INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.723

(11/93)

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI MANAGEMENT

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – STRUCTURE OF MANAGEMENT INFORMATION: GENERIC MANAGEMENT INFORMATION

ITU-T Recommendation X.723

(Previously "CCITT Recommendation")

Foreword

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology, which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation X.723 was approved on 16th November 1993. The identical text is also published as ISO/IEC International Standard 10165-5.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

© ITU 1994

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

Page

1 Scope 1
2 Normative references 1
3 Definitions 2
4 Abbreviations 3
5 Conventions 3
6 Overview 3
7 Generic managed object class definitions 4
8 Name bindings 9
9 Attributes 10
10 Attribute Groups 15
11 Actions 16
12 Notifications 17
13 ASN.1 definitions 18

Summary

This Recommendation documents management information that is applicable to more than one open system resource by documenting those common structures as generic definitions. These generic definitions may be used in the specification of managed objects for technology specific use such as OSI layers. In general, use of OSI layer specific use will require.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – STRUCTURE OF MANAGEMENT INFORMATION: GENERIC MANAGEMENT INFORMATION

1 Scope

This Recommendation | International Standard provides developers of OSI Recommendations | International Standards that contain managed object definitions with generic definitions of managed object classes that will

- provide common superclass definitions from which layer- or resource- specific object class definitions may be derived;
- assist with the development of common elements of object class definitions across multiple layers or components of layers;
- reduce duplication of effort in other working groups by identifying commonly useful definitions.

To this end, this Recommendation | International Standard defines managed object classes which may be used as superclasses, using the template notation defined in CCITT Rec. X.722 | ISO/IEC 10165-4, including the definition of any packages, attributes, attribute groups, behaviour, actions, notifications and parameters that may be associated with the superclasses. It also defines functional units required by layer- or resource-specific management standards which are not defined in the CCITT Rec. X.73X | ISO/IEC 10164 series of systems management function standards.

This Recommendation | International Standard is applicable to the development of any OSI Recommendation | International Standard which defines managed object classes that may be refined from the managed object classes defined in this Recommendation | International Standard.

This Recommendation | International Standard does not specify or imply any constraints on the development of managed object definitions in terms of their inheritance relationships. Developers of managed object class definitions are encouraged to use the definitions contained in this Recommendation | International Standard to promote consistency between similar managed object classes; however, use of these definitions is not mandated.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The ITU-T maintains a list of the currently valid Recommendations.

2.1 Identical Recommendations | International Standards

- CCITT Recommendation X.701 (1992) | ISO/IEC 10040:1992, *Information technology – Open Systems Interconnection – Systems Management Overview*.
- CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1:1992, *Information technology – Open Systems Interconnection – Structure of Management Information: Management Information Model*.
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of Management Information: Definition of Management Information*.

- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of Management Information: Guidelines for the Definition of Managed Objects*.
- CCITT Recommendation X.731 (1992) | ISO/IEC 10164-2:1992, *Information technology – Open Systems Interconnection – Systems Management: State Management Function*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.200 (1989), *Reference Model of Open Systems Interconnection for CCITT Applications*.

  ISO 7498:1984, *Information processing systems – Open Systems Interconnection – Basic Reference Model*.
- CCITT Recommendation X.208 (1989), *Specification of abstract syntax notation one (ASN.1)*.

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.
- CCITT Recommendation X.700 (1992), *Management Framework Definition for Open Systems Interconnection (OSI) for CCITT Applications*.

  ISO 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management Framework*.
- CCITT Recommendation X.710 (1991), *Common Management Information Service Definition for CCITT Applications*.

  ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition*.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 Basic reference model definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.200 | ISO 7498:

a) (N)-SAP;
b) open system;
c) systems management;
d) application process.

3.2 Management framework definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.700 | ISO/IEC 7498-4:

managed object.

3.3 Systems management overview definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.701 | ISO/IEC 10040:

a) generic definitions;
b) managed object class;
c) management information;
d) notification.

3.4 Management information model definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.720 | ISO/IEC 10165-1:

a) allomorphism;
b) attribute type;
c) behaviour;
d) containment;

e) name binding;
f) package;
g) specialization;
h) subordinate object;
i) superclass;
j) superior object.

3.5 CMIS definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.710 | ISO/IEC 9595:

attribute.

3.6 ASN.1 definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.208 | ISO/IEC 8824:

a) object identifier;
b) sequence-of type;
c) sequence type;
d) set-of type;
e) set type;
f) type.

3.7 GDMO definitions

This Recommendation | International Standard uses the following terms defined in CCITT Rec. X.722 | ISO/IEC 10165-4:

a) managed object class definition;
b) template.

4 Abbreviations

ASN.1	Abstract Syntax Notation One
CL	Connectionless
CMIS	Common Management Information Service
CO	Connection Oriented
GDMO	Guidelines for the Definition of Managed Objects
GMI	Generic Management Information
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
SAP	Service Access Point
SMI	Structure of Management Information

5 Conventions

This Recommendation | International Standard uses the template notation defined in CCITT Rec. X.722 | ISO/IEC 10165-4 as the notation for defining managed object classes and the abstract syntax notation defined in CCITT Rec. X.208 | ISO/IEC 8824 as the means of defining ASN.1 data types associated with the use of the template notation.

Text based on these notational conventions is printed in a distinctive typeface.

6 Overview

The purpose of this Recommendation | International Standard is to document structures for managed object classes and management information types that are common to more than one open system resource, by defining those common structures as generic definitions of management information. Other work items, such as those concerned with the

management of individual OSI layers, may specialize these generic definitions in ways that are appropriate for the requirements of the layer or resource concerned. In the process of specialization, the generic managed object classes defined in this Recommendation | International Standard may be defined as superclasses of allomorphic classes where appropriate. The use of generic managed object classes in this way will help to ensure the consistency and completeness of managed object class definitions for such resources as layer entities, SAPs, connections, etc., and also take account of relationships between them. Similar advantages may be gained by defining managed object classes corresponding to components of OSI layers, such as subnetworks.

Definers of management information are also encouraged to use the definitions contained in CCITT Rec. X.721 | ISO/IEC 10165-2 where they are applicable to the managed object classes or other information types being defined.

7 Generic managed object class definitions

This clause contains managed object class definitions for generic managed object classes, along with their mandatory and conditional packages and behaviour definitions. The definitions have been documented "in-line" with the exception of the attribute definitions which are to be found in clause 9.

7.1 Application process

The applicationProcess managed object class is used to represent an element within a system which performs the information processing for a particular application. The supportEntityNames attribute contains a set of distinguished names of supporting entities which provide communication services to the application process.

The semantics and syntax of the operationalState attribute are defined in CCITT Rec. X.731 | ISO/IEC 10164-2 and CCITT Rec. X.721 | ISO/IEC 10165-2.

```
applicationProcess MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        applicationProcessP1          PACKAGE

            BEHAVIOUR
               applicationProcessB1   BEHAVIOUR
                   DEFINED AS
                       !An application process supports the disabled and enabled values of the operationalState attribute as
                       described in CCITT Rec. X.731 (1992) | ISO/IEC 10164-2:1992 as follows:
                       - an application process is disabled if it is inoperable or a resource on which it depends is inoperable,
                       - an application process is enabled if it is operable.!
               ;
            ;
            ATTRIBUTES      applicationProcessId      GET,
                            applicationProcessTitle   GET,
                            supportEntityNames GET,
                            "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET
            ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) applicationProcess(8)} ;
```

7.2 Communications entity

The communications entity managed object class is used to represent an active functional element which performs communication processing functions within a system, e.g. Network Entity, Transport Entity, etc. The localSapNames attribute contains a set of distinguished names of layer (N-1) SAPs or ports at which services are provided to the entity.

NOTE 1 – Although the localSapNames attribute is for generality defined as a set valued attribute and may therefore contain zero or more SAP names, the set membership is, in most cases, limited to a single SAP name.

The semantics and syntax of the operationalState attribute are defined in CCITT Rec. X.731 (1992) | ISO/IEC 10164-2:1992 and CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992.

NOTE 2 – In circumstances where both connection oriented and connectionless services are available at a particular layer, the further refinement of the communications entity managed object class to distinguish between connection oriented and connectionless entities is discouraged, as it becomes unclear under some circumstances (particularly under error conditions) as to what belongs to the CO entity and what belongs to the CL entity (e.g. If a corrupted PDU is received, is it the CO counter of corrupted PDUs that is incremented, or is it the CL counter, or both?).

```
communicationsEntity        MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        communicationsEntityP1 PACKAGE

            BEHAVIOUR
                communicationsEntityB1 BEHAVIOUR
                    DEFINED AS
                        !A communications entity supports the disabled and enabled values of the operationalState attribute
                        as described in CCITT Rec. X.731 (1992) | ISO/IEC 10164-2:1992 as follows:
                        -  an entity is disabled if it is inoperable or a resource on which it depends is inoperable,
                        -  an entity is enabled if it is operable.!
                ;
            ;
            ATTRIBUTES      communicationsEntityId GET,
                            localSapNames GET,
                            "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState GET
            ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) communicationsEntity(0)} ;
```

7.3 Communications information record

The communicationsInformationRecord managed object class is used to define the information stored in a log as a result of receiving event reports or potential event reports with a communication information event type. The semantics of the managed object class, namely its attributes and behaviour, are derived from the communicationsInformation notification.

```
communicationsInformationRecord MANAGED OBJECT CLASS

    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":eventLogRecord;

    CHARACTERIZED BY
        communicationsInformationRecordP1 PACKAGE
            BEHAVIOUR
                communicationsInformationRecordB1 BEHAVIOUR
                    DEFINED AS
                        !Log record class for communicationsInformation events.!
                ;
            ;
            ATTRIBUTES      informationType GET;
    ;
    ;
    CONDITIONAL PACKAGES
        informationDataPackage PACKAGE
            ATTRIBUTES
                informationData GET
            ;
        REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) package(4) informationDataPackage(0)};
        PRESENT IF  !The informationData parameter is present in the communicationsInformation event report
                    corresponding to the instance of communicationsInformationRecord.!
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) informationData(1)};
```

7.4 Connectionless-mode protocol machine

The connectionless-mode protocol machine managed object class is used to represent a protocol machine performing connectionless-mode communications functions within an entity.

```
clProtocolMachine    MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

    clProtocolMachineP1   PACKAGE

        BEHAVIOUR
          clProtocolMachineB1   BEHAVIOUR
               DEFINED AS
                  !A protocol machine which performs connectionless-mode communications functions.!
          ;
        ;
        ATTRIBUTES      clProtocolMachineId       GET,
                        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState      GET;
        ;
    ;
    CONDITIONAL PACKAGES
        clProtocolMachineP2 PACKAGE
          ATTRIBUTES totalRemoteSAPs    GET ;
        REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) package(4) clProtocolMachineP2(1)} ;
        PRESENT IF      !there is a requirement to keep statistics concerning remote connectionless protocol machines
                        that this protocol machine communicates with!
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) clProtocolMachine(2)} ;
```

7.5 Connection-mode protocol machine

The connection-mode protocol machine managed object class is used to represent a protocol machine performing connection-mode communications functions within an entity.

```
coProtocolMachine   MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        coProtocolMachineP1     PACKAGE

          BEHAVIOUR
              coProtocolMachineB1    BEHAVIOUR
                   DEFINED AS
                      !A protocol machine which performs connection-mode communications functions.!
              ;
          ;
          ATTRIBUTES      coProtocolMachineId       GET,
                          "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":operationalState      GET
          ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) coProtocolMachine(3)} ;
```

7.6 Sap 1

The sap1 managed object class is used to represent a service access point at which services are provided by an entity to the user entity, where the SAP address is dependent on the SAP address of the underlying layer.

The sap1Address attribute contains the selector which is contributed to the address of the SAP by this layer.

The userEntityNames attribute contains the distinguished names of the managed objects that represent the user entities that are using the SAP.

```
sap1        MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        sap1P1 PACKAGE

            BEHAVIOUR
               sap1B1 BEHAVIOUR
                   DEFINED AS
                       !This managed object represents the point at which an entity provides services to the user entity. Refer
                       to the Basic Reference Model for the definition of (N)-SAP.!
               ;
            ;
            ATTRIBUTES      sapId GET,
                            sap1Address GET,
                            userEntityNames GET
            ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) sap1(4)} ;
```

7.7 Sap 2

The sap2 managed object class is used to represent a service access point, the address of which is independent of the SAP address of the underlying layer.

The sap2Address attribute contains the address of the SAP.

The userEntityNames attribute contains the distinguished names of the managed objects that represent the user entities that are using the SAP.

The providerEntityNames attribute contains the distinguished names of the managed objects that represent the provider entities that are supporting the SAP.

```
sap2        MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        sap2P1   PACKAGE

            BEHAVIOUR
               sap2B1   BEHAVIOUR
                   DEFINED AS
                       !A service access point, the address of which is independent of the SAP address of the underlying
                       layer.!
               ;
            ;
            ATTRIBUTES      sapId GET,
                            sap2Address GET,
                            userEntityNames GET,
                            providerEntityNames GET
            ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) sap2(5)} ;
```

7.8 Single peer connection

The single peer connection managed object class is used to represent an association or a connection established between entities for the transfer of service or protocol data.

The underlyingConnectionNames attribute contains the distinguished names of the managed objects that represent the underlying connections or physical media used by the connection.

```
singlePeerConnection  MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        singlePeerConnectionP1 PACKAGE

            BEHAVIOUR
               singlePeerConnectionB1 BEHAVIOUR
                    DEFINED AS
                        !This managed object class represents a view of a single peer connection between a pair of entities.!
               ;
            ;
            ATTRIBUTES      connectionId GET,
                            underlyingConnectionNames GET
            ;
        ;
    ;
    CONDITIONAL PACKAGES
        singlePeerConnectionP2 PACKAGE
            ATTRIBUTES      supportedConnectionNames GET
            ;
        REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) package(4) spcP2(2)} ;
        PRESENT IF !The names of the connections supported by this connection can be provided!
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) singlePeerConnection(6)} ;
```

7.9 Subsystem

The subsystem managed object class may be used as a common containment point for managed objects in a system that relate to the operation of a given layer. The choice of structuring within a system is dependent on what structure the system designer wishes to present externally for management purposes. The subsystemId attribute is used to identify the subsystem concerned.

```
subsystem           MANAGED OBJECT CLASS

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

    CHARACTERIZED BY

        subsystemP1 PACKAGE

            BEHAVIOUR
               subsystemB1 BEHAVIOUR
                    DEFINED AS
                        !This managed object class represents a portion of a system where components are named
                        independently of the components of other subsystems.!
               ;
            ;
            ATTRIBUTES      subsystemId GET
            ;
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) managedObjectClass(3) subsystem(7)} ;
```

8 Name bindings

This clause provides definitions of name bindings that may be applicable to the generic managed object classes defined in this Recommendation | International Standard. Other name bindings may be defined and used with these managed object classes.

8.1 ApplicationProcess-System

applicationProcess-system NAME BINDING

SUBORDINATE OBJECT CLASS applicationProcess AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES ;
WITH ATTRIBUTE applicationProcessId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) ap-s(7)} ;

8.2 CLProtocolMachine-CommunicationsEntity

clProtocolMachine-entity NAME BINDING

SUBORDINATE OBJECT CLASS clProtocolMachine AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS communicationsEntity AND SUBCLASSES ;
WITH ATTRIBUTE clProtocolMachineId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) clpm-e(0)} ;

8.3 CommunicationsEntity-Subsystem

communicationsEntity-subsystem NAME BINDING

SUBORDINATE OBJECT CLASS communicationsEntity AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS subsystem AND SUBCLASSES ;
WITH ATTRIBUTE communicationsEntityId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) ce-s(1)} ;

8.4 COProtocolMachine-CommunicationsEntity

coProtocolMachine-entity NAME BINDING

SUBORDINATE OBJECT CLASS coProtocolMachine AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS communicationsEntity AND SUBCLASSES ;
WITH ATTRIBUTE coProtocolMachineId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) copm-e(2)} ;

8.5 Sap1-CommunicationsEntity

sap1-communicationsEntity NAME BINDING

SUBORDINATE OBJECT CLASS sap1 AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS communicationsEntity AND SUBCLASSES ;
WITH ATTRIBUTE sapId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) s1-ce(3)} ;

8.6 Sap2-Subsystem

sap2-subsystem NAME BINDING

SUBORDINATE OBJECT CLASS	sap2 AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS	subsystem AND SUBCLASSES ;
WITH ATTRIBUTE	sapId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) s2-s(4)} ;

8.7 SinglePeerConnection-COProtocolMachine

singlePeerConnection-coProtocolMachine NAME BINDING

SUBORDINATE OBJECT CLASS	singlePeerConnection AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS	coProtocolMachine AND SUBCLASSES ;
WITH ATTRIBUTE	connectionId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) spc-copm(5)} ;

8.8 Subsystem-System

subsystem-system NAME BINDING

SUBORDINATE OBJECT CLASS	subsystem AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS	"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES ;
WITH ATTRIBUTE	subsystemId ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) nameBinding(6) su-s(6)} ;

9 Attributes

This clause defines attribute types that are referenced by the managed object class definitions contained in this Recommendation | International Standard.

9.1 ApplicationProcessId

The applicationProcessId attribute type is used in naming instances of the applicationProcess managed object class.

applicationProcessId ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.ApplicationProcessId ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) applicationProcessId(16)} ;

9.2 ApplicationProcessTitle

The applicationProcessTitle attribute type is used to specify the object identifier name form or directory name form of an application process. If not known, the value of this attribute is set to null.

applicationProcessTitle ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.ApplicationProcessTitle ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) applicationProcessTitle(17)} ;

9.3 CommunicationsEntityId

The communicationsEntityId attribute type is used in naming instances of the communicationsEntity managed object class.

communicationsEntityId ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.CommunicationsEntityId ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) communicationsEntityId(0)} ;

9.4 ConnectionId

The connectionId attribute type is used in naming instances of the connection managed object class.

connectionId ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.ConnectionId ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) connectionId(1)} ;

9.5 Connectionless-mode protocol machine Id

The clProtocolMachineId attribute type is used in naming instances of the connectionless-mode protocol machine managed object class.

clProtocolMachineId ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.ProtocolMachineId ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) clProtocolMachineId(2)} ;

9.6 Connection-mode protocol machine Id

The coProtocolMachineId attribute type is used in naming instances of the connection-mode protocol machine managed object class.

coProtocolMachineId ATTRIBUTE

WITH ATTRIBUTE SYNTAX	GMI-AttributeModule.ProtocolMachineId ;
MATCHES FOR	EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) coProtocolMachineId(3)} ;

9.7 GenericNonWrappingCounter

This is a non-instantiable generic counter definition from which non-wrapping counters with specific characteristics may be derived.

```
genericNonWrappingCounter ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GenericCounter;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        genericNonWrappingCounterB BEHAVIOUR
            DEFINED AS
                !The counter shall have a mandatory initial value of zero and places an upper bound on the attribute value. The
                value of the counter shall be incremented by an amount as specified in other behaviour definitions associated
                with this attribute definition. When the counter value reaches the upper bound, its value is held at the value of
                the upper bound. This does not require the system to maintain a counter large enough to accommodate this
                maximum value if the characteristics of the implementation are such that all achievable count values are less
                than this maximum value.!;
    ;
;   -- NOTE – Since this attribute is never instantiated, there is no REGISTERED AS construct.
```

9.8 GenericWrappingCounter

This is a non-instantiable generic counter definition from which wrapping counters with specific characteristics may be derived.

```
genericWrappingCounter ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.GenericCounter;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        genericWrappingCounterB BEHAVIOUR
            DEFINED AS
                !The counter shall have a mandatory initial value of zero and places an upper bound on the attribute value.
                The value of the counter shall be incremented by an amount as specified in other behaviour definitions
                associated with this attribute definition. When the counter value reaches the upper bound, a further increment
                causes the counter value to wrap. This does not require the system to maintain a counter large enough to
                accommodate this maximum value if the characteristics of the implementation are such that all achievable
                count values are less than this maximum value.!
        ;
    ;
 ;  -- NOTE – Since this attribute is never instantiated, there is no REGISTERED AS construct.
```

9.9 InformationData

This attribute represents the structure of the information to be held in a communicationsInformationRecord managed object.

```
informationData ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.InformationData;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) informationData(4)};
```

9.10 InformationType

This attribute represents the type of information to be held in a communicationsInformationRecord managed object.

```
informationType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.InformationType;
    MATCHES FOR EQUALITY;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) informationType(5)};
```

9.11 LocalSapNames

The semantics of the localSapNames attribute type are defined in 7.2.

```
localSapNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
     MATCHES FOR        EQUALITY, SET-COMPARISON, SET-INTERSECTION ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) localSapNames(6)} ;
```

9.12 NonWrapping64BitCounter

This is a non-wrapping counter which has an upper bound of 2^64-1.

```
nonWrapping64BitCounter ATTRIBUTE

    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.Generic64BitCounter ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR genericNonWrappingCounterB ;
;   -- As this attribute is never instantiated, there is no REGISTERED AS construct. Instantiable attribute
    -- definitions derived from this definition shall specify the circumstances under which the attribute value is
    -- incremented.
```

9.13 ProviderEntityNames

The semantics of the ProviderEntityNames attribute type are defined in 7.7.

```
providerEntityNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
     MATCHES FOR       EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
     BEHAVIOUR
        providerEntityNamesB          BEHAVIOUR
            DEFINED AS        !The distinguished names of the (N-1) layer entities which provide services to this SAP.!
        ;
    ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) providerEntityNames(7)} ;
```

9.14 Sap1Address

The semantics of the sap1Address attribute type are defined in 7.6.

```
sap1Address ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.Sap1Address ;
    MATCHES FOR             EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) sap1Address(8)} ;
```

9.15 Sap2Address

The semantics of the sap2Address attribute type are defined in 7.7.

```
sap2Address ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.Sap2Address ;
    MATCHES FOR     EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
    BEHAVIOUR
        sap2AddressB          BEHAVIOUR
            DEFINED AS        !The set of addresses associated with an (N) SAP!
        ;
    ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) sap2Address(9)} ;
```

9.16 SapId

The sapId attribute type is used in naming instances of the sap1 and sap2 managed object classes.

```
sapId ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.SapId ;
    MATCHES FOR     EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) sapId(10)} ;
```

9.17 SubsystemId

The subsystemId attribute type is used in naming instances of the subsystem managed object class.

```
subsystemId ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.SubsystemId ;
    MATCHES FOR     EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) subsystemId(11)} ;
```

9.18 SupportedConnectionNames

```
supportedConnectionNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
    MATCHES FOR     EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
    BEHAVIOUR
        supportedConnectionNamesB BEHAVIOUR
            DEFINED AS
                !This attribute contains the distinguished names of managed objects that represent connections that are
                supported by the connection that has this attribute.!
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) supportedConnectionNames(12)} ;
```

9.19 SupportEntityNames

The semantics of the supportEntityNames attribute type are defined in .

```
supportEntityNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
     MATCHES FOR        EQUALITY, SET-COMPARISON, SET-INTERSECTION ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) supportEntityNames(18)} ;
```

9.20 Timer

The timer is a generic definition of an attribute designed to hold an interval timer value.

```
timer ATTRIBUTE

    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.Timer;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        timerB BEHAVIOUR
            DEFINED AS
                !A timer whose value may be set and read to the precision implied by the syntax definition, but whose effect on
                the precision with which the protocol events controlled by this timer are generated is determined by the
                implementation. The details of this precision shall be stated in the MOCS. The value of the timer consists of
                two integer values, exponent and mantissa, and shall be interpreted as representing the value
                mantissa*10^exponent.!
        ;
    ;
;   -- Since this attribute is never instantiated, there is no REGISTERED AS construct. Timers derived from this definition
    -- shall specify the units of time that the timer value represents.
```

9.21 TotalRemoteSAPs

This attribute provides a count of the number of remote SAPs that a clProtocolMachine has communicated with during its lifetime.

```
totalRemoteSAPs    ATTRIBUTE
    DERIVED FROM wrapping64BitCounter ;
    BEHAVIOUR
        totalRemoteSAPsB BEHAVIOUR
            DEFINED AS
                !Counts the number of remote (N) SAPs that the containing clProtocolMachine has communicated with.
                Incremented by 1 every time communication with a remote (N) SAP is achieved.!
        ;
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) totalRemoteSAPs(13)} ;
```

9.22 UnderlyingConnectionNames

The semantics of the underlyingConnectionNames attribute type are defined in 7.8.

```
underlyingConnectionNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
     MATCHES FOR       EQUALITY, SET-COMPARISON, SET-INTERSECTION ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) underlyingConnectionNames(14)} ;
```

9.23 UserEntityNames

The semantics of the userEntityNames attribute type are defined in 7.6 and 7.7.

```
userEntityNames ATTRIBUTE

    WITH ATTRIBUTE SYNTAX       GMI-AttributeModule.GroupObjects ;
     MATCHES FOR       EQUALITY, SET-COMPARISON, SET-INTERSECTION ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attribute(7) userEntityNames(15)} ;
```

9.24 Wrapping64BitCounter

This is a wrapping counter which has an upper bound of 2^64-1.

```
wrapping64BitCounter ATTRIBUTE

    WITH ATTRIBUTE SYNTAX GMI-AttributeModule.Generic64BitCounter ;
     MATCHES FOR EQUALITY ;
     BEHAVIOUR genericWrappingCounterB ;
;   -- As this attribute is never instantiated, there is no REGISTERED AS construct. Instantiable attribute definitions
    -- derived from this definition shall specify the circumstances under which the attribute value is incremented.
```

10 Attribute Groups

This clause provides definitions of attribute groups.

10.1 Counters

This is an empty, extensible attribute group consisting of all counters that are part of a managed object class definition. Attributes are added to the group in the package definition(s) for the managed object class.

```
counters ATTRIBUTE GROUP
     DESCRIPTION
     !The group of all counter attributes.!
     ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) attributeGroup(8) counters(0)};
```

11 Actions

11.1 Activate

The activate action provides a means of initializing a managed object.

```
activate ACTION
    BEHAVIOUR
        activateB BEHAVIOUR
            DEFINED AS
                !The activate action initializes the resource represented by the managed object so that it can perform its
                normal functions. Managed objects that support this action cannot always be created in the enabled state.

                Although the generic behaviour of the activate action is described with the CCITT Rec. X.721 (1992) |
                ISO/IEC 10164-2:1992 (State management function) attributes Operational state and Procedural status, this
                does not imply that either attribute need be made visible to management.

                This initialization procedure is needed to enable the managed object if the Procedural status attribute contains
                the value Initialization Required and the Operational state is Disabled.

                The activate action initiates a procedure that causes the resource to enter its normal functioning mode.

                If the resource is or can be enabled, the activate action completes successfully; if the resource cannot be
                enabled, the activate action returns a processingFailure error.

                While the initialization procedure is in progress, the Procedural status attribute contains the value Initializing.
                The Initialization Required condition is absent.

                If the initialization procedure completes successfully and the resource is operable and available for use, the
                Initializing condition is removed from the Procedural status attribute and the Operational state is Enabled. If
                the initialization procedure does not complete successfully, the Initializing condition is removed from the
                Procedural status attribute and the Initialization Required condition added. The Operational state is Disabled.!
        ;
    ;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     GMI-AttributeModule.ActionInfo ;
    WITH REPLY SYNTAX           GMI-AttributeModule.ActionReply;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) action(9) activate(0)};
```

11.2 Deactivate

The deactivate action provides a means of abruptly terminating the operation of a managed object.

```
deactivate ACTION
    BEHAVIOUR
        deactivateB BEHAVIOUR
            DEFINED AS
                !The deactivate action permits the normal operation of the resource to be terminated by the manager.

                Although the generic behaviour of the deactivate action is described with the CCITT Rec. X.721
                (1992) | ISO/IEC 10164-2:1992 (State management function) attributes Operational state and Procedural
                status, this does not imply that either attribute be visible to management.

                The deactivate action initiates a procedure that causes the resource to cease its normal functioning mode.

                If the resource is or can be disabled, the deactivate action completes successfully; if the resource cannot be
                disabled, the deactivate action returns a processingFailure error.
```

```
            While the termination procedure is in progress, the Procedural status attribute contains the value Terminating
            and the Operational state is Disabled. The Terminating condition is removed when the procedure completes.!
        ;
    ;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     GMI-AttributeModule.ActionInfo ;
    WITH REPLY SYNTAX           GMI-AttributeModule.ActionReply;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) action(9) deactivate(1)};
```

11.3 DeactivateWhenNoUsers

The deactivateWhenNoUsers action provides a means of gracefully terminating the operation of a managed object.

```
deactivateWhenNoUsers ACTION
    BEHAVIOUR
        deactivateWhenNoUsersB BEHAVIOUR
            DEFINED AS
                !The deactivateWhenNoUsers action permits the normal operation of the resource represented by the
                managed object to be terminated by the manager when the number of instances of use of the resource is
                zero.

                Although the generic behaviour of the deactivate action is described with the CCITT Rec. X.721
                (1992) | ISO/IEC 10164-2:1992 (State management function) attributes Operational state and Procedural
                status, this does not imply that either attribute be visible to management.

                The deactivateWhenNoUsers action initiates a procedure that causes the resource to cease its normal
                functioning mode when the number of instances of use is zero.

                If at the time the resource has no existing users, or when, subsequently, the number of users drops to zero,
                the termination procedure is initiated and the Procedural status attribute is set to Terminating. If the resource
                is or can be disabled, the deactivateWhenNoUsers action completes successfully; if the resource cannot be
                disabled, the deactivateWhenNoUsers action returns a processingFailure error.

                While the termination procedure is in progress, the Procedural status attribute contains the value Terminating.
                When the number of users drops to zero, the Operational state is Disabled. The Terminating condition is
                removed when the procedure completes.!
        ;
    ;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     GMI-AttributeModule.ActionInfo ;
    WITH REPLY SYNTAX           GMI-AttributeModule.ActionReply;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) action(9) deactivateWhenNoUsers(2)};
```

12 Notifications

12.1 CommunicationsInformation

```
communicationsInformation NOTIFICATION
    BEHAVIOUR
        communicationsInformationB BEHAVIOUR
            DEFINED AS
                !This notification may be used to report the occurrence of events pertaining to the normal operation of a
                managed object. These are informational events; important enough to report, but not requiring any further
                action by a manager. Faults and abnormal conditions (which may require manager action) shall be reported
                using the event types defined in CCITT Rec. X.733 (1992) | ISO/IEC 10164-4:1992.!
        ;
    ;
    WITH INFORMATION SYNTAX             GMI-AttributeModule.CommunicationsInformation
                    AND ATTRIBUTE IDS
                    informationType informationType,
                    informationData informationData
    ;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part5(5) notification(10) comsinfo(0)};
```

The following object identifier

{joint-iso-ccitt ms(9) smi(3) part5(5) functionalUnitPackage(1) informationEventReports(0)}

is assigned as a value of the ASN.1 type FunctionalUnitPackageId, defined in CCITT Rec. X.701 (1992) | ISO/IEC 10040:1992 to use for negotiating the following functional unit

0 communication information functional unit

where the number identifies the bit position assigned to the functional unit.

13 ASN.1 definitions

This clause defines the ASN.1 data types required by the attribute definitions in clause 9.

```
GMI-AttributeModule {joint-iso-ccitt ms(9) smi(3) part5(5) asn1Module(2) gmiAttributeModule(0)}
DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS

    GroupObjects, ManagementExtension
        FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

    RelativeDistinguishedName
        FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)}

;-- End of IMPORTS

ActionInfo ::= SET OF ManagementExtension
ActionReply ::= SET OF ManagementExtension
ApplicationProcessId ::= GraphicString
ApplicationProcessTitle ::= CHOICE {
                                    directory       [0] SEQUENCE OF RelativeDistinguishedName,
                                    oid             [1] OBJECT IDENTIFIER,
                                    notKnown            NULL }
CommunicationsEntityId ::= GraphicString
CommunicationsInformation ::= SEQUENCE {
    informationType InformationType,
    informationData InformationData OPTIONAL }
ConnectionId ::= GraphicString
GenericCounter ::= INTEGER
Generic64BitCounter ::= INTEGER (0..18446744073709551615)
InformationType ::= OBJECT IDENTIFIER
InformationData ::= SET OF ManagementExtension
ProtocolMachineId ::= GraphicString
Sap1Address ::= INTEGER
Sap2Address ::= SET OF OCTET STRING
SapId ::= GraphicString
SubsystemId ::= GraphicString
Timer ::= SEQUENCE {
        exponent [1] INTEGER (-62..63),
        mantissa [2] INTEGER (0..65535) }

END
```