENCE {
                                          responseCode     OBJECT IDENTIFIER,
                                          responseArgs     SET OF Parameter OPTIONAL
                                          }

        -- OBJECT IDENTIFIER values used with ActivateActionReply --
        failureResponse  OBJECT IDENTIFIER ::=  { x-responseCode 1 }
        successResponse  OBJECT IDENTIFIER ::=  { x-responseCode 2 }


        Address          ::=  OCTET STRING

        AddressingSizeBase       ::= CHOICE {
                                 unknown           NULL,
                                 addressingSize    IntegerBase         -- measured in bits
                                 }

        AddressingSizeRange      ::= AddressingSizeBase
                (
                        WITH COMPONENTS
                                {
                                        unknown,
                                        addressingSize (1..64)
                                }
                )

        AnyNamesBase             ::=  SET OF ObjectInstance
        AnyNameBase              ::=  GraphicStringBase


        AnyNamesRange            ::=  SET SIZE(0..64) OF ObjectInstance
        AnyNameRange             ::=  GraphicString64


        Cause                    ::=  SEQUENCE {
                                              who INTEGER {
                                              unknown                  (0),
                                              user                     (1),
                                              provider                 (2)
```

```
                                        },
                                        why INTEGER {
                                        unknown                  (0),
                                        excessiveIdle            (1),
                                        excessiveRetransmissions (2)
                                        }}

Endianess                ::= ENUMERATED {
                                big      (1),
                                little   (2)
                                }


EquipmentIdRange         ::=     NameType
        (
                WITH COMPONENTS
                        {
                                numericName  (0..4294967295),
                                pString (SIZE(0..64))
                        }
        )


GeneralTime              ::=  GeneralizedTime


GraphicStringBase        ::=  GraphicString
GraphicString16          ::=  GraphicStringBase(SIZE(0..16))
GraphicString32          ::=  GraphicStringBase(SIZE(0..32))
GraphicString64          ::=  GraphicStringBase(SIZE(0..64))


HundredthsOfSec          ::=  IntegerBase


IntegerBase              ::=  INTEGER
Integer32                ::=  IntegerBase(0..4294967295)


MemorySizeBase           ::= CHOICE {
                         unknown NULL,
                         size            IntegerBase   -- measured in kilobytes
                         }

MemorySizeRange          ::=     MemorySizeBase
        (
                WITH COMPONENTS
                        {
                                unknown,
                                size (0..4294967295)
                        }
        )


NetworkEntityType        ::= INTEGER { other      (0),
                         oSI-clnp                 (1),
                         internet-IP              (2)
                         } (0..255)


NetworkAddresses         ::=  SET OF Address


OsInfoBase               ::=  SET OF SEQUENCE
                                {
                                osName     GraphicStringBase,
                                osRelease  GraphicStringBase
                                }

OsInfoRange              ::=     OsInfoBase
        (
                WITH COMPONENTS
                        {
                                osName (SIZE(0..64)),
                                osRelease (SIZE(0..64))
                        }
        )
```

```
Parameter                ::=  SEQUENCE {
                                 paramId    OBJECT IDENTIFIER,
                                 paramInfo  ANY DEFINED BY paramId
                                 }


PercentageRange          ::=  IntegerBase (0..100)


TransportAddresses       ::=  SET OF TransportAddress

TransportAddress         ::=  SEQUENCE {
                         transportConnectionEndpoint Address,
                         networkAddress              Address
                         }

TransportEntityType      ::= INTEGER { other      (0),
                         oSI-TP                   (1),
                         tCP                      (2),
                         sNA                      (3)
                         } (0..255)


END
```

A.4.8 Inheritance & Naming Trees

This section provides graphic depictions for the inheritance and naming trees that are defined in the previous sections.

A.4.8.1 Inheritance Tree

```
top -+-- opEquipment -----       processingEntity
     |
     +-- computerSystem
     |
     +-- opNetwork
     |
     +-- coTransportProtocolLayerEntity
     |
     +-- clNetworkProtocolLayerEntity
     |
     +-- transportConnection
```

A.4.8.2 Naming Tree

```
root -+-- opNetwork --+-- opNetwork
      |               |
      |               +-- computerSystem
      |               |
      |               +-- opEquipment
      |
      |
      +-- DMI system -+-- coTransportProtocolLayerEntity -- transportConn
                      |
                      +-- clNetworkProtocolLayerEntity
                      |
                      +-- opEquipment-+-- opEquipment
                      |               |
                      |               +-- processingEntity
                      |               |
                      |               +-- coTransportProtocolLayerEntity -- transportConn
                      |               |
                      |               +-- clNetworkProtocolLayerEntity
                      |
                      |
                      +-- computerSystem --+-- computerSystem
```

```
                                          |
                                          +-- processingEntity
                                          |
                                          +-- opEquipment
                                          |
                                          +-- coTransportProtocolLayerEntity -- transportConn
                                          |
                                          +-- clNetworkProtocolLayerEntity
--------------------------------------------------------------------------------
```

A.5 OIW NMSIG IVMO Definitions

The definitions specified in this clause can be referenced by using the label "OP1 Library Vol. 2" (e.g., "OP1 Library Vol. 2":transportConnectionIVMO).

A.5.1 Managed Object Classes and Mandatory Packages

A.5.1.1 Transport Connection IVMO

```
transportConnectionIVMO   MANAGED OBJECT CLASS

        DERIVED FROM  "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
        CHARACTERIZED BY transportConnectionIVMO-Package;

REGISTERED AS {y-objectClass 1};


transportConnectionIVMO-Package   PACKAGE
        BEHAVIOUR  transportConnectionIVMO-behaviour;
        ATTRIBUTES
                transportConnectionIVMOId  GET,
                "OP1 Library Vol. 1":inactivityTimeout  PERMITTED VALUES SYNTAX-1.Integer32  GET-REPLACE,
                "OP1 Library Vol. 1":maxPDUSize  PERMITTED VALUES SYNTAX-1.Integer32  GET-REPLACE;
        NOTIFICATIONS
                "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
                "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion,
                "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChange;;


transportConnectionIVMO-behaviour  BEHAVIOUR

        DEFINED AS

                !This managed object class is an IVMO (Initial Value Managed Object class).  It represents
                the collection of characteristic attributes which supply default and initially advertised
                attribute values to be used by instances of the Transport Connection managed object class
                when they are created.  There can be only one instance of the Transport Connection IVMO
                managed object class for each instance of the CO Transport Protocol Layer Entity managed
                object class.  Each Transport Connection IVMO instance may provide initial attribute values
                for newly-created Transport Connection instances with the same superior.

                The Attribute List parameter of the ObjectCreation notification shall contain all the
                attributes of the created transport connection IVMO instance.

                The Attribute List parameter of the ObjectDeletion notification shall contain all the
                attributes of the deleted transport connection IVMO instance.

                Attributes that are subject to the AttributeValueChange notification are : "OP1 Library
                Vol. 1":inactivityTimeout, "OP1 Library Vol. 1":maxPDUSize.  All attributeValueChange
                notifications shall include the Attribute Identifier List parameter.!;
```

A.5.1.2 Transport Connection Retransmission IVMO

```
transportConnectionRetransmissionIVMO   MANAGED OBJECT CLASS

        DERIVED FROM   transportConnectionIVMO;
        CHARACTERIZED BY  transportConnectionRetransmissionIVMO-Package;
```

```
REGISTERED AS  {y-objectClass 3};


transportConnectionRetransmissionIVMO-Package  PACKAGE
        BEHAVIOUR  transportConnectionIVMO-behaviour;
        ATTRIBUTES
                "OP1 Library Vol. 1":maxRetransmissions  PERMITTED VALUES SYNTAX-1.Integer32  GET-REPLACE,
                "OP1 Library Vol. 1":retransmissionTimerInitialValue
                                                       PERMITTED VALUES SYNTAX-1.Integer32 GET-REPLACE;;


transportConnectionRetransmissionIVMO-behaviour  BEHAVIOUR

        DEFINED AS

                !This managed object class is an IVMO (Initial Value Managed Object class).  It represents
                the collection of characteristic attributes which supply default and initially advertised
                attribute values to be used by instances of the Transport Connection managed object class
                that support retransmission, when they are created.  There can be only one instance of the
                Transport Connection Retransmission IVMO managed object class for each instance of the CO
                Transport Protocol Layer Entity managed object class.  Each Transport Connection
                Retransmission IVMO instance may provide initial attribute values for newly-created
                Transport Connection instances with the same superior.

                Attributes, additional to those inherited from the transport connection IVMO managed object
                class, that are subject to the AttributeValueChange notification are : "OP1 Library Vol.
                1":maxRetransmissions, "OP1 Library Vol. 1":retransmissionTimerInitialValue.!;
```

A.5.2 Name Bindings

A.5.2.1 Transport Connection IVMO Name Bindings

```
transportConnectionIVMO-coTransportProtocolLayerEntity  NAME BINDING
        SUBORDINATE OBJECT CLASS transportConnectionIVMO AND SUBCLASSES;
        NAMED BY
        SUPERIOR OBJECT CLASS "OP1 Library Vol. 1":coTransportProtocolLayerEntity AND SUBCLASSES;
        WITH ATTRIBUTE transportConnectionIVMOId;
REGISTERED AS  {y-nameBinding 1};
```

A.5.2.2 Transport Connection Retransmission IVMO Name Bindings

```
transportConnectionRetransmissionIVMO-coTransportProtocolLayerEntity NAME BINDING
        SUBORDINATE OBJECT CLASS transportConnectionRetransmissionIVMO
        AND SUBCLASSES;
        NAMED BY
        SUPERIOR OBJECT CLASS "OP1 Library Vol. 1":coTransportProtocolLayerEntity AND SUBCLASSES;
        WITH ATTRIBUTE transportConnectionIVMOId;
REGISTERED AS  {y-nameBinding 2};
```

A.5.3 Attributes

A.5.3.1 Transport Connection IVMO Id

```
transportConnectionIVMOId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   SYNTAX-1.GraphicStringBase;
        MATCHES FOR   EQUALITY, SUBSTRINGS;
        BEHAVIOUR  transportConnectionIVMOIdBehaviour;

REGISTERED AS     {y-attribute 1};


transportConnectionIVMOIdBehaviour   BEHAVIOUR

DEFINED AS      !This attribute is the distinguishing attribute for the managed object class
                transportConnectionIVMO.!;
```

A.5.4 Syntax Definitions

```
SYNTAX-2 { y-module 1 }
DEFINITIONS IMPLICIT TAGS ::= BEGIN

--   EXPORTS everything

--   The following OIDs are allocated from the OIW NMSIG registration arc,
--   for use in registering OIW NMSIG MIL definitions.

        nmsig             OBJECT IDENTIFIER ::=  { iso identified-organization(3) oiw(14) nmsig(2) }
        op1LibraryVol2    OBJECT IDENTIFIER ::=  { nmsig 1 }
        y-module          OBJECT IDENTIFIER ::=  { op1LibraryVol2 0 }
        y-objectClass     OBJECT IDENTIFIER ::=  { op1LibraryVol2 1 }
        y-package         OBJECT IDENTIFIER ::=  { op1LibraryVol2 2 }
        y-nameBinding     OBJECT IDENTIFIER ::=  { op1LibraryVol2 3 }
        y-attribute       OBJECT IDENTIFIER ::=  { op1LibraryVol2 4 }
        y-attributeGroup  OBJECT IDENTIFIER ::=  { op1LibraryVol2 5 }
        y-parameter       OBJECT IDENTIFIER ::=  { op1LibraryVol2 6 }
        y-action          OBJECT IDENTIFIER ::=  { op1LibraryVol2 7 }
        y-notification    OBJECT IDENTIFIER ::=  { op1LibraryVol2 8 }

END
```

A.5.5 Inheritance & Naming Trees

This section provides graphic depictions for the inheritance and naming trees that are defined in the previous sections.

A.5.5.1 Inheritance Tree

```
top ---- transportConnectionIVMO -----    transportConnectionRetransmissionIVMO
```

A.5.5.2 Naming Tree

```
coTransportProtocolLayerEntity -+-- transportConnection
                                |
                                +-- transportConnectionIVMO
                                |
                                +-- transportConnectionRetransmissionIVMO
```

A.6 OIW NMSIG Shared Management Knowledge (SMK) Definitions

(Refer to the Working Implementation Agreements Document.)

Annex B (informative)

NMSIG Object Identifiers

(Refer to the Working Implementation Agreements Document for additional information.)

B.1 Introduction

This Annex (B) specifies object identifier component values which are globally unambiguous. These object identifiers are to be used when referencing NMSIG–specified information objects. As defined in Part 6 of these agreements, the OIW has assigned the following object identifier for use by the NMSIG:

{ iso(1) identified–organization(3) oiw(14) nmsig(2) }

The following object identifiers are assigned under the { iso identified–organization oiw nmsig } node, labelled "nmsig".

Table B.1 - Object identifiers assigned under "nmsig" node

Identifier	Value	Reference
op1LibraryVol2	1	A.5
op1LibraryVol1	2	A.4

By inclusion of the managed object (MO) definitions and the object identifiers in Annex A and Annex B, respectively, of the Stable Implementors' Agreements (SIAs), these managed object (MO) definitions have become formally registered. Implementors of part 18 of the SIAs do not have to support any of these MOs. However, even though Annex A and Annex B are informative annexes, any implementation that claims to conform to these definitions must treat these definitions as normative and comply with the relevant portions of Annex A.4 and A.5, and Annex B.

B.2 Harmonized MIL Object Identifiers

Harmonized MIL Object Identifiers are assigned under the "nmsig" node as follows:

```
nmsig            OBJECT IDENTIFIER ::=  { iso identified-organization(3) oiw(14) nmsig(2) }
op1LibraryVol1   OBJECT IDENTIFIER ::=  { nmsig 2 }
x-module         OBJECT IDENTIFIER ::=  { op1LibraryVol1 0 }
x-objectClass    OBJECT IDENTIFIER ::=  { op1LibraryVol1 1 }
x-package        OBJECT IDENTIFIER ::=  { op1LibraryVol1 2 }
x-nameBinding    OBJECT IDENTIFIER ::=  { op1LibraryVol1 3 }
x-attribute      OBJECT IDENTIFIER ::=  { op1LibraryVol1 4 }
x-attributeGroup OBJECT IDENTIFIER ::=  { op1LibraryVol1 5 }
x-parameter      OBJECT IDENTIFIER ::=  { op1LibraryVol1 6 }
x-action         OBJECT IDENTIFIER ::=  { op1LibraryVol1 7 }
x-notification   OBJECT IDENTIFIER ::=  { op1LibraryVol1 8 }
x-responseCode   OBJECT IDENTIFIER ::=  { op1LibraryVol1 9 }
```

B.2.1 Object Class Object Identifiers

The following object identifiers are assigned under the { x–objectClass } node:

Table B.2 - Object identifiers assigned under "x-objectClass" node

Reference	Identifier	Value
A.4.1.1	computerSystem	1
A.4.1.2	coTransportProtocolLayerEntity	2
A.4.1.3	clNetworkProtocolLayerEntity	3
A.4.1.4	opEquipment	4
A.4.1.5	opNetwork	5
A.4.1.6	processingEntity	6
A.4.1.7	transportConnection	7

B.2.2 Package Object Identifiers

The following object identifiers are assigned under the { x–package } node:

Table B.3 - Object identifiers assigned under "x-package" node

Reference	Identifier	Value
A.4.2.1	addressingPkg	1
A.4.2.2	checksumPDUsDiscardedPkg	2
A.4.2.3	contactListPkg	3
A.4.2.4	contactNamePkg	4
A.4.2.5	cpuUtilizationPkg	5
A.4.2.6	customerListPkg	6
A.4.2.7	customerNamePkg	7
A.4.2.8	functionListPkg	8
A.4.2.9	functionNamePkg	9
A.4.2.10	incomingProtocolErrorPkg	10
A.4.2.11	locationPointerPkg	11
A.4.2.12	manufacturerListPkg	12
A.4.2.13	manufacturerNamePkg	13
A.4.2.14	maxPDUSizeIVPkg	14
A.4.2.15	maxRetransmissionsPkg	15
A.4.2.16	memorySizePkg	16
A.4.2.17	memoryUtilizationPkg	17
A.4.2.18	octetsRetransmittedPkg	18
A.4.2.19	opNetworkListPkg	19
A.4.2.20	opNetworkNamePkg	20

Reference	Identifier	Value
A.4.2.21	opVersionPkg	21
A.4.2.22	outgoingProtocolErrorPkg	22
A.4.2.23	pdusRetransmittedCounterPkg	23
A.4.2.24	pdusRetransmittedThresholdPkg	24
A.4.2.25	peripheralListPkg	25
A.4.2.26	peripheralNamePkg	26
A.4.2.27	processingEntityListPkg	27
A.4.2.28	processingEntityNamePkg	28
A.4.2.29	productLabelPkg	29
A.4.2.30	retransmissionTimePkg	30
A.4.2.31	retransmissionTimerInitialValuePkg	31
A.4.2.32	serialNumberPkg	32
A.4.2.33	serviceListPkg	33
A.4.2.34	serviceNamePkg	34
A.4.2.35	softwareListPkg	35
A.4.2.36	softwareNamePkg	36
A.4.2.37	systemTimePkg	37
A.4.2.38	typeTextPkg	38
A.4.2.39	upTimePkg	39
A.4.2.40	usageStatePkg	40
A.4.2.41	vendorListPkg	41

B.2.3 Name Bindings Object Identifiers

The following object identifiers are assigned under the { x–nameBinding } node:

Table B.4 - Object identifiers assigned under "x-nameBinding" node

Reference	Identifier	Value
A.4.3.2	computerSystem-system	1
A.4.3.2	computerSystem-opNetwork	2
A.4.3.2	computerSystem-computerSystem	3
A.4.3.3	coTransportProtocolLayerEntity-computerSystem	4
A.4.3.3	coTransportProtocolLayerEntity-system	5
A.4.3.3	coTransportProtocolLayerEntity-opEquipment	6
A.4.3.4	clNetworkProtocolLayerEntity-computerSystem	7
A.4.3.4	clNetworkProtocolLayerEntity-system	8

Reference	Identifier	Value
A.4.3.4	clNetworkProtocolLayerEntity-opEquipment	9
A.4.3.5	opEquipment-computerSystem	10
A.4.3.5	opEquipment-system	11
A.4.3.5	opEquipment-equipment	12
A.4.3.5	opEquipment-opNetwork	13
A.4.3.6	network-opNetwork-1	14
A.4.3.6	network-opNetwork-2	15
A.4.3.6	opNetwork-root	16
A.4.3.8	transportConnection-coTransportProtocolLayerEntity	17

B.2.4 Attribute Object Identifiers

The following object identifiers are assigned under the { x–attribute } node:

Table B.5 - Object identifiers assigned under "x-attribute" node

Reference	Identifier	Value
A.4.4.1	activeConnections	1
A.4.4.2	addressingSize	2
A.4.4.3	checksumPDUsDiscardedCounter	3
A.4.4.4	computerSystemId	4
A.4.4.5	clNetworkProtocolLayerEntityId	5
A.4.4.6	coTransportProtocolLayerEntityId	6
A.4.4.7	contactList	7
A.4.4.8	contactName	8
A.4.4.9	cpuType	9
A.4.4.10	cpuUtilization	10
A.4.4.11	customerList	11
A.4.4.12	customerName	12
A.4.4.13	endianess	13
A.4.4.14	functionList	14
A.4.4.15	functionName	15
A.4.4.16	inactivityTime	16
A.4.4.17	inactivityTimeout	17
A.4.4.18	localNetworkAddress	18
A.4.4.19	localNetworkAddresses	19
A.4.4.20	localTransportAddresses	20

Reference	Identifier	Value
A.4.4.21	localTransportConnectionEndpoint	21
A.4.4.22	locationPointer	22
A.4.4.23	manufacturerList	23
A.4.4.24	manufacturerName	24
A.4.4.25	maxConnections	25
A.4.4.26	maxPDUSize	26
A.4.4.27	maxRetransmissions	27
A.4.4.28	memorySize	28
A.4.4.29	memoryUtilization	29
A.4.4.30	networkEntityType	30
A.4.4.31	networkTitle	31
A.4.4.32	npduTimeToLive	32
A.4.4.33	opEquipmentList	33
A.4.4.34	opNetworkList	34
A.4.4.35	opNetworkName	35
A.4.4.36	osInfo	36
A.4.4.37	pdusForwardedCounter	37
A.4.4.38	pdusReasmbldOkCounter	38
A.4.4.39	pdusReasmbldFailCounter	39
A.4.4.40	pdusDiscardedCounter	40
A.4.4.41	peripheralList	41
A.4.4.42	peripheralName	42
A.4.4.43	processingEntityList	43
A.4.4.44	processingEntityName	44
A.4.4.45	productLabel	45
A.4.4.46	remoteNetworkAddress	46
A.4.4.47	remoteTransportConnectionEndpoint	47
A.4.4.48	retransmissionTime	48
A.4.4.49	retransmissionTimerInitialValue	49
A.4.4.50	serialNumber	50
A.4.4.51	serviceList	51
A.4.4.52	serviceName	52
A.4.4.53	softwareList	53
A.4.4.54	softwareName	54
A.4.4.55	systemTime	55

Reference	Identifier	Value
A.4.4.56	transportConnectionId	56
A.4.4.57	transportConnectionReference	57
A.4.4.58	transportEntityType	58
A.4.4.59	typeText	59
A.4.4.60	upTime	60
A.4.4.61	vendorList	61

B.2.5 Action Object Identifiers

The following object identifiers are assigned under the { x–action } node:

Table B.6 - Object identifiers assigned under "x-action" node

Reference	Identifier	Value
A.4.5.1	activate	1
A.4.5.2	deactivate	2

B.2.6 Parameter Object Identifiers

The following object identifiers are assigned under the { x–parameter } node:

Table B.7 - Object identifiers assigned under "x-parameter" node

Reference	Identifier	Value
A.4.6.1	transportDisconnectCause	1

B.2.7 Response Code Object Identifiers

The following object identifiers are assigned under the { x–responseCode } node:

Table B.8 - Object identifiers assigned under "x-responseCode" node

Reference	Identifier	Value
A.4.7	failureResponse	1
A.4.7	successResponse	2

B.2.8 Module Object Identifiers

The following object identifiers are assigned under the { x–module } node:

Table B.9 - Object identifiers assigned under "x-module" node

Reference	Identifier	Value
A.4.7	SYNTAX-1	1

B.3 Phase 1 MIL Object Identifiers

Phase 1 MIL Object Identifiers are assigned under the "nmsig" node as follows:

```
op1LibraryVol2   OBJECT IDENTIFIER ::=  { nmsig 1 }
y-module         OBJECT IDENTIFIER ::=  { op1LibraryVol2 0 }
y-objectClass    OBJECT IDENTIFIER ::=  { op1LibraryVol2 1 }
y-package        OBJECT IDENTIFIER ::=  { op1LibraryVol2 2 }
y-nameBinding    OBJECT IDENTIFIER ::=  { op1LibraryVol2 3 }
y-attribute      OBJECT IDENTIFIER ::=  { op1LibraryVol2 4 }
y-attributeGroup OBJECT IDENTIFIER ::=  { op1LibraryVol2 5 }
y-parameter      OBJECT IDENTIFIER ::=  { op1LibraryVol2 6 }
y-action         OBJECT IDENTIFIER ::=  { op1LibraryVol2 7 }
y-notification   OBJECT IDENTIFIER ::=  { op1LibraryVol2 8 }
```

B.3.1 Object Class Object Identifiers

The following object identifiers are assigned under the { y-objectClass } node:

Table B.10 - Object identifiers assigned under "y-objectClass" node

Reference	Identifier	Value
A.5.1.1	transportConnectionIVMO	1
A.5.1.2	transportConnectionRetransmissionIVMO	3 [See note below]

Note: **[**Previous version (value 2) has been deprecated in favor of this version (value 3).**]**

B.3.2 Name Bindings Object Identifiers

The following object identifiers are assigned under the { y–nameBinding } node:

Table B.11 - Object identifiers assigned under "y-nameBinding" node

Reference	Identifier	Value
A.5.2.1	transportConnectionIVMO-coTransportProtocolLayerEntity	1
A.5.2.2	transportConnectionRetransmissionIVMO-coTransportProtocolLayerEntity	2

B.3.3 Attribute Object Identifiers

The following object identifiers are assigned under the { y–attribute } node:

Table B.12 - Object identifiers assigned under "y-attribute" node

Reference	Identifier	Value
A.5.3.1	transportConnectionIVMOId	1

B.3.4 Module Object Identifiers

The following object identifiers are assigned under the { y–module } node:

Table B.13 - Object identifiers assigned under "y-module" node

Reference	Identifier	Value
A.5.4	SYNTAX-2	1

Annex C (informative)

MOCS Proforma

C.1 Introduction

The purpose of this MOCS proforma is to provide a mechanism for a supplier of an implementation of these agreements which claims conformance to a managed object class to provide conformance information in a standard form.

C.2 Symbols, abbreviations, and terms

The MOCS proforma contained in this Annex is comprised of information in a tabular format in accordance with the guidelines presented in ISO/IEC 9646–2 [ATSS] and ISO/IEC 10165–6 [MICS].

The following common notations, defined in ISO/IEC 9646–2, are used for the status column.

c conditional
m mandatory
o optional
x prohibited
– not applicable

The following common notations, defined in ISO/IEC 9646–2, are used for the support column.

Ig the item is ignored (i.e., processed syntactically but not semantically)
N not implemented
Y implemented
– not applicable

C.3 Instructions for completing the MOCS proforma to produce a MOCS

The supplier of the implementation shall enter an explicit statement in each of the boxes provided using the notation described in clause C.2. Additional instructions are provided in ISO/IEC 10165–6, Annex B.

C.4 Statements of Conformance to Managed Object Classes

This clause contains a MOCS Proforma for each managed object class defined in Annex A of these agreements, and registered by Annex B of these agreements.

C.4.1 Computer System MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":computerSystem	{ 1 3 14 2 2 1 1 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.1.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.1.1.1	computerSystem-system	{ 1 3 14 2 2 3 1 }	"Rec. X.721 | ISO/IEC 10165-2 : 1992":system	o		m	m			
C.4.1.1.2	computerSystem-opNetwork	{ 1 3 14 2 2 3 2 }	opNetwork	o		m	m			
C.4.1.1.3	computerSystem-computerSystem	{ 1 3 14 2 2 3 3 }	computerSystem	o		m	m			

Table C.4.1.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			setbycreate	get	replace	add	remove	settodefault	setbycreate	get	replace	add	remove	settodefault	
C.4.1.2.1	peripheralName	{ 1 3 14 2 2 4 42 }	c3	c3	c3	x	x	x							
C.4.1.2.2	peripheralList	{ 1 3 14 2 2 4 41 }	c4	c4	c4	c4	c4	x							
C.4.1.2.3	processingEntityName	{ 1 3 14 2 2 4 44 }	c5	c5	c5	x	x	x							
C.4.1.2.4	processingEntityList	{ 1 3 14 2 2 4 43 }	c6	c6	c6	c6	c6	x							
C.4.1.2.5	systemTime	{ 1 3 14 2 2 4 55 }	x	c0	x	x	x	x							
C.4.1.2.6	upTime	{ 1 3 14 2 2 4 60 }	x	c0	x	x	x	x							
C.4.1.2.7	"Rec. M.3100 : 1992":userLabel	{ 0 0 13 3100 0 7 50 }	c0	c0	c0	x	x	x							
C.4.1.2.8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": usageState	{ 2 9 3 2 7 39 }	x	c7	x	x	x	x							
C.4.1.2.9	"Rec. X.721 | ISO/IEC 10165-2 : 1992": operationalState	{ 2 9 3 2 7 35 }	x	m	x	x	x	x							
C.4.1.2.10	"Rec. X.721 | ISO/IEC 10165-2 : 1992": administrativeState	{ 2 9 3 2 7 31 }	m	m	m	x	x	x							
C.4.1.2.11	"Rec. X.721 | ISO/IEC 10165-2 : 1992": alarmStatus	{ 2 9 3 2 7 32 }	m	m	m	m	m	x							
C.4.1.2.12	"Rec. X.721 | ISO/IEC 10165-2 : 1992": availabilityStatus	{ 2 9 3 2 7 33 }	x	m	x	x	x	x							
C.4.1.2.13	computerSystemId	{ 1 3 14 2 2 4 4 }	m	m	x	x	x	x							
C.4.1.2.14	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.1.2.15	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.1.2.16	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							
C.4.1.2.17	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -
c3 = m if an instance supports it and the peripheralListPkg is NOT present, else x
c4 = m if an instance supports it and the peripheralNamePkg is NOT present, else x
c5 = m if an instance supports it and the processingEntityListPkg is NOT present, else x
c6 = m if an instance supports it and the processingEntityNamePkg is NOT present, else x
c7 = m if a resource can detect usage, else -

Table C.4.1.3 - Attribute Group Support

Index	Attribute Group Template Label	Value of Object Identifier for Attribute Group	Status		Support		Additional Information
			get	settodefault	get	settodefault	
C.4.1.3.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992":state	{ 2 9 3 2 8 1 }	m	x			

Table C.4.1.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.1.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.1.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.1.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.1.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.1.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.1.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.1.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.1.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.1.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.1.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.1.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": stateChange	{ 2 9 3 2 10 14 }	m				C.4.1.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.4.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
							C.4.1.5.4.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.4.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.4.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
							C.4.1.5.4.7	stateChangeDefinition	{ 2 9 3 2 7 28 }	m		
C.4.1.5.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": processingErrorAlarm	{ 2 9 3 2 10 10 }	m				C.4.1.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.1.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.1.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.1.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.1.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.1.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.1.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.1.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.1.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.1.5.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": environmentalAlarm	{ 2 9 3 2 10 3 }	m				C.4.1.5.6.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.6.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.6.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.1.5.6.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.1.5.6.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.6.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.6.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.1.5.6.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.1.5.6.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.1.5.6.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.1.5.6.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.1.5.6.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.1.5.6.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.1.5.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": equipmentAlarm	{ 2 9 3 2 10 4 }	m				C.4.1.5.7.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.1.5.7.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.1.5.7.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.1.5.7.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.1.5.7.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.1.5.7.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.1.5.7.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.1.5.7.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.1.5.7.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.1.5.7.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.1.5.7.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.1.5.7.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.1.5.7.13	trendIndication	{ 2 9 3 2 7 30 }	o		

C.4.2 Connection Oriented Transport Protocol Layer Entity MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":coTransportProtocolLayerEntity	{ 1 3 14 2 2 1 2 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.2.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status create	Status delete	Support create	Support delete	Additional Information
C.4.2.1.1	coTransportProtocolLayer Entity-computerSystem	{ 1 3 14 2 2 3 4 }	computerSystem	o		x	x			
C.4.2.1.2	coTransportProtocolLayer Entity-system	{ 1 3 14 2 2 3 5 }	"Rec. X.721 | ISO/IEC 10165-2 : 1992":system	o		x	x			
C.4.2.1.3	coTransportProtocolLayer Entity-opEquipment	{ 1 3 14 2 2 3 6 }	opEquipment	o		x	x			

Table C.4.2.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status: set by create	Status: get	Status: replace	Status: add	Status: remove	Status: set to default	Support: set by create	Support: get	Support: replace	Support: add	Support: remove	Support: set to default	Additional Information
C.4.2.2.1	manufacturerList	{ 1 3 14 2 2 4 23 }	c3	c3	c3	c3	c3	x							
C.4.2.2.2	manufacturerName	{ 1 3 14 2 2 4 24 }	c4	c4	c4	x	x	x							
C.4.2.2.3	productLabel	{ 1 3 14 2 2 4 45 }	c0	c0	c0	x	x	x							
C.4.2.2.4	"Rec. M.3100 : 1992":version	{ 0 0 13 3100 0 7 52 }	c0	c0	c0	x	x	x							
C.4.2.2.5	serialNumber	{ 1 3 14 2 2 4 50 }	c0	c0	c0	x	x	x							
C.4.2.2.6	typeText	{ 1 3 14 2 2 4 59 }	c0	c0	c0	x	x	x							
C.4.2.2.7	upTime	{ 1 3 14 2 2 4 60 }	x	c0	x	x	x	x							
C.4.2.2.8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingProtocolErrorCounter	{ 2 9 3 2 7 77 }	x	c0	x	x	x	x							
C.4.2.2.9	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingProtocolErrorCounter	{ 2 9 3 2 7 85 }	x	c0	x	x	x	x							
C.4.2.2.10	checksumPDUsDiscardedCounter	{ 1 3 14 2 2 4 3 }	x	c0	x	x	x	x							
C.4.2.2.11	maxPDUSize	{ 1 3 14 2 2 4 26 }	c5	c5	c5	x	x	x							
C.4.2.2.12	"Rec. X.721 | ISO/IEC 10165-2 : 1992": usageState	{ 2 9 3 2 7 39 }	x	c6	x	x	x	x							
C.4.2.2.13	transportEntityType	{ 1 3 14 2 2 4 58 }	x	m	x	x	x	x							
C.4.2.2.14	localTransportAddresses	{ 1 3 14 2 2 4 20 }	x	m	x	x	x	x							
C.4.2.2.15	activeConnections	{ 1 3 14 2 2 4 1 }	x	m	x	x	x	x							
C.4.2.2.16	maxConnections	{ 1 3 14 2 2 4 25 }	x	m	x	x	x	x							

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.2.2.17	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingConnectionRequestsCounter	{ 2 9 3 2 7 82 }	x	m	x	x	x	x							
C.4.2.2.18	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingConnectionRequestsCounter	{ 2 9 3 2 7 74 }	x	m	x	x	x	x							
C.4.2.2.19	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingConnectionRejectErrorCounter	{ 2 9 3 2 7 81 }	x	m	x	x	x	x							
C.4.2.2.20	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingConnectionRejectErrorCounter	{ 2 9 3 2 7 73 }	x	m	x	x	x	x							
C.4.2.2.21	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingDisconnectErrorCounter	{ 2 9 3 2 7 84 }	x	m	x	x	x	x							
C.4.2.2.22	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingDisconnectErrorCounter	{ 2 9 3 2 7 76 }	x	m	x	x	x	x							
C.4.2.2.23	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingDisconnectCounter	{ 2 9 3 2 7 83 }	x	m	x	x	x	x							
C.4.2.2.24	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingDisconnectCounter	{ 2 9 3 2 7 75 }	x	m	x	x	x	x							
C.4.2.2.25	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsSentCounter	{ 2 9 3 2 7 80 }	x	m	x	x	x	x							
C.4.2.2.26	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsReceivedCounter	{ 2 9 3 2 7 78 }	x	m	x	x	x	x							
C.4.2.2.27	"Rec. X.721 | ISO/IEC 10165-2 : 1992": operationalState	{ 2 9 3 2 7 35 }	x	m	x	x	x	x							
C.4.2.2.28	"Rec. X.721 | ISO/IEC 10165-2 : 1992": administrativeState	{ 2 9 3 2 7 31 }	m	m	m	x	x	x							
C.4.2.2.29	"Rec. X.721 | ISO/IEC 10165-2 : 1992": alarmStatus	{ 2 9 3 2 7 32 }	m	m	m	m	m	x							
C.4.2.2.30	coTransportProtocolLayerId	{ 1 3 14 2 2 4 6 }	m	m	x	x	x	x							
C.4.2.2.31	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.2.2.32	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.2.2.33	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							
C.4.2.2.34	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -
c3 = m if an instance supports it and the manufacturerNamePkg is NOT present, else x
c4 = m if an instance supports it and the manufacturerListPkg is NOT present, else x
c5 = m if the "OP1 Library Vol. 2 : 1992":transportConnectionIVMO object class is not used to provide this initial value, else x

c6 = m if resource can detect usage, else -

Table C.4.2.3 - Attribute Group Support

Index	Attribute Group Template Label	Value of Object Identifier for Attribute Group	Status		Support		Additional Information
			get	settodefault	get	settodefault	
C.4.2.3.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992":state	{ 2 9 3 2 8 1 }	m	x			

Table C.4.2.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.2.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.2.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.2.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.2.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.2.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.2.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.2.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.2.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.2.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.2.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.2.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.2.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.2.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.2.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.2.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.2.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.2.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.2.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.2.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.2.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.2.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.2.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.2.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": stateChange	{ 2 9 3 2 10 14 }	m				C.4.2.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.2.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.2.5.4.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.2.5.4.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.2.5.4.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.2.5.4.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
							C.4.2.5.4.7	stateChangeDefinition	{ 2 9 3 2 7 28 }	m		
C.4.2.5.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": processingErrorAlarm	{ 2 9 3 2 10 10 }	m				C.4.2.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.2.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.2.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.2.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.2.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.2.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.2.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.2.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.2.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.2.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.2.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.2.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.2.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		

C.4.3 ConnectionlessNetwork Protocol Layer Entity MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":clNetworkProtocolLayerEntity	{ 1 3 14 2 2 1 3 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.3.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.3.1.1	clNetworkProtocolLayerEntity-computerSystem	{ 1 3 14 2 2 3 7 }	computerSystem	o		x	x			
C.4.3.1.2	clNetworkProtocolLayerEntity-system	{ 1 3 14 2 2 3 8 }	"Rec. X.721 | ISO/IEC 10165-2 : 1992":system	o		x	x			
C.4.3.1.3	clNetworkProtocolLayerEntity-opEquipment	{ 1 3 14 2 2 3 9 }	opEquipment	o		x	x			

Table C.4.3.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.3.2.1	manufacturerList	{ 1 3 14 2 2 4 23 }	c3	c3	c3	c3	c3	x							
C.4.3.2.2	manufacturerName	{ 1 3 14 2 2 4 24 }	c4	c4	c4	x	x	x							
C.4.3.2.3	productLabel	{ 1 3 14 2 2 4 45 }	c0	c0	c0	x	x	x							
C.4.3.2.4	"Rec. M.3100 : 1992":version	{ 0 0 13 3100 0 7 52 }	c0	c0	c0	x	x	x							
C.4.3.2.5	serialNumber	{ 1 3 14 2 2 4 50 }	c0	c0	c0	x	x	x							
C.4.3.2.6	typeText	{ 1 3 14 2 2 4 59 }	c0	c0	c0	x	x	x							
C.4.3.2.7	upTime	{ 1 3 14 2 2 4 60 }	x	c0	x	x	x	x							
C.4.3.2.8	networkEntityType	{ 1 3 14 2 2 4 30 }	x	m	x	x	x	x							
C.4.3.2.9	localNetworkAddresses	{ 1 3 14 2 2 4 19 }	x	m	m	m	m	x							
C.4.3.2.10	nPDUTimeToLive	{ 1 3 14 2 2 4 32 }	m	m	m	x	x	x							
C.4.3.2.11	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusSentCounter	{ 2 9 3 2 7 88 }	x	m	x	x	x	x							
C.4.3.2.12	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusReceivedCounter	{ 2 9 3 2 7 86 }	x	m	x	x	x	x							
C.4.3.2.13	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsSentCounter	{ 2 9 3 2 7 80 }	x	m	x	x	x	x							
C.4.3.2.14	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsReceivedCounter	{ 2 9 3 2 7 78 }	x	m	x	x	x	x							
C.4.3.2.15	pdusForwardedCounter	{ 1 3 14 2 2 4 37 }	x	m	x	x	x	x							
C.4.3.2.16	pdusReasmbldOKCounter	{ 1 3 14 2 2 4 38 }	x	m	x	x	x	x							
C.4.3.2.17	pdusReasmbldFailCounter	{ 1 3 14 2 2 4 39 }	x	m	x	x	x	x							
C.4.3.2.18	pdusDiscardedCounter	{ 1 3 14 2 2 4 40 }	x	m	x	x	x	x							
C.4.3.2.19	"Rec. X.721 | ISO/IEC 10165-2 : 1992": operationalState	{ 2 9 3 2 7 35 }	x	m	x	x	x	x							
C.4.3.2.20	"Rec. X.721 | ISO/IEC 10165-2 : 1992": administrativeState	{ 2 9 3 2 7 31 }	m	m	m	x	x	x							
C.4.3.2.21	"Rec. X.721 | ISO/IEC 10165-2 : 1992": alarmStatus	{ 2 9 3 2 7 32 }	m	m	m	m	m	x							
C.4.3.2.22	clNetworkProtocolLayerId	{ 1 3 14 2 2 4 5 }	m	m	x	x	x	x							
C.4.3.2.23	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.3.2.24	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.3.2.25	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			setbycreate	get	replace	add	remove	settodefault	setbycreate	get	replace	add	remove	settodefault	
C.4.3.2.26	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -
c3 = m if an instance supports it and the manufacturerNamePkg is NOT present, else x
c4 = m if an instance supports it and the manufacturerListPkg is NOT present, else x

Table C.4.3.3 - Attribute Group Support

Index	Attribute Group Template Label	Value of Object Identifier for Attribute Group	Status		Support		Additional Information
			get	settodefault	get	settodefault	
C.4.3.3.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992":state	{ 2 9 3 2 8 1 }	m	x			

Table C.4.3.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.3.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.3.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.3.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.3.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.3.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.3.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.3.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.3.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.3.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.3.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.3.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": stateChange	{ 2 9 3 2 10 14 }	m				C.4.3.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.4.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.3.5.4.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.4.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.4.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
							C.4.3.5.4.7	stateChangeDefinition	{ 2 9 3 2 7 28 }	m		
C.4.3.5.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": processingErrorAlarm	{ 2 9 3 2 10 10 }	m				C.4.3.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.3.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.3.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.3.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.3.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.3.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.3.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.3.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.3.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.3.5.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": communicationsAlarm	{ 2 9 3 2 10 2 }	m				C.4.3.5.6.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.3.5.6.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.3.5.6.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.3.5.6.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.3.5.6.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.3.5.6.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.3.5.6.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.3.5.6.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.3.5.6.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.3.5.6.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.3.5.6.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.3.5.6.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.3.5.6.13	trendIndication	{ 2 9 3 2 7 30 }	o		

C.4.4 OMNIPoint Equipment MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":opEquipment	{ 1 3 14 2 2 1 4 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.4.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status create	Status delete	Support create	Support delete	Additional Information
C.4.4.1.1	"Rec. M.3100 : 1992": equipment-equipment	{ 0 0 13 3100 0 6 10 }	"Rec. M.3100 : 1992":equipment	o		m	m			
C.4.4.1.2	"Rec. M.3100 : 1992": equipment-ManagedElement	{ 0 0 13 3100 0 6 9 }	"Rec. M.3100 : 1992": managedElement	o		m	m			
C.4.4.1.3	opEquipment-computerSystem	{ 1 3 14 2 2 3 10 }	computerSystem	o		m	m			
C.4.4.1.4	opEquipment-system	{ 1 3 14 2 2 3 11 }	"Rec. X.721 | ISO/IEC 10165-2 : 1992":system	o		m	m			
C.4.4.1.5	opEquipment-equipment	{ 1 3 14 2 2 3 12 }	"Rec. M.3100 : 1992":equipment	o		m	m			
C.4.4.1.6	opEquipment-opNetwork	{ 1 3 14 2 2 3 13 }	opNetwork	o		m	m			

Table C.4.4.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.4.2.1	"Rec. M.3100 : 1992":affectedObjectList	{ 0 0 13 3100 0 7 2 }	x	c0	x	x	x	x							
C.4.4.2.2	"Rec. M.3100 : 1992":currentProblemList	{ 0 0 13 3100 0 7 17 }	x	c0	x	x	x	x							
C.4.4.2.3	"Rec. M.3100 : 1992":locationName	{ 0 0 13 3100 0 7 27 }	c0	c0	c0	x	x	x							
C.4.4.2.4	"Rec. M.3100 : 1992":replaceable	{ 0 0 13 3100 0 7 34 }	x	m	x	x	x	x							
C.4.4.2.5	"Rec. M.3100 : 1992":userLabel	{ 0 0 13 3100 0 7 50 }	c0	c0	c0	x	x	x							
C.4.4.2.6	"Rec. M.3100 : 1992":vendorName	{ 0 0 13 3100 0 7 51 }	c0	c0	c0	x	x	x							
C.4.4.2.7	"Rec. M.3100 : 1992":version	{ 0 0 13 3100 0 7 52 }	c14	c14	c14	x	x	x							
C.4.4.2.8	contactList	{ 1 3 14 2 2 4 7 }	c3	c3	c3	x	x	x							
C.4.4.2.9	contactName	{ 1 3 14 2 2 4 8 }	c4	c4	c4	c4	c4	x							
C.4.4.2.10	customerList	{ 1 3 14 2 2 4 11 }	c5	c5	c5	x	x	x							
C.4.4.2.11	customerName	{ 1 3 14 2 2 4 12 }	c6	c6	c6	c6	c6	x							
C.4.4.2.12	functionList	{ 1 3 14 2 2 4 14 }	c7	c7	c7	x	x	x							
C.4.4.2.13	functionName	{ 1 3 14 2 2 4 15 }	c8	c8	c8	c8	c8	x							
C.4.4.2.14	locationPointer	{ 1 3 14 2 2 4 22 }	c9	c9	c9	x	x	x							
C.4.4.2.15	manufacturerList	{ 1 3 14 2 2 4 23 }	c10	c10	c10	x	x	x							
C.4.4.2.16	manufacturerName	{ 1 3 14 2 2 4 24 }	c11	c11	c11	c11	c11	x							
C.4.4.2.17	opNetworkList	{ 1 3 14 2 2 4 34 }	c12	c12	c12	x	x	x							
C.4.4.2.18	opNetworkName	{ 1 3 14 2 2 4 35 }	c13	c13	c13	c13	c13	x							
C.4.4.2.19	productLabel	{ 1 3 14 2 2 4 45 }	c0	c0	c0	x	x	x							
C.4.4.2.20	serialNumber	{ 1 3 14 2 2 4 50 }	c0	c0	c0	x	x	x							
C.4.4.2.21	serviceList	{ 1 3 14 2 2 4 51 }	c15	c15	c15	x	x	x							
C.4.4.2.22	serviceName	{ 1 3 14 2 2 4 52 }	c16	c16	c16	c16	c16	x							
C.4.4.2.23	softwareList	{ 1 3 14 2 2 4 53 }	c17	c17	c17	x	x	x							
C.4.4.2.24	softwareName	{ 1 3 14 2 2 4 54 }	c18	c18	c18	c18	c18	x							
C.4.4.2.25	typeText	{ 1 3 14 2 2 4 59 }	c0	c0	c0	x	x	x							
C.4.4.2.26	"Rec. X.721 | ISO/IEC 10165-2 : 1992": usageState	{ 2 9 3 2 7 39 }	x	c19	x	x	x	x							
C.4.4.2.27	vendorList	{ 1 3 14 2 2 4 61 }	c20	c20	c20	c20	c20	x							
C.4.4.2.28	"Rec. X.721 | ISO/IEC 10165-2 : 1992": operationalState	{ 2 9 3 2 7 35 }	x	m	x	x	x	x							
C.4.4.2.29	"Rec. X.721 | ISO/IEC 10165-2 : 1992": administrativeState	{ 2 9 3 2 7 31 }	m	m	m	x	x	x							

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.4.2.30	"Rec. M.3100 : 1992":alarmStatus	{ 0 0 13 3100 0 7 6 }	x	c	x	x	x	x							
C.4.4.2.31	"Rec. X.721 | ISO/IEC 10165-2 : 1992": availabilityStatus	{ 2 9 3 2 7 33 }	x	m	x	x	x	x							
C.4.4.2.32	"Rec. M.3100 : 1992":equipmentId	{ 0 0 13 3100 0 7 20 }	m	m	x	x	x	x							
C.4.4.2.33	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.4.2.34	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.4.2.35	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							
C.4.4.2.36	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							

```
        c0 = m if an instance supports it, else -
        c1 = m if an object supports allormorphism, else -
        c2 = m if any any registered package (other than this package) has been instantiated, else -
        c3 = m if an instance supports it and the contactNamePkg is NOT present, else x
        c4 = m if an instance supports it and the contactListPkg is NOT present, else x
        c5 = m if an instance supports it and the customerNamePkg is NOT present, else x
        c6 = m if an instance supports it and the customerListPkg is NOT present, else x
        c7 = m if an instance supports it and the functionNamePkg is NOT present, else x
        c8 = m if an instance supports it and the functionListPkg is NOT present, else x
        c9 = m if an instance supports it and the "Rec. M.3100 : 1992":locationNamePackage is NOT present,
else x
        c10= m if an instance supports it and the manufacturerNamePkg is NOT present, else x
        c11= m if an instance supports it and the manufacturerListPkg is NOT present, else x
        c12= m if an instance supports it and the opNetworkNamePkg is NOT present, else x
        c13= m if an instance supports it and the opNetworkListPkg is NOT present, else x
        c14= m if "Rec. M.3100 : 1992":versionPackage is also present and if an instance supports it, else -
        c15= m if an instance supports it and the serviceNamePkg is NOT present, else x
        c16= m if an instance supports it and the serviceListPkg is NOT present, else x
        c17= m if an instance supports it and the softwareNamePkg is NOT present, else x
        c18= m if an instance supports it and the softwareListPkg is NOT present, else x
        c19= m if a resource can detect usage, else -
        c20= m if an instance supports it and the "Rec. M.3100 : 1992":vendorNamePackage is NOT present,
else x
```

Table C.4.4.3 - Attribute Group Support

Index	Attribute Group Template Label	Value of Object Identifier for Attribute Group	Status		Support		Additional Information
			get	settodefault	get	settodefault	
C.4.4.3.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992":state	{ 2 9 3 2 8 1 }	m	x			

Table C.4.4.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.4.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.4.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.4.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.4.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.4.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.4.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.4.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.4.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.4.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.4.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.4.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": stateChange	{ 2 9 3 2 10 14 }	m				C.4.4.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.4.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
							C.4.4.5.4.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.4.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.4.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
							C.4.4.5.4.7	stateChangeDefinition	{ 2 9 3 2 7 28 }	m		
C.4.4.5.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": communicationsAlarm	{ 2 9 3 2 10 2 }	m				C.4.4.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.4.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.4.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.4.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.4.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.4.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.4.5.5.1	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.4.5.5.1	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.4.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.4.5.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": processingErrorAlarm	{ 2 9 3 2 10 10 }	m				C.4.4.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.4.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.4.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.4.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.4.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.4.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.4.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.4.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.4.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.4.5.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": environmentalAlarm	{ 2 9 3 2 10 3 }	m				C.4.4.5.6.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.6.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.6.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.4.5.6.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.4.5.6.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.4.5.6.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.6.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.4.5.6.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.4.5.6.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.4.5.6.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.4.5.6.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.4.5.6.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.4.5.6.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.4.5.8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": equipmentAlarm	{ 2 9 3 2 10 4 }	m				C.4.4.5.7.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.4.5.7.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.4.5.7.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.4.5.7.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.4.5.7.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
							C.4.4.5.7.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.4.5.7.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.4.5.7.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.4.5.7.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.4.5.7.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.4.5.7.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.4.5.7.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.4.5.7.13	trendIndication	{ 2 9 3 2 7 30 }	o		

C.4.5 OMNIPoint Network MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":opNetwork	{ 1 3 14 2 2 1 5 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.5.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.5.1.1	"Rec. M.3100 : 1992": network-network	{ 0 0 13 3100 0 6 17 }	"Rec. M.3100 : 1992": network	o		x	x			
C.4.5.1.2	network-opNetwork-1	{ 1 3 14 2 2 3 14 }	"Rec. M.3100 : 1992": network	o		m	m			
C.4.5.1.3	network-opNetwork-2	{ 1 3 14 2 2 3 15 }	"Rec. M.3100 : 1992": network	o		m	m			
C.4.5.1.4	opNetwork-root	{ 1 3 14 2 2 3 16 }	"Rec. X.600 | ISO/IEC 9834-1 : 1992":root	o		m	m			

Table C.4.5.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.5.2.1	"Rec. M.3100 : 1992":networkId	{ 0 0 13 3100 0 7 3 }	m	m	x	x	x	x							
C.4.5.2.2	"Rec. M.3100 : 1992":userLabel	{ 0 0 13 3100 0 7 50 }	c0	c0	c0	x	x	x							
C.4.5.2.3	networkTitle	{ 1 3 14 2 2 4 31 }	m	m	x	x	x	x							
C.4.5.2.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.5.2.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.5.2.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							
C.4.5.2.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -

Table C.4.5.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
C.4.5.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.5.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.5.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.5.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.5.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.5.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.5.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.5.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.5.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.5.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.5.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.5.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.5.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.5.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.5.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.5.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.5.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.5.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.5.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.5.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.5.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.5.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		

C.4.6 Processing Entity MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":processingEntity	{ 1 3 14 2 2 1 6 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.6.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.6.1.1	"Rec. M.3100 : 1992": equipment-equipment	{ 0 0 13 3100 0 6 10 }	"Rec. M.3100 : 1992":equipment	o		m	m			
C.4.6.1.2	"Rec. M.3100 : 1992": equipment-ManagedElement	{ 0 0 13 3100 0 6 9 }	"Rec. M.3100 : 1992": managedElement	o		m	m			
C.4.6.1.3	opEquipment-computerSystem	{ 1 3 14 2 2 3 10 }	computerSystem	o		m	m			
C.4.6.1.4	opEquipment-system	{ 1 3 14 2 2 3 11 }	"Rec. X.721 | ISO/IEC 10165-2 : 1992":system	o		m	m			
C.4.6.1.5	opEquipment-equipment	{ 1 3 14 2 2 3 12 }	"Rec. M.3100 : 1992":equipment	o		m	m			
C.4.6.1.6	opEquipment-opNetwork	{ 1 3 14 2 2 3 13 }	opNetwork	o		m	m			

Table C.4.6.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.6.2.1	"Rec. M.3100 : 1992":affectedObjectList	{ 0 0 13 3100 0 7 2 }	x	c0	x	x	x	x							
C.4.6.2.2	"Rec. M.3100 : 1992":currentProblemList	{ 0 0 13 3100 0 7 17 }	x	c0	x	x	x	x							
C.4.6.2.3	"Rec. M.3100 : 1992":locationName	{ 0 0 13 3100 0 7 27 }	c0	c0	c0	x	x	x							
C.4.6.2.4	"Rec. M.3100 : 1992":replaceable	{ 0 0 13 3100 0 7 34 }	x	m	x	x	x	x							
C.4.6.2.5	"Rec. M.3100 : 1992":userLabel	{ 0 0 13 3100 0 7 50 }	c0	c0	c0	x	x	x							
C.4.6.2.6	"Rec. M.3100 : 1992":vendorName	{ 0 0 13 3100 0 7 51 }	c0	c0	c0	x	x	x							
C.4.6.2.7	"Rec. M.3100 : 1992":version	{ 0 0 13 3100 0 7 52 }	c14	c14	c14	x	x	x							
C.4.6.2.8	contactList	{ 1 3 14 2 2 4 7 }	c3	c3	c3	x	x	x							
C.4.6.2.9	contactName	{ 1 3 14 2 2 4 8 }	c4	c4	c4	c4	c4	x							
C.4.6.2.10	customerList	{ 1 3 14 2 2 4 11 }	c5	c5	c5	x	x	x							
C.4.6.2.11	customerName	{ 1 3 14 2 2 4 12 }	c6	c6	c6	c6	c6	x							
C.4.6.2.12	functionList	{ 1 3 14 2 2 4 14 }	c7	c7	c7	x	x	x							
C.4.6.2.13	functionName	{ 1 3 14 2 2 4 15 }	c8	c8	c8	c8	c8	x							
C.4.6.2.14	locationPointer	{ 1 3 14 2 2 4 22 }	c9	c9	c9	x	x	x							
C.4.6.2.15	manufacturerList	{ 1 3 14 2 2 4 23 }	c10	c10	c10	x	x	x							
C.4.6.2.16	manufacturerName	{ 1 3 14 2 2 4 24 }	c11	c11	c11	c11	c11	x							
C.4.6.2.17	opNetworkList	{ 1 3 14 2 2 4 34 }	c12	c12	c12	x	x	x							
C.4.6.2.18	opNetworkName	{ 1 3 14 2 2 4 35 }	c13	c13	c13	c13	c13	x							
C.4.6.2.19	productLabel	{ 1 3 14 2 2 4 45 }	c0	c0	c0	x	x	x							
C.4.6.2.20	serialNumber	{ 1 3 14 2 2 4 50 }	c0	c0	c0	x	x	x							
C.4.6.2.21	serviceList	{ 1 3 14 2 2 4 51 }	c15	c15	c15	x	x	x							
C.4.6.2.22	serviceName	{ 1 3 14 2 2 4 52 }	c16	c16	c16	c16	c16	x							
C.4.6.2.23	softwareList	{ 1 3 14 2 2 4 53 }	c17	c17	c17	x	x	x							
C.4.6.2.24	softwareName	{ 1 3 14 2 2 4 54 }	c18	c18	c18	c18	c18	x							
C.4.6.2.25	typeText	{ 1 3 14 2 2 4 59 }	c0	c0	c0	x	x	x							
C.4.6.2.26	"Rec. X.721 | ISO/IEC 10165-2 : 1992": usageState	{ 2 9 3 2 7 39 }	x	c19	x	x	x	x							
C.4.6.2.27	vendorList	{ 1 3 14 2 2 4 61 }	c20	c20	c20	c20	c20	x							
C.4.6.2.28	"Rec. X.721 | ISO/IEC 10165-2 : 1992": operationalState	{ 2 9 3 2 7 35 }	x	m	x	x	x	x							
C.4.6.2.29	"Rec. X.721 | ISO/IEC 10165-2 : 1992": administrativeState	{ 2 9 3 2 7 31 }	m	m	m	x	x	x							

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.6.2.30	"Rec. M.3100 : 1992":alarmStatus	{ 0 0 13 3100 0 7 6 }	x	c	x	x	x	x							
C.4.6.2.31	"Rec. X.721 | ISO/IEC 10165-2 : 1992": availabilityStatus	{ 2 9 3 2 7 33 }	x	m	x	x	x	x							
C.4.6.2.32	"Rec. M.3100 : 1992":equipmentId	{ 0 0 13 3100 0 7 20 }	m	m	x	x	x	x							
C.4.6.2.33	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	c1	c1	x	x	x	x							
C.4.6.2.34	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	m	m	x	x	x	x							
C.4.6.2.35	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	m	m	x	x	x	x							
C.4.6.2.36	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	c2	c2	x	x	x	x							
C.4.6.2.37	addressingSize	{ 1 3 14 2 2 4 2 }	x	c21	x	x	x	x							
C.4.6.2.38	endianess	{ 1 3 14 2 2 4 13 }	x	c21	x	x	x	x							
C.4.6.2.39	cpuUtilization	{ 1 3 14 2 2 4 10 }	x	c0	x	x	x	x							
C.4.6.2.40	memorySize	{ 1 3 14 2 2 4 28 }	x	c21	x	x	x	x							
C.4.6.2.41	memoryUtilization	{ 1 3 14 2 2 4 29 }	x	c0	x	x	x	x							
C.4.6.2.42	upTime	{ 1 3 14 2 2 4 60 }	x	c0	x	x	x	x							
C.4.6.2.43	cpuType	{ 1 3 14 2 2 4 9 }	x	m	x	x	x	x							
C.4.6.2.44	osInfo	{ 1 3 14 2 2 4 36 }	x	m	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -
c3 = m if an instance supports it and the contactNamePkg is NOT present, else x
c4 = m if an instance supports it and the contactListPkg is NOT present, else x
c5 = m if an instance supports it and the customerNamePkg is NOT present, else x
c6 = m if an instance supports it and the customerListPkg is NOT present, else x
c7 = m if an instance supports it and the functionNamePkg is NOT present, else x
c8 = m if an instance supports it and the functionListPkg is NOT present, else x
c9 = m if an instance supports it and the "Rec. M.3100 : 1992":locationNamePackage is NOT present, else x
c10= m if an instance supports it and the manufacturerNamePkg is NOT present, else x
c11= m if an instance supports it and the manufacturerListPkg is NOT present, else x
c12= m if an instance supports it and the opNetworkNamePkg is NOT present, else x
c13= m if an instance supports it and the opNetworkListPkg is NOT present, else x
c14= m if "Rec. M.3100 : 1992":versionPackage is also present and if an instance supports it, else -
c15= m if an instance supports it and the serviceNamePkg is NOT present, else x
c16= m if an instance supports it and the serviceListPkg is NOT present, else x
c17= m if an instance supports it and the softwareNamePkg is NOT present, else x
c18= m if an instance supports it and the softwareListPkg is NOT present, else x
c19= m if a resource can detect usage, else -
c20= m if an instance supports it and the "Rec. M.3100 : 1992":vendorNamePackage is NOT present, else x
c21= m if relevant to the underlying resource, else -

Table C.4.6.3 - Attribute Group Support

Index	Attribute Group Template Label	Value of Object Identifier for Attribute Group	Status		Support		Additional Information
			get	settodefault	get	settodefault	
C.4.6.3.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992":state	{ 2 9 3 2 8 1 }	m	x			

Table C.4.6.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.6.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.6.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.6.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.6.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.6.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.6.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.6.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.6.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.6.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.6.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.6.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": stateChange	{ 2 9 3 2 10 14 }	m				C.4.6.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.4.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.6.5.4.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.4.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.4.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
							C.4.6.5.4.7	stateChangeDefinition	{ 2 9 3 2 7 28 }	m		
C.4.6.5.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": communicationsAlarm	{ 2 9 3 2 10 2 }	m				C.4.6.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.6.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.6.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.6.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.6.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.6.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.6.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.6.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.6.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.6.5.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": processingErrorAlarm	{ 2 9 3 2 10 10 }	m				C.4.6.5.5.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.5.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.5.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.6.5.5.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.6.5.5.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.5.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.5.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.6.5.5.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.6.5.5.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.6.5.5.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.6.5.5.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.6.5.5.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.6.5.5.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.6.5.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": environmentalAlarm	{ 2 9 3 2 10 3 }	m				C.4.6.5.6.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.6.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.6.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.6.5.6.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.6.5.6.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.6.5.6.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.6.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.6.5.6.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.6.5.6.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.6.5.6.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.6.5.6.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.6.5.6.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.6.5.6.13	trendIndication	{ 2 9 3 2 7 30 }	o		
C.4.6.5.8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": equipmentAlarm	{ 2 9 3 2 10 4 }	m				C.4.6.5.7.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.6.5.7.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.6.5.7.3	backUpObject	{ 2 9 3 2 7 41 }	o		
							C.4.6.5.7.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.6.5.7.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
							C.4.6.5.7.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.6.5.7.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.6.5.7.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.6.5.7.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.6.5.7.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.6.5.7.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.6.5.7.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.6.5.7.13	trendIndication	{ 2 9 3 2 7 30 }	o		

C.4.7 Transport Connection MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 1":transportConnection	{ 1 3 14 2 2 1 7 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.7.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status create	Status delete	Support create	Support delete	Additional Information
C.4.7.1.1	transportConnection-coTransportProtocolLayerEntity	{ 1 3 14 2 2 3 17 }	coTransportProtocolLayerEntity	o		x	m			

Table C.4.7.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			setbycreate	get	replace	add	remove	settodefault	setbycreate	get	replace	add	remove	settodefault	
C.4.7.2.1	maxRetransmissions	{ 1 3 14 2 2 4 27 }	x	c0	x	x	x	x							
C.4.7.2.2	retransmissionTime	{ 1 3 14 2 2 4 48 }	x	c0	x	x	x	x							
C.4.7.2.3	retransmissionTimerInitialValue	{ 1 3 14 2 2 4 49 }	x	c0	x	x	x	x							
C.4.7.2.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusRetransmittedErrorCounter	{ 2 9 3 2 7 87 }	x	c0	x	x	x	x							
C.4.7.2.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsRetransmittedErrorCounter	{ 2 9 3 2 7 79 }	x	c0	x	x	x	x							
C.4.7.2.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusRetransmittedErrorThreshold	{ 2 9 3 2 7 102 }	x	c0	c0	x	x	x							
C.4.7.2.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": outgoingProtocolErrorCounter	{ 2 9 3 2 7 85 }	x	c0	x	x	x	x							
C.4.7.2.8	checksumPDUsDiscardedCounter	{ 1 3 14 2 2 4 3 }	x	c0	x	x	x	x							
C.4.7.2.9	localTransportConnectionEndpoint	{ 1 3 14 2 2 4 21 }	x	m	x	x	x	x							
C.4.7.2.10	remoteTransportConnectionEndpoint	{ 1 3 14 2 2 4 47 }	x	m	x	x	x	x							
C.4.7.2.11	transportConnectionReference	{ 1 3 14 2 2 4 57 }	x	m	x	x	x	x							
C.4.7.2.12	localNetworkAddress	{ 1 3 14 2 2 4 18 }	x	m	x	x	x	x							
C.4.7.2.13	remoteNetworkAddress	{ 1 3 14 2 2 4 46 }	x	m	x	x	x	x							
C.4.7.2.14	inactivityTimeout	{ 1 3 14 2 2 4 17 }	x	m	x	x	x	x							
C.4.7.2.15	inactivityTime	{ 1 3 14 2 2 4 16 }	x	m	x	x	x	x							
C.4.7.2.16	maxPDUSize	{ 1 3 14 2 2 4 26 }	x	m	x	x	x	x							
C.4.7.2.17	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusSentCounter	{ 2 9 3 2 7 88 }	x	m	x	x	x	x							
C.4.7.2.18	"Rec. X.721 | ISO/IEC 10165-2 : 1992": pdusReceivedCounter	{ 2 9 3 2 7 86 }	x	m	x	x	x	x							
C.4.7.2.19	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsSentCounter	{ 2 9 3 2 7 80 }	x	m	x	x	x	x							
C.4.7.2.20	"Rec. X.721 | ISO/IEC 10165-2 : 1992": octetsReceivedCounter	{ 2 9 3 2 7 78 }	x	m	x	x	x	x							
C.4.7.2.21	"Rec. X.721 | ISO/IEC 10165-2 : 1992": incomingProtocolErrorCounter	{ 2 9 3 2 7 77 }	x	m	x	x	x	x							
C.4.7.2.22	transportConnectionId	{ 1 3 14 2 2 4 56 }	x	m	x	x	x	x							
C.4.7.2.23	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	x	c1	x	x	x	x							

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			setbycreate	get	replace	add	remove	settodefault	setbycreate	get	replace	add	remove	settodefault	
C.4.7.2.24	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	x	m	x	x	x	x							
C.4.7.2.25	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	x	m	x	x	x	x							
C.4.7.2.26	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	x	c2	x	x	x	x							

c0 = m if an instance supports it, else -
c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -

Table C.4.7.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support conf	Support non	Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
C.4.7.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.7.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.7.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.7.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.7.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.7.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.7.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.7.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.7.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.7.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.7.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.7.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.7.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.7.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.7.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.7.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.7.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.7.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.7.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.7.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.7.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.7.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.7.5.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": communicationsAlarm	{ 2 9 3 2 10 2 }	c0				C.4.7.5.4.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.7.5.4.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.7.5.4.3	backUpObject	{ 2 9 3 2 7 41 }	o		

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
							C.4.7.5.4.4	backedUpStatus	{ 2 9 3 2 7 11 }	o		
							C.4.7.5.4.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.7.5.4.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.7.5.4.7	perceivedSeverity	{ 2 9 3 2 7 17 }	m		
							C.4.7.5.4.8	probableCause	{ 2 9 3 2 7 18 }	m		
							C.4.7.5.4.9	proposedRepairActions	{ 2 9 3 2 7 19 }	o		
							C.4.7.5.4.10	specificProblems	{ 2 9 3 2 7 27 }	o		
							C.4.7.5.4.11	stateChangeDefinition	{ 2 9 3 2 7 28 }	o		
							C.4.7.5.4.12	thresholdInfo	{ 2 9 3 2 7 29 }	o		
							C.4.7.5.4.13	trendIndication	{ 2 9 3 2 7 30 }	o		

c0 = m if instance supports it, else -

C.4.8 Transport Connection IVMO MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 2":transportConnectionIVMO	{ 1 3 14 2 1 1 1 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.8.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.8.1.1	transportConnectionIVMO-coTransportProtocolLayerEntity	{ 1 3 14 2 1 3 1 }	"OP1 Library Vol. 1":coTransportProtocolLayerEntity	o		x	x			

Table C.4.8.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			set by create	get	replace	add	remove	set to default	set by create	get	replace	add	remove	set to default	
C.4.8.2.1	"OP1 Library Vol. 1":inactivityTimeout	{ 1 3 14 2 2 4 17 }	m	m	m	x	x	x							
C.4.8.2.2	"OP1 Library Vol. 1":maxPDUSize	{ 1 3 14 2 2 4 26 }	m	m	m	x	x	x							
C.4.8.2.3	transportConnectionIVMOId	{ 1 3 14 2 1 4 1 }	x	m	x	x	x	x							
C.4.8.2.4	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	x	c1	x	x	x	x							
C.4.8.2.5	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	x	m	x	x	x	x							
C.4.8.2.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	x	m	x	x	x	x							
C.4.8.2.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	x	c2	x	x	x	x							

c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -

Table C.4.8.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
C.4.8.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.8.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.8.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.8.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.8.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.8.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.8.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.8.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.8.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.8.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.8.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.8.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.8.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.8.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.8.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.8.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.8.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.8.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.8.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.8.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.8.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.8.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		

C.4.9 Transport Connection Retransmission IVMO MOCS Proforma

Managed Object class template label	Value of Object identifier for class
"OP1 Library Vol. 2":transportConnectionRetransmissionIVMO	{ 1 3 14 2 1 1 3 }

Are all mandatory features of the class supported? Yes______ No______

Table C.4.9.1 - Name Binding Support

Index	Name Binding Template Label	Value of Object Identifier for Name Binding	Superior Object Class Template Label	Status	Support	Status		Support		Additional Information
						create	delete	create	delete	
C.4.9.1.1	transportConnectionIVMO-coTransportProtocolLayerEntity	{ 1 3 14 2 1 3 1 }	"OP1 Library Vol. 1": coTransportProtocolLayerEntity	o		x	x			
C.4.9.1.2	transportConnectionRetransmissionIVMO-coTransportProtocolLayerEntity	{ 1 3 14 2 1 3 2 }	"OP1 Library Vol. 1": coTransportProtocolLayerEntity	o		x	x			

Table C.4.9.2 - Attribute Support

Index	Attribute Template Label	Value of Object Identifier for Attribute	Status						Support						Additional Information
			setbycreate	get	replace	add	remove	settodefault	setbycreate	get	replace	add	remove	settodefault	
C.4.9.2.1	"OP1 Library Vol. 1":maxRetransmissions	{ 1 3 14 2 2 4 27 }	m	m	m	x	x	x							
C.4.9.2.2	"OP1 Library Vol. 1": retransmissionTimerInitialValue	{ 1 3 14 2 2 4 49 }	m	m	m	x	x	x							
C.4.9.2.3	"OP1 Library Vol. 1":inactivityTimeout	{ 1 3 14 2 2 4 17 }	m	m	m	x	x	x							
C.4.9.2.4	"OP1 Library Vol. 1":maxPDUSize	{ 1 3 14 2 2 4 26 }	m	m	m	x	x	x							
C.4.9.2.5	transportConnectionIVMOId	{ 1 3 14 2 1 4 1 }	x	m	x	x	x	x							
C.4.9.2.6	"Rec. X.721 | ISO/IEC 10165-2 : 1992": allomorphs	{ 2 9 3 2 7 50 }	x	c1	x	x	x	x							
C.4.9.2.7	"Rec. X.721 | ISO/IEC 10165-2 : 1992": nameBinding	{ 2 9 3 2 7 63 }	x	m	x	x	x	x							
C.4.9.2.8	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectClass	{ 2 9 3 2 7 65 }	x	m	x	x	x	x							
C.4.9.2.9	"Rec. X.721 | ISO/IEC 10165-2 : 1992": packages	{ 2 9 3 2 7 66 }	x	c2	x	x	x	x							

c1 = m if an object supports allormorphism, else -
c2 = m if any any registered package (other than this package) has been instantiated, else -

Table C.4.9.5 - Notification Support

Index	Notification Type Label	Value of Notification Type Identifier	Status	Support		Add Info	Sub-Index	Notification Field Name Label	Value of OID of Attr Type associated with Field	Status	Support	Additional Information
				conf	non							
C.4.9.5.1	"Rec. X.721 | ISO/IEC 10165-2 : 1992": attributeValueChange	{ 2 9 3 2 10 1 }	m				C.4.9.5.1.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.9.5.1.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.9.5.1.3	attributeIdentifierList	{ 2 9 3 2 7 8 }	o		
							C.4.9.5.1.4	attributeValueChangeDefinition	{ 2 9 3 2 7 10 }	m		
							C.4.9.5.1.5	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.9.5.1.6	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.9.5.1.7	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.9.5.2	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectCreation	{ 2 9 3 2 10 6 }	m				C.4.9.5.2.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.9.5.2.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.9.5.2.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.9.5.2.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.9.5.2.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.9.5.2.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		
C.4.9.5.3	"Rec. X.721 | ISO/IEC 10165-2 : 1992": objectDeletion	{ 2 9 3 2 10 7 }	m				C.4.9.5.3.1	additionalInformation	{ 2 9 3 2 7 6 }	o		
							C.4.9.5.3.2	additionalText	{ 2 9 3 2 7 7 }	o		
							C.4.9.5.3.3	attributeList	{ 2 9 3 2 7 9 }	o		
							C.4.9.5.3.4	correlatedNotifications	{ 2 9 3 2 7 12 }	o		
							C.4.9.5.3.5	notificationIdentifier	{ 2 9 3 2 7 16 }	o		
							C.4.9.5.3.6	sourceIndicator	{ 2 9 3 2 7 26 }	o		

Annex D (normative)

Management Ensemble Annex

(Refer to the Working Implementation Agreements Document.)

D.1 Introduction

(Refer to the Working Implementation Agreements Document.)

D.2 Systems Management for OSI Transport and Network Layers Ensemble

Editor's Note: [Because the Systems Management for OSI Transport and Network Layers Ensemble is intended to be a self–contained, standalone document, the clauses and subclauses of the Systems Management for OSI Transport and Network Layers Ensemble (as shown here in Annex D.2) are numbered as they would be in a separate, standalone document, and not as they would be according to their position in Annex D.2.]

Systems Management for OSI Transport and Network Layers Ensemble
Draft - 6

Table of Contents

Revision History

Issue 1.0, Draft 6 – September 1993

This is the sixth draft of this Ensemble, submitted as input to the OIW NMSIG meeting held September 13–16. The following summary details the history of this document:

September 1992	Add to OIW NMSIG Working Agreements
December 1992	Review/resolve comments, update WIAs
March 1993	Review/resolve comments, update WIAs
June 1993	Progress to OIW NMSIG Stable Agreements
September 1993	Review/resolve comments, update WIAs

1 Introduction

This Ensemble identifies a specific management problem and provides a solution to that problem. The management problem is specified in terms of a set of requirements and constraints, stated in section 2, while the management solution, stated in section 3, is specified in terms of the resources to be managed and the functions to be applied. Informative scenarios, describing likely interactions, are presented in an annex.

This document is organized as follows:

Section 1 "Introduction", provides a high level overview describing the Ensemble and the structure of the document.

Section 2 "Management Context", identifies the managed resources and management capabilities of the Ensemble.

Section 3 "Ensemble Conformance Requirements", provides or references statements of conformance for this Ensemble.

Section 4 "Attachments", provides a glossary and list of a references.

Annex A This informative annex specifies scenarios which show possible ways of managing information objects to solve the management problem addressed by this Ensemble.

1.1 Unique Identity

In order to provide a unique identity for this ensemble, this ensemble is assigned a registered Object Identifier with the following value:

iso(1) identified–organization(3) oiw(14) nmsig(2) ???

Editor's Note: [To be provided when Ensemble is complete.]

1.2 Scope and Purpose

ISO has defined various Transport and Relay International Standardized Profiles (ISP) that profile Network and Transport layer options. The purpose of this document is to collect management information definitions and profiles and show how they can be applied to solve specific systems management problems (see next paragraph) pertaining to the management of the OSI Transport and Network Layers. The scope of this Ensemble encompasses systems management support for OSI Transport and Network Layer resources specified in the referenced ISPs (see Section 2.2). Note that support for dynamic routing, ISO 10589, has not, to date, been included in an ISP and, therefore, has not been included in this Ensemble. Furthermore, this Ensemble only addresses X.25 DTE management capabilities. Therefore, the X.25 DCE management capabilities and managed objects are not included in this Ensemble.

This Ensemble specifically addresses how to accomplish the following systems management tasks.

– Configure Network or Transport Resources

– Determine current Network or Transport configuration
– Determine logical Network or Transport connectivity
– Monitor Network or Transport traffic
– Detect Network or Trasnport errors and changes
– Receive Network or Transport traffic counts

Although not specifically addressed in this Ensemble, other functions can be supported by the managed objects and the network management ISPs.

This document references the protocols and management information standards and ISPs upon which the Ensemble is based.

2 Management Context

One method of dealing with the complexity of today's networks is to partition managed resources into groups. Ensembles are a method of providing the grouping of sets of resources for the purposes of management. This Ensemble's management context includes the Network and Transport ISP resources to be managed (see Section 3.2.2), and the management tasks (see Section 3.2.1) which can be performed on these resources. This Ensemble groups OSI Transport and Network Layer resources, providing a systems management view, as shown in Figure 1.

```
                        +------------------+
                        |     MANAGER      |
                        +------------------+
                            |    |      |
  +---------------------+   |      +------------------+
  |                          |                         |
  |                          |                         |
  |     ES                   |         IS              |       ES
+----+----------+        +----+---------+         +----+---------+
| M  |   OSI    |        | M  |\   R   /|         | M  |   OSI   |
| G  |Transport |        | G  | \     / |         | G  |Transport|
| M  +----------+        | M  |  \   /  |         | M  +---------+
| T  |   OSI    |        | T  |Sub\ /Sub|         | T  |   OSI   |
|    | Network  |        |    | NW | NW |         |    | Network |
+----+----------+        +----+---------+         +----+---------+
       |   |                       |   |                   |   |
       |   +--------------------+  +--------------------+  |
       |                                                   |
       +---------------------------------------------------+
```

Figure 1. Systems Management for OSI Transport and Network Layers Environment

2.1 Management View

This Ensemble describes systems management of OSI Network and Transport layer resources from the viewpoint of the service provider of the Network and Transport services. This Ensemble addresses specific configuration and fault management aspects pertinent to the protocol entities.

2.2 Resources

This section specifies the resources addressed by this Ensemble. If a system supports one or more of the following ISPs, this Ensemble indicates how the fault and configuration management requirements identified in section 1.2 can be applied to the Transport and Network resources addressed in the following ISPs.

TA51	CO–TS over CL–NS in LAN with CSMA/CD
TA1111/21	CO–TS over CL–NS in PSDN
TB1111/21	CO–TS (Class 0, 2, 4) over CO–NS in PSDN
TC1111/21	CO–TS (Class 0, 2) over CO–NS in PSDN
TD1111/21	CO–TS (Class 0) over CO–NS
TE1111/21	CO–TS (Class 2) over CO–NS
RA51.1111/21	CL–NS Network Layer Relay

The specific Transport and Network resources included in these ISPs, and, thereore, included in this Ensemble are listed in Figure 2.

TA51	TA1111/21	TB1111/21	TC1111/21	TD1111/21	TE1111/21	RA51.1111/21	
IS 8073 CO-TS Class 4/0	IS 8073 CO-TS Class 4/0	IS 8073 CO-TS Class 4/2/0	IS 8073 CO-TS Class 2/0	IS 8073 CO-TS Class 0	IS 8073 CO-TS Class 2		
IS 8473 CL-NS IS 9542 ES-IS	IS 8473 CL-NS IS 8208 IS 9542 ES-IS	IS 8208 CO-NS	IS 8208 CO-NS	IS 8208 CO-NS	IS 8208 CO-NS	IS 8473 CL-NS IS 9542 ES-IS	IS 8208 CO-NS

Figure 2. Resources Addressed By This Ensemble

2.3 Functions

This section specifies the management functions that can be performed on the resources described in section 2.2.

Object Management

In the context of this Ensemble, Object Management provides configuration monitoring, control, and change notification services for OSI Layer 3–4 protocol entities and configuration/routing information. This function might be used, for example, to acquire topological information to aid in depicting OSI end systems and intermediate systems, to create, delete, or modify routing information, to retrieve network/transport address information, or to configure packet segmentation and reassembly mechanisms.

State Management

In the context of this Ensemble, State Management provides status monitoring, control, and change notification services for OSI Layer 3–4 protocol entities and configuration/routing information. This function might be used, for example, to provide current state information for OSI end systems and intermediate systems displayed on topological information, to identify routes which are currently in use, or to provide administrative control over the usage of specific protocol entities.

Alarm Reporting

In the context of this Ensemble, Alarm Reporting provides notification of faults, protocol errors, quality of service degradation, etc., relating to OSI Layer 3–4 protocol entities and configuration/routing information. This function might be used, for example, to detect and signal crossing of established threshold criteria for QOS, or errors relating to layer protocol or routing entities.

Event Report Management

In the context of this Ensemble, Event Report Management provides administrative control and filtering (discrimination) of alarms and configuration change event reports relating to OSI Layer 3–4 protocol entities and configuration/routing information. This function might be used, for example, to start or stop forwarding of selected alarms to a specific application responsible for fault management of a given OSI subnetwork.

Log Control

In the context of this Ensemble, Log Control provides for administrator–controlled storage and subsequent retrieval of alarms and configuration change log records relating to OSI Layer 3–4 protocol entities and configuration/routing information. This function might be used, for example, to retrieve previously stored configuration change information necessary to diagnose a routing fault.

3 Ensemble Conformance Requirements

This section defines the overall Ensemble conformance requirements.

3.1 General Conformance Requirements

All the conformance requirements identified in this part of the document are based on the referenced base standards and profile conformance requirements, clause 18.9 of the OIW Implementors' Agreements, and the "Ensemble Concepts and Format" [ENSCON] specification.

An implementation supporting this Ensemble must prove conformance to:

– all the managed object classes representing resources within the scope specified by this Ensemble (see Section 3.2.2), and

– all the management functionality of the Ensemble resources (see Section 3.2.1).

The supplier of an implementation that claims conformance to this Ensemble must complete tables 1, 2, and 3, indicating which role (manager/agent) the implementation supporting this Ensemble adopts.

3.2 Specific Conformance Requirements

This section presents the specific conformance requirements for this Ensemble. The relationship of Ensemble conformance to OSI Management Functions ISP conformance is discussed, and Ensemble function support requirements are presented.

3.2.1 OSI Management Functions Profiles Conformance

Table 1 lists all the ISPs relevant to this Ensemble and identifies which of these profiles is required to be supported when the implementation adopts a manager or an agent role. These ISPs specify the management functionality to be supported and, therefore, specify what management functionality can be applied to the various resources designated in Section 3.2.2.

The following notation convention has been used:

m defines a mandatory requirement
– stands for out of scope

Table 1. Ensemble Functional ISP Conformance Requirements

ISP Supported	Manager Role	Agent Role
AOM211 - General Management Capabilities	c1	c2
AOM221 - General Event Report Management	c1	c2
AOM231 - General Log Control	c1	c2

c1: m if a system claims conformance to the ISP in the manager role, else –
c2: m if a system claims conformance to the ISP in the agent role, else –

As can be seen from Table 1, if the implementor claims conformance to the manager role, the implementation shall support the manager role capabilities of all three ISPs (i.e., AOM211, AOM221 and AOM231). Likewise, if conformance is claimed to the agent role, the implementation shall support the agent role capabilities of all three ISPs (i.e., AOM211, AOM221 and AOM231).

3.2.2 Managed Object Conformance

Table 2 lists all the management information required to be supported when an implementation claims conformance to this Ensemble. The table shows which document specifies management information for each of the relevant resources.

Table 2. Ensemble Management Information Conformance Requirements

Management Information Model	Managed Resources
OSI Network Layer OSI Transport Layer Management Function Support Superclasses for OSI Layers	ISO/IEC 10733 ISO/IEC 10737-1 ISO/IEC 10165-2 ISO/IEC 10165-5

System namebinding is out–of–scope for this Ensemble.

The following table lists, for each managed object included in this Ensemble from the base standards, which profile(s) address that object.

Object Class	Relevant Profiles	Notes
networkSubsystem	All	
networkEntity	All	
nSAP	All	
cLNS	TA51,TA1111/21, RA51.1111/21	
linkage	All	(1)(3)(4)(5)
cONS	TB1111/21, TC1111/21, TD1111/21, TE1111/21	
networkConnection	TB1111/21, TC1111/21, TD1111/21, TE1111/21	
x25PLE	All except TA51	
x25PLE-DTE	All except TA51	
x25PLEIVMO	All except TA51	(6)
x25PLEIVMO-DTE	All except TA51	(6)
virtualCircuit	All except TA51	
virtualCircuit-DTE	All except TA51	
virtualCallIVMO	All except TA51	(2)(6)
virtualCall-DTE	All except TA51	
permanentVirtualCall-DTE	All except TA51	
transportSubsystem	All	
transportEntity	All	
comodePM	All	
tSAP	All	
transportConnection	All	(7)(8)(9)
transportConnectionIVMO	All	(6)7)(8)(9)

Object Class	Relevant Profiles	Notes
alarmRecord	All	(10)
attrValueChangeRecord	All	(10)
discriminator	All	(11)
eventFwdingDiscrim	All	(11)
eventLogRecord	All	(12)
log	All	(12)
logRecord	All	(12)
objectCreationRecord	All	(10)
objectDeletionRecord	All	(10)
stateChangeRecord	All	(10)
system	All	
top	All	
subsystem	All	
communicationsEntity	All	
coProtocolMachine	All	
clProtocolMachine	TA51, TA1111/21, RA51.1111/21	
sap1	All	
sap2	All	
singlePeerConnection	All	
commInfoRecord	All	

NOTES:

(1) Name Bindings differ for CL-NS and CO-NS T-Profiles
(2) receiving/transmittingWindowRotationRecoveryProcedures-P packages apply only to TA/TB/TC1111/21
(3) linkage-ISO9542IS-P and linkage-ISO9542ES-P packages apply only to TA51
(4) linkage-ISO8473-ISO8208SNDCF-P package applies only to TA1111/21 and RA51.1111/21
(5) linkageIdleTimer-P, linkageInitialMinimumTimer-P, and linkageReserveTimer-P packages apply to all profiles except TA51
(6) IVMOs are relevant to the profile, but are not "mandatory"
(7) class0-P package applies only to TD1111/21
(8) class2-P package applies only to TB, TC1111/21 and TE1111/21
(9) class4-P package applies only to TA and TB 1111/21, TA51
(10) These MOs are covered by AOM211 profile
(11) These MOs are covered by AOM221 profile
(12) These MOs are covered by AOM231 profile

Table 3. Managed Object Conformance Statement Summary

Index Info	Resource	Standard	MO Class Label of Proforma	Base Std	Profile Status	Addtl Info
1.0	Network Subsystem	10733	networkSubsystem	m	m	
2.0	Network Entity	10733	networkEntity	c4	m	
3.0	NSAP	10733	nSAP	c4	m	
4.0	Connectionless NS	10733	cLNS	c1	c7	
5.0	Linkage	10733	linkage	c4	m	
6.0	Connection-Oriented NS	10733	cONS	c2	c8	
7.0	Network Connection	10733	networkConnection	c2	c8	
8.0	X.25 PLE	10733	x25PLE	c5	c9	
9.0	X.25 PLE DTE	10733	x25PLE-DTE	c3	c9	
10.0	X.25 PLE IVMO	10733	x25PLEIVMO	o	c9	
11.0	X.25 PLE IVMO DTE	10733	x25PLEIVMO-DTE	o	c9	
12.0	Virtual Circuit	10733	virtualCircuit	c5	c9	
13.0	Virtual Circuit DTE	10733	virtualCircuit-DTE	c3	c9	
14.0	Virtual Call IVMO	10733	virtualCallIVMO	o	c9	
15.0	Virtual Call DTE	10733	virtualCall-DTE	o	c9	
16.0	PVC DTE	10733	permVirtualCall-DTE	o	c9	
17.0	Transport Subsystem	10737	transportSubsystem	m	m	

Index Info	Resource	Standard	MO Class Label of Proforma	Base Std	Profile Status	Addtl Info
18.0	Transport Entity	10737	transportEntity	c6	m	
19.0	CO Mode Protocol Machine	10737	comodePM	c6	m	
20.0	TSAP	10737	tSAP	c6	m	
21.0	Transport Connection	10737	transportConnection	c6	m	
22.0	Transport Connection IVMO	10737	transportConnIVMO	o	m	
23.0	Alarm Record	10165-2	alarmRecord	o	m	
24.0	Attr Value Change Record	10165-2	attrValueChangeRec	o	m	
25.0	Discriminator	10165-2	discriminator	o	m	
26.0	EFD	10165-2	eventFwdDiscrim	o	m	
27.0	Event Log Record	10165-2	eventLogRecord	o	m	
28.0	Log	10165-2	log	o	m	
29.0	Log Record	10165-2	logRecord	o	m	
30.0	Object Creation Record	10165-2	objectCreationRec	o	m	
31.0	Object Deletion Record	10165-2	objectDeletionRec	o	m	
32.0	State Change Record	10165-2	stateChange Record	o	m	
32.0	System	10165-2	system	o	m	
33.0	Top	10165-2	top	o	m	

Index Info	Resource	Standard	MO Class Label of Proforma	Base Std	Profile Status	Addtl Info
34.0	Subsystem	10165-5	subsystem	o	m	
35.0	Communications Entity	10165-5	commEntity	o	m	
36.0	CO Protocol Machine	10165-5	coProtocolMachine	o	c2	
37.0	CL Protocol Machine	10165-5	clProtocolMachine	o	m	
38.0	SAP (Format 1)	10165-5	sap1	o	m	
39.0	SAP (Format 2)	10165-5	sap2	o	m	
40.0	Single Peer Connection	10165-5	singlePeerConn	o	m	
41.0	Communication Info Record	10165-5	communicationsInformationRecord	o	m	

c1: m if system claims conformance to 10733 CLNS management, else –
c2: m if system claims conformance to 10733 CONS management, else –
c3: m if system claims conformance to 10733 X.25 DTE management, else –
c4: m if system claims conformance to 10733 CLNS or CONS management, else –
c5: m if system claims conformance to 10737
c6: m if system claims conformance to IS 8073 management, else –
c7: m if system claims conformance to TA51, TA1111/21, or RA51.1111/21 management, else–
c8: m if system claims conformance to TB1111/21 or TC1111/21 management, else –
c9: – if system claims conformance to TA51 management, else m
c10: m if system claims conformance to RA51.1111/21 management, else –

3.2.3 Management Capability Support/SMFUs Support

This Ensemble references the AOM 221 ISP for the required management capability/SMFU services.

3.2.4 MOCS Proforma for Ensemble Managed Object Classes

The MOCS specified in the base standards, ISO/IEC 10733 and ISO/IEC 10737–1, shall be supported.

3.2.5 Association Initiator/Responder

This Ensemble references the AOM 221 ISP for the required ACSE services.

3.2.6 CMIS Services (CMIP PDU) Requirements

This Ensemble references the AOM 221 ISP for the required CMIS services.

4 Attachments

4.1 Glossary

CL–NS	Connectionless Network Service
CO–NS	Connection–Oriented Network Service
CO–TS	Connection–Oriented Transport Service
ES	End System
IS	Intermediate System
ISP	International Standardized Profile
LAN	Local Area Network
MOCS	Managed Object Conformance Statement
NW	Network
PICS	Protocol Implementation Conformance Statement
PSDN	Packet Switch Data Network
R	Relay
RA51.1111/21	CL–NS Network Layer Relay
TA1111/21	CO–TS over CL–NS in PSDN
TA51	CO–TS over CL–NS in LAN with CSMA/CD
TB1111/21	CO–TS (Class 0, 2, 4) over CO–NS in PSDN
TC1111/21	CO–TS (Class 0, 2) over CO–NS in PSDN
TD1111/21	CO–TS (Class 0) over CO–NS
TE1111/21	CO–TS (Class 2) over CO–NS

4.2 Reference List

This section identifies the documents which are referenced by this Ensemble.

Editor's Note: [This clause might need to be merged with the references clause for Chapter 18 of the OIW Implementors' Agreements, to avoid duplication of references. Formal references with complete titles will be provided to the OIW NMSIG IA editor. Documents currently referenced by this Ensemble include the following:]

Forum 025	Ensemble Concepts and Format
ISO/IEC 10165–4	Guidelines for the Definition of Managed Objects
ISO/IEC 10165–6	Guidelines for MOCS Proforma
ISO/IEC 9595	CMIS (1991)
ISO/IEC 9596–1	CMIP (1991)
ISP 11183–2	AOM12: Enhanced Management Communication
ISO/IEC 10164–1	Object Management Function
ISO/IEC 10164–2	State Management Function
ISO/IEC 10164–4	Alarm Reporting Function
ISO/IEC 10164–5	Event Report Management Function
ISO/IEC 10164–6	Log Control Function
ISP 12059–1	AOM211: General Management Capability
ISP 12059–2	AOM221: General Event Report Management
ISP 12059–3	AOM231: General Log Control
ISO/IEC 8073	Connection–Oriented Transport Protocol
ISO/IEC 8473	Connectionless Network Protocol
ISO/IEC 9542	OSI ES–IS Routing Protocol
ISO/IEC 8208	X.25 Packet Layer Protocol for Data Terminal Equipment
ISP TA1111/21	CO–TS over CL–NS in PSDN
ISP TA51	CO–TS over CL–NS in LAN with CSMA/CD
ISP RA51.1111/21	CL–NS Network Layer Relay
ISP TB1111/21	CO–TS (Class 0, 2, 4) over CO–NS in PSDN
ISP TC1111/21	CO–TS (Class 0, 2) over CO–NS in PSDN
ISO/IEC 10733	Management Info for OSI Network Layer
ISO/IEC 10737	Management Info for OSI Transport Layer
ISO/IEC 10165–2	Definition of Management Information
ISO/IEC 10165–5	Generic Management Information

Annex A (Informative)

A. Scenarios

This Annex defines the Ensemble scenarios. Each of these definitions consists of a brief textual description and message flow diagrams.

Scenarios are strictly informative. Each scenario is an example of one possible way to show how the managed objects in the information model can be used. This section is to supply the reader additional information that facilitates understanding of how the Ensemble can be applied to real world situations.

In the scenarios that follow, CMIP flows between (and corresponding CMIS primitives within) manager and agent systems are indicated by arrows with a three character abbreviation for request (Req), indicate (Ind), response (Rsp), and confirm (Cnf) primitives shown at the head and tail of the arrow. For example:

```
o––Req––––––––––––––––––––––––Ind––>
        CMIS request
<––Cnf––––––––––––––––––––––––Rsp––o
        CMIS response
```

There are many possible scenarios which can be applied to the Systems Management for OSI Transport and Network Layers Ensemble. The following scenarios have been selected as examples.

Derive Current Network and/or Transport Layer Path Connectivity

> The first step in performing almost any service provider view management task is likely to involve determining the current connectivity of the managed network. End–to–End transport connectivity and network path connectivity are included in this Ensemble.

Reconfigure Network and/or Transport Layer

> Almost any proactive service provider management of the OSI network will involve some sort of reconfiguration (for example, deactivating a malfunctioning protocol entity, or tuning performance–related parameters to improve QOS).

Monitor Network and/or Transport Layer Changes

> It is expected that many service provider management systems will simply monitor the managed network, displaying changes to connectivity, status, and resource configuration.

"Preconfigure" Network and/or Transport Layer Connections

> Most OSI Network and/or Transport layer products provide configurable parameters which can be set to provide default values for connection characteristics such as maxPDUSize or retryLimit. This capability can also be provided remotely in a distributed management environment.

These scenarios are further expanded in the following subsections. Each provides a detailed step–by–step discussion of how these tasks can be performed using the management capability provided by this Ensemble. The specified calls are the pass through (PT) calls as the functions performed are users of the CMISE service. Only relevant parameters or options are specified. This is to enable the reader to focus on the salient points of the discussion.

A.1 Relevant Information for Management Operations

This section identifies the managed object and relevant packages and attributes of those managed objects which are to be used in one or more of the following scenarios. This information has been gathered in a seperate sub–section for the convenience of the reader in identifying the scope of all pertinent information.

A.1.1 Connectivity Information

A.1.1.1 Transport Layer

Managed Object and Package Names	Attribute Names
transportSubsystem	subsystemId
transportEntity	actualNSAP, targetNSAP,
comodeTPM	openConnections, localSucccessfulConnections, remoteSuccessfulConnections,
transportConnection	localReference, remoteReference, callingTSelector, calledTSelector, callingNSAPAddress, calledNSAPAddress, respondingNSAPAddress, connectionDirection, networkConnectionIDs

A.1.1.2 Network Layer

Managed Object and Package Names	Attribute Names
networksubsystem	sap1Address, actualNSAP, userEntity, subsystemId
networkEntity	networkEntityTitles, systemTypes
nSAP	sap2Address
cLNS	clProtocolMachineId, operationalSystemType
linkage	linkageId, sN-ServiceProvider, sN-SAP, operationalProtocols
linkageCODLService -P-PACKAGE	callsPlaced
linkage-ISO8473 -ISO8208SNDCF-P-	callsPlaced

-PACKAGE	
linkage-ISO9542ES-P PACKAGE	iSO9542OperationalSubsets, manualISSNPAAddress
linkage-ISO9542IS-P -PACKAGE	iSO9542OperationalSubsets
cONS	coProtocolMachineId, operationalSystemType
x25PLE	localDTEAddress, sN-ServiceProvider, sN-SAP, logicalChannelAssignements
virtualCircuit	virtualCircuitId, logicalChannel
virtualCall-DTE	callingAddressExtension, calledAddressExtension, direction, originallyCalledAddress, remoteDTEAddress

A.1.2 Configuration Information

A.1.2.1 Transport Layer

Managed Object and Package Names	Attribute Names
transportSubsystem	subsystemId
clmodeTPM	clProtocolMachineId, clChecksumOption
coProtocolMachineId	coProtocolMachineId
transportConnection	protocolClass, maxTPDUSize

For each of the Transport protocol classes, the appropriate Conditional Package is applicable as delineated in ISO 10737.

A.1.2.2 Network Layer

Managed Object and Package Names	Attribute Names
networkEntity	systemTypes
cLNS	clProtocolMachineId, supportedProtocols
cLNS8473-P PACKAGE	maximumLifetime, enableChecksum
linkage	linkageId, operationalProtocols

linkaage- -ISO9542Checksum-P- -PACKAGE	enableChecksum
linkage-ISO9542ES-P -PACKAGE	iSO9542OperationalSubsets holdingTimeMultiplier, defaultESConfigurationTimer, activeESConfigurationTimer
linkage-ISO9542IS-P -PACKAGE	iSO9542OperationalSubsets, holdingTimeMultiplier, iSConfigurationTimer, suggestedESConfigurationTimer, redirectHoldingTime
cONS	coProtocolMachineId, operationalSystemType
x25PLE	protocolVersionSupported, localDTEAddress, x25PLEMode, defaultThroughputClasses, flowControlParameterNegotiation, defaultPacketSizes, defaultWindowSizes, throughputClassNegotiation, logicalChannelAssignements
x25PLE-DTE	callDeflectionSubscription, callRequestResponseTimer, extendedPacketSequenceNumbering, maxActiveCircuits, minimumRecallTimer, resetRequestResponseTimer, restartRequestResponseTimer, clearRequestResponseTimer, interruptResponseTimer
packetRetransmission Procedures-P PACKAGE	rejectResponseTimer PACKAGE
receivingWindow RotationRecovery Procedures-P PACKAGE	windowStatusTransmissionTimer
transmittingWindow RotationRecovery Procedures-P PACKAGE	windowRotationTimer
onlineRegistration -P PACKAGE	registrationRequestResponseTimer, registrationPermitted
permanentVirtual- -Circuit-DTE	logicalChannel, packetSizes, throughputClasses, windowSizes

A.2 Determine Current Network/Transport Layer Connectivity

A.2.1 Obtain layer subsystem configuration

Obtain layer subsystem configuration by having the Manager send to each Agent in its domain a request for relevant attributes.

To promote readability, the local relative distinguished name (RDN) has been used to identify the MOInstance. In actual practice, the full distinguished name (DN) is expected to be used.

A.2.1.1 For Transport Connectivity Configuration Information

```
o--Req----------------------Ind-->
  PT-GET (MOClass: transportSubsystem,
  MOInstance: subsystemID="TransportSubsystem",
   Scope: Transport Entity, COTP, Transport Connection,
   AttrIdList: {See list for Connectivity in A.1 above})
```

Agent receives PT-GET Ind, returns a PT-GET Response containing all or selected attributes for every MO instance contained in NW System(s)

```
<--Cnf----------------------Rsp--o
  PT-GET (MOClass: transportSubsystem,
  MOInstance: subsystemID="TransportSubsystem",
  AttrIdList: {See list for Connectivity in A.1 above})
```

Manager receives PT-GET Confirm and accumulates managed object instance/attribute information.

A.2.1.2 For Network Connectivity Configuration Information

```
o--Req----------------------Ind-->
  PT-GET (MOClass: networkSubsystem,
  MOInstance: subsystemID="NetworkSubsystem",
  Scope: subsystemId,
  AttrIdList: {See list for Connectivity in A.1 above})
```

Agent receives PT-GET Ind, returns a PT-GET Response containing all or selected attributes for every MO instance contained in NW System(s)

```
<--Cnf----------------------Rsp--o
  PT-GET (MOClass: networkSubsystem,
  MOInstance: subsystemID="NetworkSubsystem",
  AttrIdList: {See list for Connectivity in A.1 above})
```

Manager receives PT-GET Confirm and accumulates managed object instance/attribute information.

A.2.2 Determine current connectivity

Use addressing information and relationship attributes to determine layer 3–4 connectivity. For example:

nSAP's userEntity attribute identifies transportEntity

transportEntity's actualNSAP attribute identifies nSAP

transportConnection's underlyingConnectionNames identifies nSAP of peer system

Editor's Note: [For examples of the use of relationship attributes see ANNEX C of ISO/IEC 10733 and/or ISO/IEC 10737.]

A.3 Reconfigure Network and/or Transport Layer

In order to reconfigure network and/or transport layer resources, the following CMIS sequence could occur.

A.3.1 Identify type of configuration

This step identifies a selection of possible configuration changes which are covered in this scenario. Select one step from the following for the type of change to be made:

(1) Deactivate comodePM, cLNS, cONS, X25PLE–DTE

(2) Modify Configurable Parameters

(3) Activate comodePM, cLNS, cONS, X25PLE–DTE

(4) Preconfigure Connection

A.3.1.1 Request deactivate for affected resources

Request deactivate for all affected resources:

o––Req––––––––––––––––––––––Ind––>

PT_ACTION((MOClass = { see (A.3.1, list item 1) },
MOInstance = { target },
ActionId = { see (A.3.1, list item 1) })

Agent receives PT–ACTION deactivates, or shuts down protocol machine as requested

<––Cnf––––––––––––––––––––––Rsp––o

PT_ACTION((MOClass = { see (A.3.1, list item 1) },
MOInstance = { target },
ActionId = { see (A.3.1, list item 1) })

Manager receives PT–ACTION Confirmation

A.3.1.2 Modify configurable parameters

Manager sends PT–SET Request parameter modification

o––Req––––––––––––––––––––––Ind––>

PT_SET(MOclass = {any class,
MOInstance = { target },
AttrList = { param(s) to be changed and value(s)})

Agent receives PT–SET Indication, modifies attribute value(s), returns response

<––Cnf––––––––––––––––––––––Rsp––o

```
PT_SET(MOClass  = {any class,
 MOInstance = { target },
 AttrList = { param(s) to be  changed and  value(s)})
```

A.3.1.3 Request activate for affected resources

Request activate for all affected resources:

```
        o––Req–––––––––––––––––––––––––Ind––>

PT_ACTION((MOClass = { see (A.3.1, list item 3) },
 MOInstance = { target },
 ActionId = { see (A.3.1, list item 3) })
```

Agent receives PT–ACTION, activates or starts protocol machine as requested

```
        <––Cnf–––––––––––––––––––––––––Rsp––o

PT_ACTION((MOClass = { see (A.3.1, list item 1) },
 MOInstance = { target },
 ActionId = { see (A.3.1, list item 1) })
```

Manager receives PT–ACTION Confirmation

A.3.1.4 Preconfigure connection

The Agent instantiates a new connection MO with initial attribute values supplied by corresponding IVMO MO.

"Preconfigured" values can also be modified or deleted altogether. These steps are similar to those defined in A.3, Reconfigure, where the target MO is an IVMO.

A.4 Monitor Network/Transport Layer Changes

There are several ways in which to monitor changes to network resources, depending upon the needs of the application and of the management environment. For example: polling, real time monitoring, or off–line monitoring. These methods are not mutually exclusive, and would typically be combined to achieve the desired mix of reaction time and network load.

A.4.1 Polling

The manager can poll the agent periodically to detect changes in the configuration. This can be accomplished by following steps as described in A.2.2, Determine Connectivity, and using the appropriate attributes.

A.4.2 Real Time Monitor

The manager can request to receive event reports whenever changes of interest occur in the managed network. This scenario is described below.

To perform real–time monitoring, the following CMISE sequence occurs:

A.4.2.1 Identify resources of interest

– MO Class(es)
– MO Instance(s)
– NotificationIds
– AttributeIds
– Attribute Values

A.4.2.2 Event forwarding for notifications

Regarding event forwarding for notifications of interest, a manager requesting event forwarding sends a PT–CREATE Request for notifications of interest.

```
        o––Req–––––––––––––––––––––––––Ind––>

     M_CREATE(MOClass = DMI:EFD,
      AttrIdList = selection of
          – discriminatorConstruct
          – adminState
          – destination
          – other options)

     <--Ind------------------------Req--o

     PT_EVENT_REPORT(ObjectCreation)
```

Agent receives PT–CREATE, sets up EFD MO instance with desired values, returns PT–CREATE Response

```
        <––Cnf–––––––––––––––––––––––––Rsp––o

     M_CREATE(MOClass = DMI:EFD,
      AttrIdList  =
          – discriminatorConstruct
          – adminState
          – destination
          – other options
```

Manager receives PT–CREATE Confirm.

A.4.2.3 Monitor changes received as incoming events

An Agent detects a notification, analyzes the EFDs, then forwards appropriate event report for all notifications which evaluate to TRUE:

For example, an Agent sends one of the following:

```
o––Req––––––––––––––––––––––––Ind––>
```

PT–EVENT–REPORT(objectCreation)
PT–EVENT–REPORT(objectDeletion)
PT–EVENT–REPORT(stateChange)
PT–EVENT–REPORT(communicationsAlarm)

Manager receives Indication. Real–Time monitoring can also be suspended, resumed, or terminated, and the forwarding conditions can be modified. These steps are similar to those defined in A.3, Reconfigure, where the target MO is an EFD.

A.4.3 Off-Line Monitor

The manager can request to log event records whenever changes of interest occur in the managed network. This scenario is also described below. To perform off–line monitoring, the following sequence could occur over the interoperable interface:

A.4.3.1 Identify notifications and select target

Establish criteria by which notifications shall be retrieved from log reports, with the following possible selection criteria:

– MO Class(es)
– MO Instance(s)
– NotificationIds
– AtributesIds
– Attribute Values

A.4.3.2 Logging notifications of interest

A manager requesting logging sends a PT–CREATE Request for notifications of interest

```
o––Req––––––––––––––––––––––––Ind––>
```

PT_CREATE((MOClass = DMI:Log) AttrIdList includes:
– discriminatorConstruct
– adminState
– other options)

Agent receives PT–CREATE, sets up Log MO instance with desired values, returns PT–CREATE Response

```
<––Cnf––––––––––––––––––––––––Rsp––o
```

PT_CREATE((MOClass = DMI:Log) AttrIdList includes:
– discriminatorConstruct
– adminState
– other options)

```
<--Ind-----------------------Req--o

PT_EVENT_REPORT(ObjectCreation)
```

Manager receives PT–CREATE Confirm

A.4.3.3 Detecting Log changes

An agent detecting a log change notification, analyzes Logs, then instantiates appropriate event records for all notifications which evaluate to TRUE:

For example Agent creates one or more of the following:
- – objectCreationRecord
- – objectDeletionRecord
- – stateChangeRecord
- – communicationAlarmRecord

A.4.3.4 Retrieving logged notifications

A manager desiring to retrieve logged notifications sends a PT–GET Request from each agent in the domain. The manager then sends a get request for the log information:

```
o––Req–––––––––––––––––––––––Ind––>

PT_GET(MOClass = Log,MOInstance = { target },
   Scope = Entire Subtree, Filter = {optional})
```

Agent receives PT–GET Ind, returns a PT–GET Response containing all or selected eventRecords contained in the {target} log. The Manager receives the PT–GET Confirmation containing eventRecord(s). Repeat for all agents/logs as desired.

```
<––Cnf–––––––––––––––––––––––Rsp––o

PT_GET(MOClass = Log,MOInstance = { target },
    Scope = Entire Subtree, Filter = {optional})
```

Off–line monitoring can also be suspended, resumed, or terminated, and the logging conditions can be modified. These steps are similar to those defined in A.3, Reconfigure, where the target MO is a Log. Logs can also be emptied by having the Manager send the Agent a PT–DELETE request.

```
o––Req–––––––––––––––––––––––Ind––>

PT_DELETE (MOClass = LogRecord, MOInstance = null)

<––Cnf–––––––––––––––––––––––Rsp––o

PT_DELETE (MOClass = LogRecord, MOInstance = null)
```

A.5 "Preconfigure" Network/Transport Layer Connections

In order to "preconfigure" network/transport layer connections, the following sequence could occur over the interoperable interface:

A.5.1 Identify type of connection

For example, the applicable objects from ISP TB1111/21 could be:

(a) transportConnection
(b) x25PLE–DTE
(c) virtualCall–DTE
(d) PVC–DTE (no IVMO)

A.5.2 Establishing a connection

A manager desiring to establish a connection sends a create request.

o––Req–––––––––––––––––––––Ind––>

M_CREATE(MOClass = { IVMO })* AttrList includes desired attribute ids and values representing configurable parameters

Agent receives PT–CREATE, sets up IVMO MO instance with desired values, returns PT–CREATE Response with success or failure. Manager receives PT–CREATE Confirm

<––Cnf–––––––––––––––––––––Rsp––o

M_CREATE(MOClass = { IVMO })* AttrList includes desired attribute ids and values representing configurable parameters

* The PVC–DTE initial attribute values are either supplied via the PT–CREATE or are the default values.

A.5.3 Configuring the new connection

The Agent instantiates a new connection MO with initial attribute values supplied by corresponding IVMO MO.

"Preconfigured" values can also be modified or deleted altogether. These steps are similar to those defined in A.3, Reconfigure, where the target MO is an IVMO.