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

Output from the June 1991 NIST Workshop for Implementors of OSI

SIG Co-Chair **Paul Brusil, The Mitre Corporation**
SIG Co-Chair **George Mouradian, AT&T Bell Labs**
SIG Editor **Robert Aronoff, NIST**

Foreword

This part of the Working Implementation Agreements was prepared by the Network Management Special Interest Group (NMSIG) of the National Institute of Standards and Technology (NIST) Workshop for Implementors of Open Systems Interconnection (OSI). 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 redlining; deleted text is left in but marked with strikeouts.

Table of Contents

List of Tables

List of Figures

18 Network Management

0 Introduction

(Refer to the Stable Implementation Agreements Document.)

1 Scope

(Refer to the Stable Implementation Agreements Document.)

2 Normative References

The following documents are referenced in the statements of the agreements relating to OSI network management. The notation "*" indicates that tentative object identifiers contained in these DIS–level documents are superseded by the NMSIG Phase 1 object identifiers contained in ANNEX B.2 of these agreements.

[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.

[AMF] ISO/IEC CD 10164–10, Information Technology – Open Systems Interconnection – Systems Management – Part 10: Accounting Meter Function, ISO/IEC JTC1/SC21 N4958, 4 July 1990.

[AMWD] Information Processing Systems – Open Systems Interconnection – Accounting Management Working Document (Fourth Version), ISO/IEC JTC1/SC21, May 30, 1990.

[ARF]* ISO/IEC DIS 10164–4, Information Technology – Open Systems Interconnection – Systems Management – Part 4: Alarm Reporting Function, ISO/IEC JTC1/SC21 N4858, June 1990.

[ARR]* ISO/IEC DIS 10164–3, Information Technology – Open Systems Interconnection – Systems Management – Part 3: Attributes for Representing Relationships, ISO/IEC JTC1/SC21 N4857, June 1990.

[ASN1] ISO/IEC 8824, Information Technology – Open System Interconnection – Specification of Abstract Syntax Notation One (ASN.1), 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.

[CDTC] Information Processing Systems – Open Systems Interconnection – Systems Management – Part Z: Confidence and Diagnostic Test Classes (First Version) ISO/IEC JTC1/SC21 N4957, May 1990.

[CMIP] ISO/IEC 9596–1, Information Technology – Open Systems Interconnection – Common Management Information Protocol Specification – Part 1: Specification, 24 November l990.

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

[CMO] Information Processing Systems – Open Systems Interconnection – Working Draft of the Configuration Management Overview, ISO/IEC JTC1/SC21 N3311, 16 January 1989.

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

[DMI]* ISO/IEC DIS 10165–2, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 2: Definition of Management Information, ISO/IEC JTC1/SC21 N4867, June 1990.

[ERMF]* ISO/IEC DIS 10164–5, Information Technology – Open Systems Interconnection – Systems Management – Part 5: Event Report Management Function, ISO/IEC JTC1/SC21 N4860, June 1990.

[FMWD] Information Processing Systems – Open Systems Interconnection – Systems Management – Fault Management Working Document, ISO/IEC JTC1/SC21 N4077, December 1989.

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

[GDMO]* ISO/IEC DIS 10165–4, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 4: Guidelines for the Definition of Managed Objects, ISO/IEC JTC1/SC21 N4852, 15 June 1990.

[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.

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

[LCF] ISO/IEC DIS 10164–6, Information Technology – Open Systems Interconnection – Systems Management – Part 6: Log Control Function, ISO/IEC JTC1/SC21 N4862, June 1990.

[MIM] ISO/IEC DIS 10165–1, Information Technology – Open Systems Interconnection – Management Information Services – Structure of Management Information – Part 1: Management Information Model, ISO/IEC JTC1/SC21 N5252, June 1990.

[OAAC] ISO/IEC CD 10164–9, Information Technology – Open Systems Interconnection – Systems Management – Part 9: Objects and Attributes for Access Control, ISO/IEC JTC1/SC21 N4956, June 1990.

[OMF]* ISO/IEC DIS 10164–1, Information Technology – Open Systems Interconnection – Systems Management – Part 1: Object Management Function, ISO/IEC JTC1/SC21 N4855, June 1990.

[PMWD] Information Processing Systems – Open Systems Interconnection – Performance Management Working Document (Sixth Draft), ISO/IEC JTC1/SC21 N4981, July 4, 1990.

[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 DP 10164–7, Information Technology – Open Systems Interconnection – Systems Management – Part 7: Security Alarm Reporting Function, May 1990.

[SATF] ISO/IEC CD 10164–8, Information Technology – Open Systems Interconnection – Systems Management – Part 8: Security Audit Trail Function, ISO/IEC JTC1/SC21 N4955, June 1990.

[SD35] EWOS/EG/NM/91/xx, Information Technology – International Standardized Profiles AOMnn – OSI Management – Part 3: AOM12 – Enhanced Management Communications, February 1991.

[SF] ISO/IEC CD 10164–13, Information Technology – Open Systems Interconnection – Systems Management – Part 13: Summarization Function, ISO/IEC JTC1/SC21 N4972, 2 July 1990.

[SMO]* ISO/IEC DIS 10040, Information Technology – Open Systems Interconnection – Systems Management Overview, ISO/IEC JTC1/SC21 N4865R, 16 June 1990.

[SMWD] Information Processing Systems – Open Systems Interconnection – Systems Management – OSI Security Management Working Document – 7th Draft, ISO/IEC JTC1/SC21 N4091, 15 November 1989.

[STMF]* ISO/IEC DIS 10164–2, Information Technology – Open Systems Interconnection – Systems Management – Part 2: State Management Function, ISO/IEC JTC1/SC21 N4856, June 1990.

[TMF] Information Processing Systems – Open Systems Interconnection – Systems Management – Part Y: Test Management Function, ISO/IEC JTC1/SC21 N4978, June 1990.

[WMF] ISO/IEC CD 10164–11, Information Technology – Open Systems Interconnection – Systems Management – Part 11: Workload Monitoring Function, ISO/IEC JTC1/SC21 N4959, June 28, 1990.

3 Status

The following clauses were moved into the Stable Agreements in June 1990:

0 INTRODUCTION

2 NORMATIVE REFERENCES (i.e., only those relevant to the Stable Agreements)

6 MANAGEMENT COMMUNICATIONS

- 6.2 General Agreements on Users of CMIS
- 6.3 Specific Agreements on Users of CMIS
- 6.4 Specific Agreements on CMIP

The following clauses were moved to the Stable Agreements in December 1990:

1 SCOPE

1.1 Phased Approach

1.1.1 Alignment With Evolving Standards

1.1.2 Definition of Phase 1

1.1.3 Future Phases

2 NORMATIVE REFERENCES (i.e., only those relevant to the newly added Stable Agreements)

5 MANAGEMENT FUNCTIONS AND SERVICES

5.1 General Agreements

5.2 Object Management Function Agreements

5.3 State Management Function Agreements

5.4 Attributes For Representing Relationships Agreements

5.5 Alarm Reporting Function Agreements

5.6 Event Report Management Function Agreements

6 MANAGEMENT COMMUNICATIONS

6.1 Association Policies

7 MANAGEMENT INFORMATION

7.1 The Information Model

7.2 Principles of Naming

7.3 Guidelines for the Definition of Management Information

The following clause was added to the Stable Agreements in March 1991:

6 MANAGEMENT COMMUNICATIONS

6.5 Services Required by CMIP (added as subclause 7 of part 13, Upper Layer Agreements)

The following three subclauses contain technical errata that are planned to be brought to a vote for the Stable Agreements in September 1991:

Editor's Note: **[**The text in the following subclauses is entered here in the Working Agreements to allow for review prior to moving the text to the Stable Agreements.**]**

6.2.5 CMIS Subsets

6.4.5 Parameters
6.4.6 Access Control Parameter

The following clauses are planned to be added to the Stable Agreements in December 1991:

8 CONFORMANCE

ANNEX B.2 Phase 1 Object Identifiers

4 Errata

(Refer to the Stable Implementation Agreements Document.)

5 Management Functions and Services

ISO has partitioned network management into five Specific Management Functional Areas (SMFAs) as a convenience for developing requirements particular to configuration management (CM), fault management (FM), performance management (PM), security management (SM), and accounting management (AM). These requirements are specified in five separate SMFA standards ([CMO], [FMWD], [SMWD], [AMWD], and [PMWD]). Since the SMFAs have overlapping requirements, management functions and management information applicable to one SMFA are often applicable to other SMFAs. Therefore, the SMFAs point to separate standards that contain the management functions needed to satisfy particular requirements.

This set of management functions is referred to as the System Management Functions (SMFs). They provide a generic platform of common network management capabilities available to any management application. For example, the event report management function [ERMF] may be used to report events to satisfy FM, PM, AM, and SM requirements. The log control function [LCF] may be used to satisfy both FM and SM requirements.

The following schematic (figure 1) depicts the functional hierarchy of SMFs and SMFAs. There are currently seven SMF draft international standards: Object Management [OMF], State Management [STMF], Attributes For Representing Relationships [ARR], Alarm Reporting [ARF], Event Report Management [ERMF], Log Control [LCF], and Security Alarm Reporting [SARF]. These SMFs provide much of the network management capabilities needed by CM and FM. When additional requirements are identified in other SMFAs, additional SMFs may be developed. Committee drafts are currently in progress for the following additional SMFs: Security Audit Trail [SATF], Accounting Metering [AMF], and Workload Monitoring [WMF]. Working drafts are currently in progress for the following additional SMFs: Confidence and Diagnostic Testing (consisting of two documents, one specifying a Test Management Function [TMF], and the other defining related management support objects classes and attributes [CDTC]), and Summarization [SF].

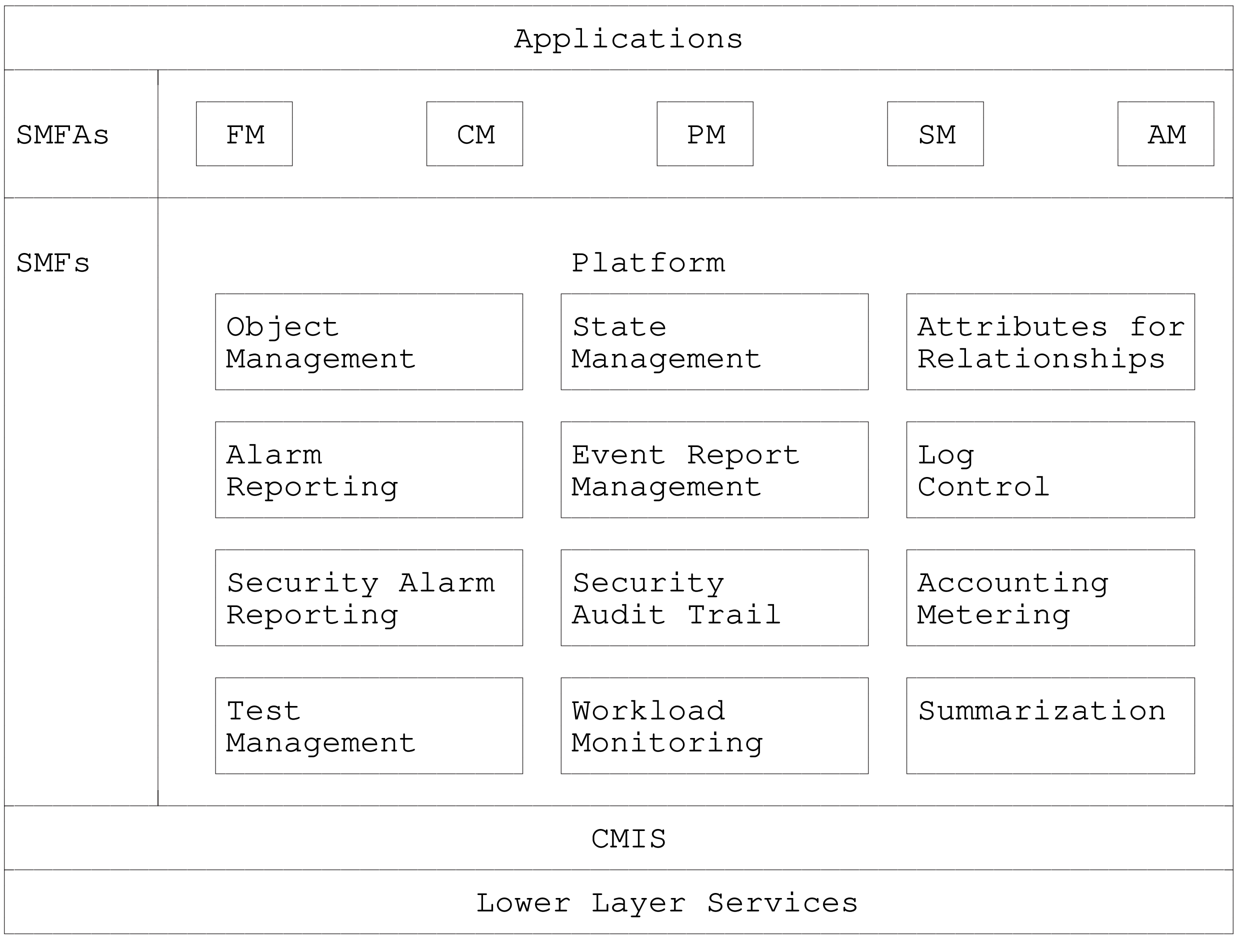

Figure 1: Functional Hierarchy of SMFs and SMFAs.

5.1 General Agreements

(Refer to the Stable Implementation Agreements Document.)

5.2 Object Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.3 State Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.4 Attributes For Representing Relationships Agreements

(Refer to the Stable Implementation Agreements Document.)

5.5 Alarm Reporting Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.6 Event Report Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.7 Log Control Function Agreements

5.7.1 Introduction

This subclause provides agreements pertinent to the Log Control Function defined by [LCF].

The Log Control Function provides SMF services by which event reports and other PDUs can be selected and stored. Log activity can be scheduled. Events and other PDUs are selected for logging by use of a "Discriminator Construct" attribute within a Log object. Log Control provides the services to initiate, terminate, suspend, or resume the logging activity through the manipulation of a Log object specified in [DMI]. In addition, Log Control can further alter the selection behavior by changing the distribution attributes in a Log object (e.g., Discriminator Construct).

According to the Log Control Model defined by [LCF], the Log object receives event reports, or other PDUs, from various sources, and adds information to their contents to form "potential log records". If the Log object is in a condition that allows it to be active, then it will evaluate the "potential log records" according to matching criteria in the Log objects Discriminator Construct attribute. The result of this sieve process will yield zero, one or more log records to be stored in the Log object for later retrieval.

The Log Control Function uses the State Management Function for the notification of state changes, and the Object Management Function for creating and deleting Log objects, retrieving Log attribute values, and notification of Log attribute value changes, Log record retrieval, and Log record deletion. It also uses the processing alarm notification of the Alarm Reporting Function [ARF].

The Log Control Function makes use of the following management support objects defined in [DMI]:

log, and
logRecord.

The Log Control Function makes use of the following attributes defined in [DMI], in addition to those attributes defined for the object class top:

logID,
discriminatorConstruct,
administrativeState,
operationalState,
usageState,
availabilityStatus,
maxLogSize,
currentLogSize,
numberOfRecords,
capacityAlarmThreshold,
logFullAction,
intervalsOfDay,
startTime,
stopTime,
weekMask, and
schedularName.

The Log Control Function makes use of the following notification types defined in [DMI]:

objectCreation,
objectDeletion,
stateChange,
attributeValueChange, and
processingErrorAlarm.

Editor's Note: [The [LCF] specifies "alarmNotification" which does not exist in [DMI]; the correct notification is "processingErrorAlarm". All other notifications are spelled incorrectly in [LCF]; the [DMI] spellings are used here. [LCF] does not specify "usageState" or "intervalsOfDay", but both are included here and in the [DMI] definition of the "Log" object class.**]**

5.7.2 General Agreements

These agreements address the following SMF services defined by the event report standard [LCF]:

Table 1: Scope of Agreements Relating to SMF Services Defined by the Log Control Standard [LCF]

Log Control SMF Service	Within Scope Of Agreements	Related Management Support Objects
Initiation of LCF	Yes	Log
Termination of LCF	Yes	Log
Log Modification, Suspension, Resumption	Yes	Log
Retrieving Logging Attributes	Yes	Log
Retrieval of Log Records	Yes	Log, Log Record
Deletion of Log Records	Yes	Log, Log Record

5.7.3 Initiation Of Event Report Logging

5.7.3.1 Introduction

This SMF service allows one open system to request that another open system create a Log object, thereby requesting that new or additional logs be defined.

The following informative table defines the mapping between LCF Initiation of Logging, OMF PT–Create, and CMIS M–CREATE service parameters. This tutorial information has been extracted from sections 9.2 and 11.2 of [LCF] and section 8.3.4 of [CMIS].

Table 2: Mapping Between LCF Initiation of Logging, OMF PT-Create, and CMIS M-CREATE Service Parameters

SMF Initiation of LCF Parameter	Req	Rsp	OMF PT-Create & CMIS M-CREATE Parameter	Req	Rsp
Invoke Identifier	M	M(=)			

SMF Initiation of LCF Parameter	Req	Rsp	OMF PT-Create & CMIS M-CREATE Parameter	Req	Rsp
Managed Object Class	M	C			
Managed Object Instance	U	C			
Support Object Instance	U	-			
Access Control	U	-			
Reference Object Instance	U	-			
Discriminator Construct	U	C	Attribute List		
Administrative State	U	C	Attribute List		
Operational State	-	C	Attribute List		
Usage State	-	C	Attribute List		
Availability Status	-	C	Attribute List		
Max Log Size	U	C	Attribute List		
Current Log Size	U	C	Attribute List		
Number Of Records	U	C	Attribute List		
Capacity Alarm Threshold	U	C	Attribute List		
Log Full Action	U	C	Attribute List		
Packages	U	C	Attribute List		
Week Mask	U	C	Attribute List		
Intervals Of Day	U	C	Attribute List		
Start Time	U	C	Attribute List		
Stop Time	U	C	Attribute List		
Schedular Name	U	C	Attribute List		
Current Time	-	U			
Errors	-	C			

5.7.3.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Initiation of Logging SMF service defined by section 9.2 of [LCF]. Relevant CMIS agreements defined in subclause 6.3.5 are repeated here for completeness.

Table 3: Agreements On Parameter Usage Pertinent to the Initiation of Logging SMF Service

SMF Initiation of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Managed Object Class	M	C		6.4.5
Managed Object Instance	U	C		6.2.1, 6.3.5.1
Support Object Instance	U	-		6.2.1
Access Control	U	-		6.2.4
Reference Object Instance	U	-		6.2.1
Discriminator Construct	U	C	[1], 5.1.2.1	6.4.5, 6.3.5.2
Administrative State	U	C		6.4.5, 6.3.5.2
Operational State	-	C		6.4.5, 6.3.5.2
Usage State	-	C		6.4.5, 6.3.5.2
Availability Status	-	C		6.4.5, 6.3.5.2
Max Log Size	U	C		6.4.5, 6.3.5.2
Current Log Size	U	C		6.4.5, 6.3.5.2
Number Of Records	U	C		6.4.5, 6.3.5.2
Capacity Alarm Threshold	U	C		6.4.5, 6.3.5.2
Log Full Action	U	C		6.4.5, 6.3.5.2
Packages	U	C		6.4.5, 6.3.5.2
Week Mask	U	C	[3]	6.4.5, 6.3.5.2
Intervals Of Day	U	C	[2]	6.4.5, 6.3.5.2
Start Time	U	C	[3]	6.4.5, 6.3.5.2
Stop Time	U	C	[3]	6.4.5, 6.3.5.2

SMF Initiation of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Schedular Name	U>I	C>I	[4]	6.4.5, 6.3.5.2
Current Time	-	U		6.2.3
Errors	-	C		6.4.4, 6.3.5.2

[1] As specified in [CMIP], the value "AND {}" shall be used to represent an all–pass Discriminator Construct. If this parameter is omitted from the request, the all–pass value shall be assigned to the Discriminator Construct attribute.

[2] The Daily Scheduling Package, if supported by an object, shall support at minimum the default 24 hour interval.

[3] The Weekly Scheduling Package, if supported by an object, shall support the default values for Start Time and Stop Time attributes. The Week Mask attribute shall support scheduling for each day of the week, and, at a minimum, the default 24 hour period for intervals of the day.

[4] Support for the External Schedular Package is beyond the scope of these agreements.

Editor's Note: **[**It is unclear whether "read-only" Log attributes such as LogId, objectClass, nameBindings, allomorphs, and name are permitted in the Attribute List parameter of the PT-CREATE request. This question has been submitted to ANSI X3T5.4. Depending upon the answer, it may be necessary to add an agreement on the initial values of these attributes. For now, the attribute list shown here has been made consistent with the attribute list shown for the corresponding [ERMF] service.**]**

5.7.4 Termination Of Logging

5.7.4.1 Introduction

This SMF service allows one open system to request that another open system delete one or more logs.

The following informative table defines the mapping between LCF Termination of Logging, OMF PT–Delete, and CMIS M–DELETE service parameters. This tutorial information has been extracted from sections 9.3 and 11.2 of [LCF] and section 8.3.5 of [CMIS].

Table 4: Mapping Between LCF Termination of Logging, OMF PT-Delete, and CMIS M-DELETE Service Parameters

SMF Termination of LCF Parameter	Req	Rsp	PT-Delete & CMIS M-DELETE Parameter	Req	Rsp
Invoke Identifier	M	M(=)			
Linked Id	-	C			
Base Object Class	M	-			
Base Object Instance	M	-			
Scope	U	-			
Filter	U	-			
Access Control	U	-			
Synchronization	U	-			
Managed Object Class	-	C			
Managed Object Instance	-	C			
Current Time	-	U			
Errors	-	C			

5.7.4.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Termination of Logging SMF service defined by section 9.3 of [LCF]. Relevant CMIS agreements defined in subclause 6.3.6 are repeated here for completeness.

Table 5: Agreements On Parameter Usage Pertinent to the Termination of Logging SMF Service

SMF Termination of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Linked Id	-	C		6.4
Base Object Class	M	-		6.4.5

SMF Termination of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Base Object Instance	M	-		6.2.1
Scope	U	-		6.2.2.1
Filter	U	-	5.1.2.1	6.2.2.2
Access Control	U	-		6.2.4
Synchronization	U	-		6.2.2.3
Managed Object Class	-	C		6.4.5
Managed Object Instance	-	C		6.2.1
Current Time	-	U		6.2.3
Errors	-	C		6.3.6.1, 6.4.4

5.7.5 Log Modification, Suspension, and Resumption

5.7.5.1 Introduction

This SMF service allows one open system to request that another open system change the Administrative State attribute, or any other settable attribute, of a Log object.

The following informative table defines the mapping between LCF Log Modification, Suspension, and Resumption, OMF PT–Set, and CMIS M–SET service parameters. This tutorial information has been extracted from sections 9.4 and 11.2 of [LCF] and section 8.3.2 of [CMIS].

Table 6: Mapping Between LCF Log Modification, Suspension, and Resumption, OMF PT-Set, and CMIS M-SET Service Parameters

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	PT-Set & CMIS M-SET Parameter	Req	Rsp
Invoke Identifier	M	M(=)			
Linked Id	-	C			
Mode	M	-			
Base Object Class	M	-			

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	PT-Set & CMIS M-SET Parameter	Req	Rsp
Base Object Instance	M	-			
Scope	U	-			
Filter	U	-			
Access Control	U	-			
Synchronization	U	-			
Managed Object Class	-	C			
Managed Object Instance	-	C			
Discriminator Construct	U	C	Mod & Attribute List	M	
Administrative State	U	C	Mod & Attribute List	M	
Max Log Size	U	C	Mod & Attribute List	M	
Capacity Alarm Threshold	U	C	Mod & Attribute List	M	
Log Full Action	U	C	Mod & Attribute List	M	
Week Mask	U	C	Mod & Attribute List	M	
Intervals Of Day	U	C	Mod & Attribute List	M	
Start Time	U	C	Mod & Attribute List	M	
Stop Time	U	C	Mod & Attribute List	M	
Schedular Name	U	C	Mod & Attribute List	M	
Current Time	-	U			
Errors	-	C			

5.7.5.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Log Control Modification, Suspension, and Resumption SMF service defined by section 9.4 of [LCF]. Relevant CMIS agreements defined in subclause 6.3.3 are repeated here for completeness.

Table 7: Agreements On Parameter Usage Pertinent to the Log Control Modification, Suspension, and Resumption SMF Service

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Linked Id	-	C		6.4
Mode	M	-	5.1.2.2	
Base Object Class	M	-		6.4.5
Base Object Instance	M	-		6.2.1
Scope	U	-		6.2.2.1
Filter	U	-	5.1.2.1	6.2.2.2
Access Control	U	-		6.2.4
Synchronization	U	-		6.2.2.3
Managed Object Class	-	C		6.4.5
Managed Object Instance	-	C		6.2.1
Discriminator Construct	U	C	[1], 5.1.2.1	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Administrative State	U	C		6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Max Log Size	U	C	[5]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Capacity Alarm Threshold	U	C		6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Log Full Action	U	C		6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Week Mask	U	C	[3]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Intervals Of Day	U	C	[2]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Start Time	U	C	[3]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Stop Time	U	C	[3]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Schedular Name	U>I	C>I	[4]	6.4.5, 6.3.3.1, 6.3.3.3, 6.3.3.4
Current Time	-	U		6.2.3
Errors	-	C		6.3.3.2, 6.4.4

[1] As specified in [CMIP], the value "AND {}" shall be used to represent an all–pass Discriminator Construct.

[2] The Daily Scheduling Package, if supported by an object, shall support at minimum the default 24 hour interval.

[3] The Weekly Scheduling Package, if supported by an object, shall support the default values for Start Time and Stop Time attributes. The Week Mask attribute shall support scheduling for each day of the week, and, at a minimum, the default 24 hour period for intervals of the day.

[4] Support for the External Schedular Package is beyond the scope of these agreements.

[5] 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.

5.7.6 Retrieving Logging Attributes

5.7.6.1 Introduction

This SMF service allows one open system to retrieve any of the readable attributes of the log using the PT–Get SMF service.

5.7.6.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Log Control Retrieving Logging Attributes SMF service defined by section 9.5 of [LCF]. No agreements have been made beyond those defined for the PT–Get SMF service; refer to subclause 5.2.10 of these agreements.

Editor's Note: [A table will be added to this subclause if any additional LCF agreements are defined.**]**

5.7.7 Retrieval Of Log Records

5.7.7.1 Introduction

This SMF service allows one open system to retrieve log records from a log using the PT–Get SMF service.

5.7.7.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Log Control Retrieval Of Log Records SMF service defined by section 9.6 of [LCF]. No agreements have been made beyond those defined for the PT–Get SMF service; refer to subclause 5.2.10 of these agreements.

Editor's Note: [A table will be added to this subclause if any additional LCF agreements are defined.**]**

5.7.8 Deletion Of Log Records

5.7.8.1 Introduction

This SMF service allows one open system to delete log records from a log using the PT–Delete SMF service.

5.7.8.2 Agreements On Parameter Usage

This subclause provides agreements pertinent to the Log Control Deletion Of Log Records SMF service defined by section 9.7 of [LCF]. No agreements have been made beyond those defined for the PT−Delete SMF service; refer to subclause 5.2.7 of these agreements.

Editor's Note: [A table will be added to this subclause if any additional LCF agreements are defined.**]**

5.8 Security Alarm Reporting Function Agreements

5.8.1 Introduction

This subclause provides agreements pertinent to the Security Alarm Reporting Function defined by [SARF].

The Security Alarm Reporting Function:

* defines a set of management information pertinent to security alarms; and

* provides a security alarm reporting service.

The Security Alarm Reporting Function [SARF] alerts the user that misoperation, attack, or breach of security has occurred. Security alarms may indicate, for example:

* duplicate information received (potential replay attack),
* information modification detected (data integrity violated),
* cable tamper detected (potential breach of a physical resource),
* intrusion detected (site entered illegally),
* authentication failed (potential masquerade), or
* time domain violation (event occurred outside permitted time period).

The Security Alarm Reporting Function provides facilities for the creation, deletion, and modification of an event forwarding discriminator such that the control of security alarm reporting may be achieved.

The Security Alarm Reporting Function makes use of the following management support objects:

securityAlarmReportRecord.

The Security Alarm Reporting Function makes use of the following attributes, in addition to those attributes defined for the object class top:

eventType,
eventTime,
securityAlarmCause,
securityAlarmSeverity,
securityAlarmGenerator,
serviceUser,
serviceProvider,

notificationId,
correlatedNotifications,
securityAlarmText, and
securityAlarmData.

The Security Alarm Reporting Function makes use of the following notification types:

integrityViolation,
operationalViolation,
physicalViolation,
securityServiceOrMechanismViolation, and
timeDomainViolation.

5.9 Security Audit Trail Function Agreements

5.9.1 Introduction

This subclause provides agreements pertinent to the Security Audit Trail Function defined by [SATF].

The Security Audit Trail Function:

* defines a set of management information pertinent to security audit trails;

* provides the Security Audit Trail Record management support object class; and

* provides a security audit trail notification service.

The Security Audit Trail Function maintains an ongoing record of all potential security–related events that occur in the managed IIE. Security–related events that may be subject to security auditing include, for example:

* communications resource connect and disconnect,
* user registration/deregistration,
* invocation of security mechanisms (e.g., attempted access), or
* attempted and successful management operations.

The security policy of an open system may require that certain event reports be sent to a security audit trail log in the same or in a different open system. In this case, the event reports shall be logged according to the procedures defined in [LCF]. The discriminator construct within the audit trail log shall be specified so as to permint the capture of incoming events that the security policy requires to be logged. If the event reports are to be sent to a different destination, then event forwarding discriminators as defined in [ERF] shall be created and the destination address shall be set to send the event to the system where the selected security audit trail log is located. The security audit trail log is a log as defined in [LCF].

The Security Audit Trail Function makes use of the following management support objects:

securityAuditTrailRecord.

The Security Audit Trail Function makes use of the following attributes, in addition to those attributes defined for the object class top:

eventType,
eventTime,
serviceReportCause,
notificationId,
correlatedNotifications,
securityAuditText, and
securityAuditData.

The Security Audit Trail Function makes use of the following notification types:

serviceReport, and
statisticsReport.

5.10 Objects and Attributes for Access Control Agreements

5.10.1 Introduction

This subclause provides agreements pertinent to Objects and Attributes for Access Control defined by [OAAC].

Objects and Attributes for Access Control:

* defines a conceptual model for the administration of managed object access control; and
* provides the Access Control Descriptor, Target Access Control Information, and Authorized Initiators management support object classes to facilitate object access control.

There is a need to prevent unauthorized access to management resources at various levels:

* management notifications must not be sent to unauthorized recipients,
* unauthorized initiators must not have access to management operations, and
* management information must be protected from unintended disclosure.

This function defines mechanisms for controlling access to management associations and operations.

Objects and Attributes for Access Control makes use of the following management support objects:

accessControlDescriptor,
targetACI, and
authorisedInitiators.

Objects and Attributes for Access Control makes use of the following attributes, in addition to those attributes defined for the object class top:

accessControlDomainNames,
accessControlPolicyName,
ACDName,
ACDRules,
ACIOperations,
ACIRules,
AIName,
defaultRules,
globalRules,
initiatorACI,
initiatorList,
MIOperations,
MIRules,
objectList, and
targetACIName.

Objects and Attributes for Access Control makes use of the following notification types:

objectCreation,
objectDeletion,
attributeChange, and
securityServiceOrMechanismViolation.

5.11 Accounting Meter Function Agreements

5.11.1 Introduction

This subclause provides agreements pertinent to the Accounting Meter Function defined by [AMF].

The Accounting Meter Function:

* defines a conceptual model for collecting, recording, and reporting accounting information;

* provides a set of management information pertinent to account metering;

* provides the Accounting Record, Accounting Meter Control, and Accounting Meter Data management support object classes;

* provides a number of notifications regarding account metering; and

* provides a set of services to effect account metering.

In general, any accounting activity begins by monitoring resources to identify who is using them and to what extent they are being used. An accounting meter records the use of a resource in the form of accounting records or logs. Accounting meters record information such as:

* the identity of the user and the resource,
* the quality and type of service requested and provided,
* the usage start time and current time,
* the current state of usage (running or suspended), and
* the unit of measurement and number of units consumed.

The Accounting Meter Function defines the following management support objects:

accountingMeterControlObject,
accountingMeterDataObject, and
accountingRecordObject.

The Accounting Meter Function defines the following attributes:

controlObjectReference,
dataObjectReference,
dataObjectState,
meterInfo,
notificationCause,
notificationTime,
recordingTrigger,
reportingTrigger,
requesterId,
responderId,
resourceName,
serviceProvided,
serviceRequested,
subscriberId,
unitsOfUsage,
usageMeterTime, and
usageStartTime.

The Accounting Meter Function defines the following notification types:

accountingStarted,
accountingSuspended,
accountingResumed,
accountingRecord, and
accountingInfoLost.

The Accounting Meter Function defines the following actions:

startMetering,
suspendMetering, and
resumeMetering.

5.12 Workload Monitoring Function Agreements

5.12.1 Introduction

This subclause provides agreements pertinent to the Workload Monitoring Function defined by [WMF].

The Workload Monitoring Function:

* defines three conceptual models for the monitoring of system resources;

* provides the Gauge Monitor Metric and Mean Monitor Metric management support objects to facilitate workload monitoring;

* provides a number of notifications regarding workload monitoring; and

* provides a set of services to effect workload monitoring.

Three conceptual models are defined within the Workload Monitoring Function.

* Utilization Model: Provides monitoring of instantaneous use of an OSI resource.

* Rejection Rate Model: Provides monitoring of service request rejection.

* Resource Request Rate Model: Provides monitoring of requests for usage of OSI resources.

Together, these three models provide an estimate of the workload for a managed resources.

The Workload Monitoring Function defines the following management support objects:

gaugeMonitor, and
meanMonitor.

The Workload Monitoring Function defines the following attributes:

adminstrativeState,
counterT,
counterTMinusDT,
derivedGauge,

derivedGaugeThold,
estimateOfMean,
estimateOfMeanThold,
gaugeMonitorId,
granularityPeriod,
meanMonitorId,
observedAttributeId,
observedObjectClass,
observedObjectInstance,
schedularName, and
timeConstant.

The Workload Monitoring Function references the following notification types:

attributeChange,
stateChange,
qualityOfServiceAlarm,
objectCreation, and
objectDeletion.

5.13 Summarization Function Agreements

5.13.1 Introduction

This subclause provides agreements pertinent to the Summarization Function defined by [SF].

The Summarization Function:

* defines a conceptual model for the summarization, reporting by notification, and logging of measurements pertaining to managed objects;

* provides the Measurement Summarization, Measurement Request, Observed Object Request, Running Summary Metric, Measures Threshold Control, and Measurement Object Summary Record management support object classes;

* provides a Measurement Summary notification to report summary information; and

* provides a set of services to effect measurement summarization.

The Summarization Function defines the following management support objects:

measurementSummarizationObject,
measurementRequest,
observedObjectRequest,
runningSummaryMetric,

measuresThresholdControl, and
measurementObjSummRecord.

At this time, the Summarization Function does not contain a complete list of services, attributes, or notifications.

5.14 Test Management Function Agreements

5.14.1 Introduction

This subclause provides agreements pertinent to the Test Management Function defined by [TMF].

The Test Management Function:

* defines a conceptual model for the initiation, control and execution of tests and reporting of test results;
* provides the Test Results Record management support object;
* provides a Test Result notification for information reporting;
* provides a set of services to effect test management.

The Test Management Function defines the following management support objects:

testResultsRecord.

The Test Management Function defines the following attributes:

testSessionId,
testState,
testOutcome,
mOTS,
associatedObjects, and
timeoutPeriod.

The Test Management Function defines the following notification types:

testResultNotification.

The Test Management Function defines the following actions:

testRequestAsyncAction,
testRequestSyncAction,
testSuspendResumeAction, and

testTerminateAction.

5.15 Confidence and Diagnostic Test Classes Agreements

5.15.1 Introduction

This subclause provides agreements pertinent to the Confidence and Test Classes defined by [TMF].

Confidence and Diagnostic Test Classes:

* identifies certain characteristics which are common to all classes of tests;

* identifies general test categories;

Confidence and Diagnostic Test Classes defines the following management support objects:

internalResourceResultsRecord,
connectivityResultsRecord,
dataIntegrityResultsRecord,
loopbackResultsRecord, and
protocolIntegrityResultsRecord.

Confidence and Diagnostic Test Classes defines the following attributes:

effectiveTime,
establishmentTime,
testDuration, and
loopCounter.

6 Management Communications

(Refer to the Stable Implementation Agreements Document.)

6.1 Association Policies

(Refer to the Stable Implementation Agreements Document.)

6.1.1 Application Context Negotiation

(Refer to the Stable Implementation Agreements Document.)

6.1.2 Functional Unit Negotiation

(Refer to the Stable Implementation Agreements Document.)

6.1.3 Security Aspects of Associations

The ACSE authentication mechanisms and associated data types shall be as defined in clause 8 (Upper Layers Security) of part 13 of the OIW Working Agreements.

Support of ACSE authentication is optional.

6.2 General Agreements on Users of CMIS

(Refer to the Stable Implementation Agreements Document.)

6.2.1 Object Naming

(Refer to the Stable Implementation Agreements Document.)

6.2.2 Multiple Object Selection

(Refer to the Stable Implementation Agreements Document.)

6.2.3 Current/Event Time

(Refer to the Stable Implementation Agreements Document.)

6.2.4 Access Control

(Refer to the Stable Implementation Agreements Document.)

6.2.5 CMIS Subsets

Editor's Note: **[**The header and text for this subclause (6.2.5) are intended to supersede the text for this subclause in the Stable Agreements. This text constitutes technical errata planned to be voted as technical errata to the Stable Agreements in September 1991. They are entered here in the Working Agreements to allow for review prior to that action.**]**

Currently the only way to subset CMIP capabilities is through the use of CMIP Functional Units (FUs).

An implementation shall support CMIP FUs as indicated by one of the following defined subsets:

		FUNCTIONAL UNITS					
		Kernel	Multiple Reply	Multiple Object Selection	Cancel Get	Filter	Extended Service
SUBSET	Basic	M	O	I	I	I	I
	Enhanced	M	M	M	M	M	I

M = Mandatory O = Optional I = Out of Scope

On each association, use of functional units shall be negotiated as specified in [CMIS].

6.3 Specific Agreements on Users of CMIS

(Refer to the Stable Implementation Agreements Document.)

6.4 Specific Agreements on CMIP

(Refer to the Stable Implementation Agreements Document.)

6.4.1 Invoke/Linked Identifier Size

(Refer to the Stable Implementation Agreements Document.)

6.4.2 Version

(Refer to the Stable Implementation Agreements Document.)

6.4.3 Linked Reply Values

(Refer to the Stable Implementation Agreements Document.)

6.4.4 Error Codes

(Refer to the Stable Implementation Agreements Document.)

6.4.5 Parameters

Editor's Note: **[**The following text for this subclause (6.4.5) is intended to be added as the final paragraph for this subclause in the Stable Agreements. This text constitutes a technical erratum planned to be moved to the Stable Agreements in September 1991. It is entered here in the Working Agreements to allow for review prior to that action.**]**

Implementations shall receive all forms of all ASN.1 CHOICE types and values defined in [CMIP]. Receipt shall require passing the value to the CMIS user.

6.4.6 Access Control Parameter

Editor's Note: **[**This entire subclause (6.4.6) is intended to be added to the Stable Agreements. This text constitutes a technical erratum planned to be moved to the Stable Agreements in September 1991. It is entered here in the Working Agreements to allow for review prior to that action.**]**

An implementation shall be able to send the accessControl parameter.

6.5 Services Required by CMIP

(Refer to Stable Implementation Agreements Document.)

Authentication -- if peer entity authentication is supported.

6.5.1 P-DATA Encoding

For encoding of each CMIP/ROSE PDU in a P–DATA, implementations shall be able to parse and process a maximum of 10,240 octets as they would be encoded in the Presentation "User–data" type according to the Basic Encoding Rules for ASN.1.

6.6 CMIP PICS

Refer to "Profile AOM12: Full CMIP for Managing and Managed Systems" [SD35].

7 Management Information

(Refer to the Stable Implementation Agreements Document.)

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 conformance requirements and criteria which shall be used as a basis for tests of implementations claiming conformance to these agreements.

8.2 General Requirements of Conformance

Conformance for these agreements is designed to specify a well–defined set of management capabilities and features. In addition, a taxonomy of managed object classes is needed. For the purposes of organization and clarity of these agreements, management has been divided into three classification areas. Clauses 5 (Management Functions and Services), 6 (Management Communications) and 7 (Management Information) state the agreements which respectively comprise three conformance classification areas. Within these classification areas, particular conformance classes 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 classification area, a conformance class is specified which defines conformance to the State Management Function). Once a conformance class is delineated which specifies the set of requirements for that class, tests can be developed to evaluate conformance of products to that conformance class. And finally, for some conformance classes, roles (Manager, Agent, or Manager/Agent) are specified. One or more roles may be supported for those conformance classes to which an implementation is conformant.

The development of conformance classes will enable:

1) users to define procurement specifications.

2) vendors to define management capabilities and features.

3) 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 classification areas:

- o Management Communication
- o Management Functions and Services
- o Management Information

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

8.3 Specific Conformance Classification Areas

8.3.1 Management Communication

8.3.1.1 General Management Communication

To be conformant within the General Management Communication classification area, an implementation shall conform to agreements in clause 6. Conformance to management communication also requires an implementor to state which of the CMIP susets defined in clause 6 (e.g., CMIP functional units) are supported in the implementation. The implementor's statement of which CMIP subset is supported shall be indicated by in a CMIP PICS as follows. For an implementation of the enchanced management communication profile, the implementor shall complete the AOM12 pDISP (ISO/IEC JTC1 SGFS pDISP 11183 Part 1 and 2 — AOM12 (EWOS/EG/NM91/072)). For an implementation of the basic management communication profile, the implementor shall complete the AOM11 pDISP.

Editor's Note: [pDISP should be changed to ISP or ISPICS??**]**

An implementation which is conformant to the management communication classification area of these agreements shall not violate the protocol requirements specified in clause 6 of these agreements. Every implementation shall respond appropriately to correct and erroneous PDUs.

Conformance to agreements in clause 6 requires conformance to referenced ISO standards/CCITT Recommendations and to all other clauses referenced in 6, including the underlying services required by CMIP.

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**]**

8.3.1.2 Dependent Management Communication

Editor's Note: [See proposed Errata to 18.6.2.5 regarding CMIS subsets.**]**

8.3.2 Management Functions and Services

Editor's Note: [This clause needs to be harmonized with 18.5.x (for Arles output (IS) alignment).**]**

To be conformant within the Management Functions and Services classification area, an implementation shall conform to the agreements in clause 5 on at least one Systems Management Function in either a manager or agent role. Conformance to agreements in clause 5 requires conformance to referenced ISO standards/CCITT Recommendations and to all other clauses referenced in 5, including the underlying services required by the SMFs.

The implementor shall state which Systems Management Functions are supported, and for each SMF, the implementor shall state which functional unit(s) and role(s) it supports.

To be conformant to the Object Management Function, an implementation shall conform to the requirements stated in [OMF]. In addition, an implementation supporting the AllEvents functional unit shall conform to clauses 5.2.2, 5.2.3, 5.2.5, and 5.2.11. The implementation shall also conform to all other clauses referenced by these clauses. An implementation supporting the PT–Operations functional unit shall conform to clauses 5.2.6, 5.2.7, 5.2.8, 5.2.9, and 5.2.10. The implementation shall also conform to all other clauses referenced by these clauses.

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 and all clauses referenced in 5.3.

To be conformant to the Attributes for Representing Relationships SMF, an implementation shall conform to the requirements stated in [ARR]. In addition, an implementation shall conform to clause 5.4 and all 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 and all clauses referenced in 5.5.

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 and all clauses referenced in 5.6.

Editor's Note: [The SMF clauses (18.5) have parameters specified which are conditional. The base standard SMFs are ambiguous with respect to the conditions under which parameters are present. This ambiguity was pointed out to the U.S. delegates (in X3T5.4) to the ISO SC21 editing meetings in France. It was agreed that clarification of conditionality in SMF tables is needed. Presumably this will be resolved at the ISO SMF editing meetings.**]**

8.3.3 Management Information

To be conformant within the Management Information classification area, an implementation shall include at least one managed object defined according to clause 7. Furthermore, this object shall use OIW management communication as specified in clause 6 and, as needed, one or more OIW SMFs as specified in clause 5. Managed object class definitions shall be provided either in full or by reference. Associated with any such managed object definition shall be a registered managed object class identifier. All mandatory abstract syntaxes and semantics associated with that identifier shall be used. Note that all managed objects defined in the OIW Network Management MIL shall 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. The implementor shall supply all necessary information to allow access to the managed object.

Editor's Note: [Issue: Should "conformant to a managed object" really be "conformant to a managed object class definition"?**]**

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.

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

- o a statement of pragmatic constraints (e.g., attribute values/ranges, initial values) supported, unless such constraints are defined in the managed object class definition;
- o a statement of conditional packages supported;
- o a statement of role(s) (manager, agent, or manager/agent) in which the object class definition is supported.

8.3.4 Management Application Contexts

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

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

Editor's Note: [This needs to be harmonized with IS version of SMO (10040).**]**

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

8.4.1 Management Communication

To demonstrate conformance to the management communication general conformance classification area claimed to satisfy clause 8.3.1.1, the system shall contain a test or other OIW Network Management MIL object that can be addressed in all CMIP PDUs. All capabilities stated as supported in the PICS shall be demonstrated to be correctly implemented. The test object to be used is **TBD??**

Editor's Note: [The NMSIG will examine CTS-3 CMIP project for such a test object. The OSI/NM Forum uses a similar test object.**]**

Editor's Note: [Demonstration of conformance to the management communication dependent classification area claimed to supported is **TBD??]**

8.4.2 Systems Management Functions

To demonstrate conformance to the systems management functions classification area claimed to satisfy clause 8.3.2, the system must support a communication profile that satisfies clause 8.3.1 and a real object importing the SMF or the test object defined in **TBD??**

Editor's Note: [This needs to be harmonized with revised clause 5 (after Arles alignment).**]**

8.4.3 Management Information

Conformance to the management information classification area is provided through conformance to managed objects. To demonstrate conformance to the managed objects, the system shall support a conformant communications profile that satisfies that part of clause 8.3.1 required to test the managed object.

Editor's Note: [There is an issue here regarding general and dependent conformance. We should await output text from the May/June ISO SC21 Editing meetings for possible resolutions.**]**

For conformance to an object supported in the Agent role, the implementation shall demonstrate that all appropriate PDUs for the defined attributes and functions claimed to be supported in the MOCS can be correctly received and transmitted. Appropriate behavior of the implementation in response to received PDUs shall be demonstrated.

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.

Annex A (informative)

Management Information Library (MIL)

MANAGEMENT INFORMATION LIBRARY

(MIL)

Version 6.0

December 10, 1990

A.1 INTRODUCTION

This document is produced by the OSI MIB Working Group (a subgroup of the NMSIG). It provides definitions of management information - managed object classes, name bindings, attributes, actions and notifications. Provision of these definitions is made by a) references to standards' documents that contain these definitions, or b) inclusion of the actual definitions in this document; in which case they will be registered in the NMSIG arc of the ISO ASN.1 Object Identifier Tree.

Management information definitions provided by the OSI MIB Working Group have been introduced to accelerate the process of defining management information. They are intended to be implementable but also serve as a basis from which other implementations may define refinements or alternatives. These definitions do not override those provided by standards' groups or other OIW SIGs.

Editor's Note: [The intention is to progress these definitions to an International Management Information Library.**]**

Managed objects in the MIL are not normative as far the NMSIG IAs are concerned. Implementors do not have to support any of the MIL managed objects; they may choose to define their own managed objects using the agreements on [GDMO] specified in clause 18.7. However, supporting managed objects from the MIL will increase the potential for interoperability with other network management implementations.

Editor's Note: [Following is proposed text for this section, to replace all the above text:

The Management Information Library provides definitions of management information - managed object classes, name bindings, attributes, actions and notifications. Provision of these definitions is made by a) references to standards' documents that contain these definitions, or b) inclusion of the actual definitions in this document; in which case they are registered in the NMSIG arc of the ISO ASN.1 Object Identifier Tree.

The reasons why the NMSIG has opted to define management information are as follows:

(i) There is an urgent need for network management within the community. Managed objects are critical ingredients of network management; but standards' defined managed objects that represent network/system resources are not available yet. However, there does exist an ISO standard that specifies guidelines for defining managed objects : [GDMO]. Different organizations, including private companies, etc, can use [GDMO] to define their own managed objects. However, two network management implementations can interoperate only if there is a common subset of managed objects supported on both sides. The NMSIG has used the [GDMO] standard to define "public domain" managed objects that meet the needs of the community and foster interoperability.

(ii) Standards' groups are not addressing all the network/system resources that need to be managed; i.e. there is no standards' activity for defining managed objects that represent such resources. The NMSIG has attempted to fill these holes by defining managed objects for these resources, and thus fulfil the needs of the community.

As mentioned earlier, managed objects in the MIL have been provided to foster interoperability. They are not normative as far the NMSIG IAs are concerned. Implementors do not have to support any of the MIL managed objects; they may choose to define their own managed objects using the agreements on [GDMO] specified in Section 18.7. However, supporting managed objects from the MIL will increase the potential for interoperability with other network management implementations.

The NMSIG defined managed objects in the MIL are intended to be implementable but they also serve as a basis from which other implementations may define refinements or alternatives. These definitions do not override or duplicate those provided by standards' groups or other OIW SIGs.

More specifically, the transport and network layer managed objects that have been defined in the MIL are "generally applicable" objects, in that they do not represent any particular transport or network layer protocols, but contain characteristics common across different transport or network layer protocols. These managed objects provide a high level view of the transport and network layers, and are especially useful in managing heterogeneous networks that support various different types of transport and network layer protocols. These managed objects do not override the OSI Transport and Network Layer managed objects that are being defined in ISO. The ISO specified OSI Transport and Network Layer managed objects are "specific" managed objects that represent strictly the OSI Transport and Network protocol layers.**]**

A.2 RULES AND PROCEDURES

The following rules and procedures apply to managed object class definitions that are to be included in the MIL :

(i) All managed object class definitions provided by the MIL must comply with the NMSIG (ISO) object templates.

(ii) A managed object class definition provided by the MIL must represent an abstraction of an identifiable logical or physical resource that can be managed via OSI management.

(iii) All managed object classes in the MIL will have registered ASN.1 object identifiers assigned either by a standards' body if it is defining the managed object class, or, if the managed object class definition is being progressed within the NMSIG, by the NMSIG in its branch of the ISO Registration Tree.

(iv) A managed object class will be selected as a candidate for inclusion into the MIL if there are at least two NMSIG members from different companies who express a requirement (strong interest) for the managed object class. If this is not a standards' defined managed object class, then there must be at least one NMSIG member who is committed to developing the definition of the managed object class.

(v) A managed object class selected for the MIL will be given a priority based on the number of members who express interest in it.

(vi) All managed object class definitions that are proposed for inclusion into the MIL will undergo a review process within the NMSIG. NMSIG member defined managed object classes will additionally undergo a ballotting process. If problems are found with a standards' defined managed object class, the appropriate standards' body will be approached. If problems are found with a member defined managed object class, it will be returned with comments.

(vii) Based on its priority, there will be a call for contributions on the definition of a managed object class at an NMSIG meeting. Contributions could be in the form of a) identification of a standards' body that is currently working on the definition, or b) an NMSIG member definition of the managed object class.

(viii) An element of management information, once registered, i.e., given an ASN.1 Object Identifier, will never be deleted from the Registration Tree (ASN.1 Object Identifier tree). It may, however, fall into disuse due to lack of requirements for it.

A.3 GENERAL GUIDELINES

It is recommended that the following guidelines be used in general for all managed object definitions, unless there is a specific exception condition:

a) For the objectCreation Notification, send all the attributes of the created managed object instance in the CreateInfo field.

A.4 MANAGED OBJECT CLASSES

A.4.1 NMSIG Agent

A.4.1.1 NMSIG Agent Definition

```
nmsig-agent    MANAGED OBJECT CLASS
  DERIVED FROM  ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
  CHARACTERIZED BY  nmsig-agent-Package;

REGISTERED AS  {nmsig-objectClass 1};
```

A.4.1.2 NMSIG Agent Package

```
nmsig-agent-Package   PACKAGE
        BEHAVIOUR DEFINITIONS   agent-behaviour;
    ATTRIBUTES   nmsig-agentId  GET;

REGISTERED AS {nmsig-package 1};
```

A.4.1.3 NMSIG Agent Behaviour

agent-behaviour BEHAVIOUR

DEFINED AS

This managed object class represents an NMSIG agent system, which is an open system that supports the NMSIG agreements to make one or more managed objects visible to other open systems that support the NMSIG agreements.

An NMSIG agent system may not support more than one instances of the NMSIG Agent managed object class. If supported, this instance is assumed to be pre-existent when the NMSIG agent system comes up; i.e. management CREATE or DELETE is not supported.

At this time, the NMSIG Agent managed object class only serves to name management support managed objects (e.g. EventForwardingDiscriminator).

;

A.4.2 NMSIG Computer System

Editor's Note: [A model has been proposed for defining managed object classes related to computers, as follows : The philosophy behind the proposed model is to define a composite or aggregrate managed object class called "computerSystem" that provides a high level view of a computer system, including its physical and logical, as well as its hardware and software components. Detailed views of these components are then modelled as object classes contained within the computerSystem object class, as shown in the CONTAINMENT TREE below. (NOTE : This is NOT an inheritance tree)

```
                 computerSystem
                       |
                       |
 ---------------------------------------------------------........
    |         |     |      |       |     |      |
tapeDrive     |   printer  |       |     |   applicationX  ........
         discDrive     processing  |    os
                        Entity     |
                                   |
                       coTransportProtocolLayerEntity
                                   |
                          transportConnection
```

A great benefit provided by this model is flexibility. As and when more computer components need to be specified, they can be defined as individual object classes and "plugged" into the above structure under computerSystem, without upsetting the other object classes.

The 'system' managed object class defined in [DMI] was not used because it's definition was considered to be inappropriate.**]**

A.4.2.1 NMSIG Computer System Definition

```
nmsig-computerSystem    MANAGED OBJECT CLASS
  DERIVED FROM  ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
  CHARACTERIZED BY  nmsig-computerSystem-Package;
REGISTERED AS  {nmsig-objectClass 2};
```

A.4.2.2 NMSIG Computer System Package

```
nmsig-computerSystem-Package   PACKAGE
        BEHAVIOUR DEFINITIONS  computerSystem-behaviour;
    ATTRIBUTES   nmsig-systemId  GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
             administrativeState  GET-REPLACE,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
             operationalState   GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
             usageState  GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
```

```
            managementState  GET,
            nmsig-systemTime   GET,
            nmsig-peripheralNames  GET,
            nmsig-userFriendlyLabel  GET-REPLACE;
    NOTIFICATIONS   ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectCreation,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectDeletion,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            attributeValueChange,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            stateChange,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            processingErrorAlarm,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            environmentalAlarm,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            equipmentAlarm;

REGISTERED AS {nmsig-package 2};
```

A.4.2.3 NMSIG Computer System Behaviour

computerSystem-behaviour BEHAVIOUR

DEFINED AS

The nmsig-computerSystem managed object class is a composite or aggregate object class that provides a high level view of a general purpose business computer system, including its physical, logical, hardware and software components.

The Computer System Package supports all the values of the administrative and operational states. The values supported by the usage state are implementation specific; the 'idle' and 'unknown' values are to be used for computer systems that do not keep track of their usage.

The 'enabled' value of the operational state indicates that the underlying computer system resources are together capable of providing minimal computing services. These enabled resources may or may not be modelled as managed objects, and may or may not include the entire set of resources which together are viewed as the computer system.

The 'disabled' value of the operational state indicates that the underlying computer system resources are incapable of providing minimal services at the current time.

The peripheralNames attribute specifies the names of auxiliary devices that are used by the underlying computer system resource.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created computer sytem instance.

The additionalDeleteInfo field of the objectDeletion notification shall be NULL.

Attributes that are subject to the attributeValueChange notification are :
nmsig-peripheralNames, nmsig-userFriendlyLabel.
Attributes that are subject to the stateChange notification are :
administrativeState, operationalState and usageState.

;

A.4.3 NMSIG Connection Oriented Transport Protocol Layer Entity

A.4.3.1 NMSIG CO Transport Protocol Layer Entity Definition

```
nmsig-coTransportProtocolLayerEntity    MANAGED OBJECT CLASS

   DERIVED FROM      ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
   top;
   CHARACTERIZED BY  nmsig-coTransportProtocolLayerEntity-Package
                 nmsig-productInfo-Package;

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

A.4.3.2 NMSIG CO Transport Protocol Layer Entity Package

```
nmsig-coTansportProtocolLayerEntity-Package   PACKAGE
           BEHAVIOUR DEFINITIONS  coTransportProtocolLayerEntity-behaviour;
      ATTRIBUTES       nmsig-coTransportProtocolLayerEntityId  GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      administrativeState  GET-REPLACE,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      operationalState  GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      usageState        GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      managementState   GET,
                      nmsig-localTransportAddresses  GET,
                      nmsig-maxConnections  GET-REPLACE,
                      nmsig-openConnections  GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      outgoingConnectionsRequestCounter  GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      incomingConnectionsRequestCounter  GET,
                 ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      outgoingConnectionRejectErrorCounter  GET,
```

```
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    incomingConnectionRejectErrorCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    outgoingDisconnectErrorCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    incomingDisconnectErrorCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    incomingDisconnectCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    outgoingDisconnectCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    octetsSentCounter   GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    octetsReceivedCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    incomingProtocolErrorCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    outgoingProtocolErrorCounter  GET,
                    nmsig-checksumTPDUsDiscardedCounter  GET,
                    nmsig-transportEntityType GET,
                    nmsig-entityUpTime GET;
    NOTIFICATIONS
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    objectCreation,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    objectDeletion,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    attributeValueChange,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    stateChange,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    processingErrorAlarm,
                    nmsig-counterWrap,

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

A.4.3.3 NMSIG CO Transport Protocol Layer Entity Behaviour

```
coTransportProtocolLayerEntity-behaviour  BEHAVIOUR

  DEFINED AS
```

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 being 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.

The managed object class nmsig-coTransportProtocolLayerEntity represents an instantiation of any connection-oriented transport layer protocol e.g. the ISO Transport Protocol layer or the Internet Transmission Control Protocol (TCP). 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.

The connection-oriented transport protocol layer entity Package supports all values of the administrative and operational states.

The 'enabled' value of the operational state indicates that the underlying transport protocol layer entity resource is capable of supporting transport connections but currently has no open transport connections.

The 'disabled' value of the operational state indicates that the underlying transport protocol layer entity resource is not capable of supporting any transport connections.

The 'active' value of the usage state indicates that the underlying transport protocol layer entity resource is currently supporting at least one transport connections and is capable of supporting additional transport connections.

The 'busy' value of the usage state indicates that the underlying transport protocol layer entity resource is supporting the maximum number of transport connections that it is capable of supporting.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created connection-oriented transport protocol layer entity instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted connection-oriented transport protocol layer entity instance.

Attributes that are subject to the attributeValueChange notification are : nmsig-localTransportAddresses, nmsig-maxConnections.

Attributes that are subject to the stateChange notification are :
administrativeState, usageState and operationalState.

The counterWrap notification is emitted when any of the counter attributes wrap.

;

A.4.3.4 NMSIG Product Info Package

```
12nmsig-productInfo-Package   PACKAGE
          BEHAVIOUR DEFINITIONS  productInfo-behaviour;
     ATTRIBUTES          nmsig-manufacturerInfo  GET,
                    nmsig-productLabel  GET,
                    nmsig-release      GET,
                    nmsig-serialNumber  GET;

REGISTERED AS        {nmsig-package 4};

productInfo-behaviour  BEHAVIOUR

  DEFINED AS

  This package specifies product information of the underlying resource.
  ;
```

A.4.4 NMSIG Connectionless Network Protocol Layer Entity

A.4.4.1 NMSIG Connectionless Network Protocol Layer Entity Definition

```
nmsig-clNetworkProtocolLayerEntity    MANAGED OBJECT CLASS

   DERIVED FROM    ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
   CHARACTERIZED BY  nmsig-clNetworkProtocolLayerEntity-Package,
               nmsig-productInfo-Package;
   CONDITIONAL PACKAGES
          nmsig-clNetworkProtocolLayerEntityRedirection-Package
               PRESENT IF connectionless network protocol layer entity supports redirection
                              of received PDUS;

REGISTERED AS  {nmsig-objectClass 4}
```

A.4.4.2 NMSIG Connectionless Network Protocol Layer Entity Package

```
nmsig-clNetworkProtocolLayerEntity-Package   PACKAGE
          BEHAVIOUR DEFINITIONS  clNetworkProtocolLayerEntity-behaviour;
     ATTRIBUTES          nmsig-clNetworkProtocolLayerEntityId  GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    administrativeState  GET-REPLACE,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    operationalState  GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    usageState GET,
               ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                    managementState  GET,
                    nmsig-localNetworkAddresses  GET,
```

```
            nmsig-nPDUTimeToLive  GET-REPLACE,
            nmsig-maxPDUSize   GET,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            pDUsSentCounter  GET,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            pDUsReceivedCounter  GET,
            nmsig-PDUsForwardedCounter  GET,
            nmsig-PDUsReasmbldOKCounter  GET,
            nmsig-PDUsReasmblFailCounter  GET,
            nmsig-PDUsDiscardedCounter   GET,
            nmsig-networkEntityType GET,
            nmsig-entityUpTime GET;

    NOTIFICATIONS
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectCreation,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectDeletion,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            attributeValueChange,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            processingErrorAlarm,
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            stateChange,
            nmsig-counterWrap;

REGISTERED AS      {nmsig-package 5};
```

A.4.4.3 NMSIG Connectionless Network Protocol Layer Entity Behaviour

clNetworkProtocolLayerEntity-behaviour BEHAVIOUR

DEFINED AS

This is a generally applicable managed object class, in that it does not represent any specific connectionless network protocol -rather 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 being 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.

This 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. The connectionless network protocol layer does not make use of a network connection for the purposes of transferring data. No dynamic peer to peer agreement is involved in the process of data transfer.

An instance of this managed object class supports only one type of protocol and one address domain.

The NMSIG connectionless network protocol layer entity Package supports all the values of the administrative and operational state attributes. The values supported by the usage state are implementation specific; the 'idle' and 'unknown' values are to be used for connectionless network protocol layer entities that do not keep track of their usage.

The 'enabled' value of the operational state indicates that the underlying connectionless network protocol layer entity resource is capable of providing connectionless network layer services.

The 'disabled' value of the operational state indicates that the underlying connectionless network protocol layer entity resource is incapable of supporting any network services at the current time.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created connectionless network protocol layer entity instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted connectionless network protocol layer entity instance.

Attributes that are subject to the attributeValueChange notification are : nmsig-localNetworkAddresses, nmsig-nPDUTimeToLive.

Attributes that are subject to the stateChange notification are :
administrativeState, usageState and operationalState.

The counterWrap notification is emitted when any of the counter attributes wrap.

;

A.4.4.4 NMSIG CL Network Protocol Layer Entity Redirection Package

```
nmsig-clNetworkProtocolLayerEntityRedirection-Package  PACKAGE
      BEHAVIOUR DEFINITIONS  clNetworkProtocolLayerEntityRedirection-behaviour;
      ATTRIBUTES  nmsig-PDUsRedirected  GET;

REGISTERED AS  {nmsig-package 6};

clNetworkProtocolLayerEntityRedirection-behaviour  BEHAVIOUR

  DEFINED AS
```

This package reflects the redirection capability of the underlying connectionless network protocol layer entity resource.

;

A.4.5 NMSIG Equipment

30

A.4.5.1 NMSIG Equipment Definition

```
nmsig-equipment   MANAGED OBJECT CLASS
  DERIVED FROM   ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
  CHARACTERIZED BY  nmsig-equipment-Package,
              productInfo-Package;
REGISTERED AS  {nmsig-objectClass 5};
```

A.4.5.2 NMSIG Equipment Package

```
nmsig-equipment-Package  PACKAGE
    BEHAVIOUR DEFINITIONS  equipment-behaviour;
    ATTRIBUTES    nmsig-equipmentId  GET,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            operationalState  GET,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            administrativeState  GET-REPLACE,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            usageState  GET,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            managementState  GET,
            nmsig-locationName  GET-REPLACE,
            nmsig-contactNames  ADD-REMOVE, GET-REPLACE,
            nmsig-equipmentPurpose      GET-REPLACE,
            nmsig-vendorName    GET-REPLACE,
            nmsig-userFriendlyLabel  GET-REPLACE;

  NOTIFICATIONS
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            environmentalAlarm,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            equipmentAlarm,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectCreation,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            objectDeletion,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            attributeValueChange,
              ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
            stateChange;

REGISTERED AS  {nmsig-package 7};
```

A.4.5.3 NMSIG Equipment Behaviour

```
equipment-behaviour  BEHAVIOUR
```

DEFINED AS

The NMSIG equipment managed object class represents physical entities. Instances of this managed object class are located in specific geographic locations and support some type of functions. For example, a PBX, which may be regarded as an instance of this managed object class, performs switching functions. Multiplexers, amplifiers, and repeaters which can also be regarded as instances of this managed object class perform transmission functions. Equipment may be nested in equipment, thereby creating a containment relationship. For example, a line card is contained in an equipment shelf which is nested in a relay rack which is part of a switch.

Other organizations such as the OSI NM Forum and T1M1.5 have specified definitions for an equipment managed object class. The NMSIG has not adopted those definitions because a) they are not specified in the NMSIG supported version of [GDMO], and b) they do not use generic management information from [DMI].

Instances of this managed object class may be endpoints of a circuit or facility.

The NMSIG Contact Names attribute specifies who (persons or organizations) are to be contacted about the equipment.

The NMSIG Location Name attribute identifies where the equipment is located.

The NMSIG Vendor Name attribute identifies the organization from whom the equipment was obtained (i.e. purchased, leased, etc.).

The NMSIG equipment Package supports all permissible values of the administrative, usage and operational states.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created equipment instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted equipment instance.

Attributes that are subject to the attributeValueChange notification are : nmsig-locationName, nmsig-contactNames,
nmsig-equipmentPurpose, nmsig-vendorName, nmsig-userFriendlyLabel.

Attributes that are subject to the stateChange notification are :
administrativeState, usageState and operationalState.

;

A.4.6 NMSIG Network

A.4.6.1 NMSIG Network Definition

```
nmsig-network   MANAGED OBJECT CLASS

    DERIVED FROM  ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
       CHARACTERIZED BY  nmsig-network-Package;
REGISTERED AS {nmsig-objectClass 6};
```

A.4.6.2 NMSIG Network Package

```
nmsig-network-Package   PACKAGE
      BEHAVIOUR DEFINITIONS  network-behaviour;
      ATTRIBUTES   nmsig-networkId   GET,
                 nmsig-networkPurpose  GET,
                 nmsig-userFriendlyLabel  GET-REPLACE;

      NOTIFICATIONS
                   ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                  objectCreation,
                   ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                  objectDeletion,
                   ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                  attributeValueChange;

REGISTERED AS  {nmsig-package 8};
```

A.4.6.3 NMSIG Network Behaviour

```
network-behaviour  BEHAVIOUR

  DEFINED AS
```

The NMSIG Network managed object class represents a collection of connecting and interconnected resources (logical and physical) capable of exchanging information. A network may be contained in another network, thereby creating a superior/subordinate relationship.

Other organizations such as the OSI NM Forum and T1M1.5 have specified definitions for a network managed object class. The NMSIG has not adopted those definitions because a) they are not specified in the NMSIG supported version of [GDMO], and b) they do not use generic management information from [DMI].

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created network instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted network instance.

Attributes that are subject to the attributeValueChange notification are : nmsig-networkPurpose, nmsig-userFriendlyLabel.

;

A.4.7 NMSIG Processing Entity

A.4.7.1 NMSIG Processing Entity Definition

```
nmsig-processingEntity   MANAGED OBJECT CLASS
    DERIVED FROM   {nmsig-equipment};
       CHARACTERIZED BY  nmsig-processingEntity-Package;
REGISTERED AS {nmsig-objectClass };
```

A.4.7.2 NMSIG Processing Entity Package

```
nmsig-processingEntity-Package   PACKAGE
        BEHAVIOUR DEFINITIONS  processingEntity-behaviour;
        ATTRIBUTES      nmsig-cPU-Type  GET,
                      nmsig-memorySize  GET,
                      nmsig-osInfo   GET,
                      nmsig-entityUpTime  GET;
        NOTIFICATIONS
                    ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                      processingErrorAlarm;

REGISTERED AS  {nmsig-package 9};
```

A.4.7.3 NMSIG Processing Entity Behaviour

processingEntity-behaviour BEHAVIOUR

DEFINED AS

The NMSIG processing entity managed object class represents the physical portion of the computer system that performs the processing function. A processing entity may be composed of such components as arithmetic logic units (ALUs) registers for processing memory, limited storage often in the form of Random Access Memory (RAM), and various other types of memory used in the processing function. It does not include components such 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 may be viewed as components of a superior object, e.g. a computer system, or even shared among several computer systems.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created processing entity instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted processing entity instance.

Attributes, additional to those inherited from Equipment, that are subject to the attributeChange notification are :
nmsig-cPU-Type, nmsig-memorySize, nmsig-osInfo.

;

A.4.8 NMSIG Transport Connection

A.4.8.1 NMSIG Transport Connection Definition

```
nmsig-transportConnection          MANAGED OBJECT CLASS
        DERIVED FROM  ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
        CHARACTERIZED BY   nmsig-transportConnection-Package;
        CONDITIONAL PACKAGES
                nmsig-transportConnectionRetransmission-Package   PRESENT IF   transport protocol
supports                        retransmission;
REGISTERED AS {nmsig-objectClass 8};
```

A.4.8.2 NMSIG Transport Connection Package

```
nmsig-transportConnection-Package   PACKAGE
        BEHAVIOUR DEFINITIONS  transportConnection-behaviour;
        ATTRIBUTES          nmsig-transportConnectionId  GET,
                        nmsig-localTransportConnectionEndpoint GET,
                        nmsig-remoteTransportConnectionEndpoint GET,
                        nmsig-transportConnectionReference  GET,
                        nmsig-localNetworkAddress  GET,
                        nmsig-remoteNetworkAddress  GET,
                        nmsig-inactivityTimeout  GET,
                        nmsig-maxPDUSize  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                        pdusSentCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                        pdusReceivedCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                        octetsSentCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                        octetsReceivedCounter  GET,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                                peer GET
        NOTIFICATIONS
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
                        objectCreation,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
```

```
        objectDeletion,
    ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
        relationshipChange,
        nmsig-counterWrap;

REGISTERED AS {nmsig-package 10};
```

A.4.8.3 NMSIG Transport Connection Behaviour

transportConnection-behaviour BEHAVIOUR

DEFINED AS

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 being 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.

The managed object class nmsig-transportConnection 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.

The additionalCreateInfo field of the objectCreation notification shall contain all the attributes of the created transport connection instance.

The additionalDeleteInfo field of the objectDeletion notification shall contain all the attributes of the deleted transport connection instance. In addition it shall also contain a 'cause' field and a corresponding 'cause' attribute whose syntax is defined as follows:

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

[The full definition of this attribute is specified in the Attributes Section - Section 6 of the MIL.]

The counterWrap notification is emitted when any of the counter attributes wrap.

The relationshipChange notification is emitted whenever the peer attribute changes in value.

;

A.4.8.4 NMSIG Transport Connection Retransmission Package

```
nmsig-transportConnectionRetransmission  PACKAGE
    BEHAVIOUR DEFINITIONS  transportConnectionRetransmission-behaviour;
    ATTRIBUTES  nmsig-maxRetransmissions  GET,
          nmsig-retransmissionTimerInitialValue GET,
            ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
          pdusRetransmittedErrorCounter  GET,
            ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
          pdusRetransmittedErrorThreshold  GET-REPLACE,
          nmsig-octetsRetransmittedErrorCounter  GET;

    NOTIFICATIONS
            ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
              attributeValueChange,
            ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
              communicationAlarm;

REGISTERED AS  {nmsig-package 11};

transportConnectionRetransmission-behaviour  BEHAVIOUR

  DEFINED AS

      This package reflects the retransmitting capability of the underlying transport protocol resource.

      Attributes that are subject to the attributeValueChange notification are:
      pdusRetransmittedErrorThreshold.
   ;
```

A.4.9 NMSIG Transport Connection Profile

A.4.9.1 NMSIG Transport Connection Profile Definition

```
nmsig-transportConnectionProfile   MANAGED OBJECT CLASS
  DERIVED FROM  ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] top;
    CHARACTERIZED BY nmsig-transportConnectionProfile-Package;
REGISTERED AS {nmsig-objectClass 9};
```

A.4.9.2 NMSIG Transport Connection Profile Package

```
nmsig-transportConnectionProfile-Package   PACKAGE
   BEHAVIOUR DEFINITIONS  transportConnectionProfile-behaviour;
     ATTRIBUTES   nmsig-transportConnectionProfileId  GET,
            nmsig-inactivityTimeout  GET-REPLACE,
          nmsig-maxTPDuSize  GET-REPLACE;
```

```
    NOTIFICATIONS
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
               objectCreation,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
               objectDeletion,
                ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :]
               attributeValueChange;
REGISTERED AS  {nmsig-package 12};
```

A.4.9.3 NMSIG Transport Connection Profile Behaviour

transportConnectionProfile-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 NMSIG Transport Connection managed object class when they are created. There can be only one instance of the NMSIG Transport Connection Profile managed object class for each instance of the NMSIG CO Transport Protocol Layer Entity managed object class.

The additionalCreateInfo field of the ObjectCreation notification shall contain all the attributes of the created transport connection profile instance.

The additionalDeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted transport connection profile instance.

Attributes that are subject to the AttributeValueChange notification are : nmsig-inactivityTimeout, nmsig-maxTPDuSize.

;

A.4.10 NMSIG Transport Connection Retransmission Profile

A.4.10.1 NMSIG Transport Connection Retransmission Profile Definition

```
nmsig-transportConnectionRetransmissionProfile   MANAGED OBJECT CLASS

  DERIVED FROM   nmsig-transportConnectionProfile;
     CHARACTERIZED BY  nmsig-transportConnectionProfile-Package;

REGISTERED AS  {nmsig-objectClass 10};
```

A.4.10.2 NMSIG Transport Connection Retransmission Profile Package

nmsig-transportConnectionRetransmissionProfile-Package PACKAGE

BEHAVIOUR DEFINITIONS transportConnectionProfile-behaviour;
ATTRIBUTES nmsig-maxRetransmissions GET-REPLACE,
nmsig-retransmissionTimerInitialValue GET-REPLACE;

REGISTERED AS {nmsig-package 13};

A.4.10.3 NMSIG Transport Connection Retransmission Profile Behaviour

transportConnectionRetransmissionProfile-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 NMSIG Transport Connection managed object class that support retransmission, when they are created. There can be only one instance of the NMSIG Transport Connection Retransmission Profile managed object class for each instance of the NMSIG CO Transport Protocol Layer Entity managed object class.

Attributes, additional to those inherited from the transport connection profile managed object class, that are subject to the AttributeValueChange notification are : nmsig-maxRetransmissions, nmsig-retransmissionTimerInitialValue.

;

A.5 NAME BINDINGS

This section provides definitions of NAME BINDINGS for the managed object classes defined by the NMSIG.

'Root' is a fictitious object class that represents the root of the containment tree. A name binding with 'root' as the superior object class means that the object class specified as the subordinate object class is effectively the top of the containment subtree within the context of the management entity that supports this name binding.

A.5.1 NMSIG Agent Name Bindings

```
agent-root   NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-agent;
    NAMED BY
  SUPERIOR OBJECT CLASS   root;
  WITH ATTRIBUTE nmsig-agentId;
REGISTERED AS  {nmsig-nameBindings 1};
```

A.5.2 NMSIG Computer System Name Bindings

```
computerSystem-network  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-computerSystem;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-network;
  WITH ATTRIBUTE nmsig-systemId;
REGISTERED AS  {nmsig-nameBindings 2};

computerSystem-computerSystem  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-computerSystem;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-computerSystem;
  WITH ATTRIBUTE nmsig-systemId;
REGISTERED AS  {nmsig-nameBindings 3};

computerSystem-root  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-computerSystem;
    NAMED BY
    SUPERIOR OBJECT CLASS root;
  WITH ATTRIBUTE nmsig-systemId;
REGISTERED AS  {nmsig-nameBindings 4};
```

A.5.3 NMSIG CO Transport Protocol Layer Entity Name Bindings

```
coTransportProtocolLayerEntity-computerSystem  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-coTransportProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-computerSystem;
  WITH ATTRIBUTE nmsig-coTransportEntityId;
REGISTERED AS  {nmsig-nameBindings 5};

coTransportProtocolLayerEntity-system  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-coTransportProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS
      ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] system;
  WITH ATTRIBUTE nmsig-coTransportEntityId;
REGISTERED AS  {nmsig-nameBindings 6};

coTransportProtocolLayerEntity-equipment  NAME BINDING
  SUBORDINATE OBJECT CLASS  nmsig-coTransportProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-equipment;
  WITH ATTRIBUTE nmsig-coTransportEntityId;
REGISTERED AS  {nmsig-nameBindings 7};
```

A.5.4 NMSIG CL Network Protocol Layer Entity Name Bindings

```
clNetworkProtocolLayerEntity-computerSystem  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-clNetworkProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-computerSystem;
  WITH ATTRIBUTE nmsig-clNetworkEntityId;
REGISTERED AS  {nmsig-nameBindings 8};

clNetworkProtocolLayerEntity-system  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-clNetworkProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS
      ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] system;
  WITH ATTRIBUTE nmsig-clNetworkEntityId;
REGISTERED AS  {nmsig-nameBindings 9};

clNetworkProtocolLayerEntity-equipment  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-clNetworkProtocolLayerEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-equipment;
  WITH ATTRIBUTE nmsig-clNetworkEntityId;
```

```
REGISTERED AS  {nmsig-nameBindings 10};
```

A.5.5 NMSIG Equipment Name Bindings

```
equipment-equipment  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-equipment;
     NAMED BY
     SUPERIOR OBJECT CLASS nmsig-equipment;
  WITH ATTRIBUTE nmsig-equipmentId;
REGISTERED AS  {nmsig-nameBindings 11};

equipment-network  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-equipment;
     NAMED BY
     SUPERIOR OBJECT CLASS nmsig-network;
  WITH ATTRIBUTE nmsig-equipmentId;
REGISTERED AS  {nmsig-nameBindings 12};

equipment-root  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-equipment;
     NAMED BY
     SUPERIOR OBJECT CLASS root;
  WITH ATTRIBUTE nmsig-equipmentId;
REGISTERED AS  {nmsig-nameBindings 13};
```

A.5.6 NMSIG Network Name Bindings

```
network-network  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-network;
     NAMED BY
     SUPERIOR OBJECT CLASS nmsig-network;
  WITH ATTRIBUTE nmsig-networkId;
REGISTERED AS  {nmsig-nameBindings 14};

network-root  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-network;
     NAMED BY
     SUPERIOR OBJECT CLASS root;
  WITH ATTRIBUTE nmsig-networkId;
REGISTERED AS  {nmsig-nameBindings 15};
```

A.5.7 NMSIG Processing Entity Name Bindings

```
processingEntity-computerSystem  NAME BINDING
```

```
  SUBORDINATE OBJECT CLASS nmsig-processingEntity;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-computerSystem;
  WITH ATTRIBUTE nmsig-equipmentId;
  DELETE  deletes-contained-objects;
REGISTERED AS  {nmsig-nameBindings 16};
```

A.5.8 NMSIG Transport Connection Name Bindings

```
transportConnection-coTransportProtocolLayerEntity  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-transportConnection;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-coTransportProtocolLayerEntity;
  WITH ATTRIBUTE nmsig-transportConnectionId;
  BEHAVIOUR  transportConnection-nb-behaviour  DEFINED AS
        The expected real effect of the DELETE operation when applied to an instance of the NMSIG
        transport connection managed object class is that the underlying transport connection resource
        is aborted.
    ;
  DELETE  deletes contained objects;
REGISTERED AS  {nmsig-nameBindings 17};
```

A.5.9 NMSIG Transport Connection Profile Name Bindings

```
transportConnectionProfile-coTransportProtocolLayerEntity  NAME BINDING
  SUBORDINATE OBJECT CLASS nmsig-transportConnectionProfile;
    NAMED BY
    SUPERIOR OBJECT CLASS nmsig-coTransportProtocolLayerEntity;
  WITH ATTRIBUTE nmsig-transportConnectionProfileId;
REGISTERED AS  {nmsig-nameBindings 18};
```

A.5.10 NMSIG Transport Connection Retransmission Profile Name Bindings

```
transportConnectionRetransmissionProfile-coTransportProtocolLayerEntity NAME BINDING
 SUBORDINATE OBJECT CLASS nmsig-transportConnectionRetransmissionProfile;
 NAMED BY
 SUPERIOR OBJECT CLASS nmsig-coTransportProtocolLayerEntity;
  WITH ATTRIBUTE nmsig-transportConnectionProfileId;
REGISTERED AS  {nmsig-nameBindings 19};
```

A.6 ATTRIBUTES

This section provides definitions of attributes contained in the managed object classes specified in this document.

All attribute syntaxes have been defined with external type references. External type references take the form:

ModuleName.NamedType

where the ModuleName refers to the name of an ASN.1 module and NamedType referes to a defined type in that module. The ASN.1 module referenced in this section is the NMSIG-SYNTAX-1 module found in Section A.10 of this Annex.

A.6.1 NMSIG Agent Id

```
nmsig-agentId  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX NMSIG-SYNTAX-1.Id   ;
    BEHAVIOUR  agentId-behaviour;
REGISTERED AS     {nmsig-attribute 1} ;

agentId-behaviour  BEHAVIOUR
 DEFINED AS
     This is the distinguishing attribute for the managed object class NMSIG Agent.
 ;
```

A.6.2 NMSIG Cause

```
nmsig-cause   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.Cause;
    MATCHES FOR  Equality;
    BEHAVIOUR  cause-behaviour;

   REGISTERED AS     {nmsig-attribute 2} ;

 cause-behaviour   BEHAVIOUR
DEFINED AS
     This attribute specifies the reason why a transport connection was deleted.  It is included in the
     additionalDeleteInfo field of the objectDeletion notification.
 ;
```

A.6.3 NMSIG Checksum TPDUs Discarded Counter

```
nmsig-checksumTPDUsDiscardedCounter   ATTRIBUTE
        DERIVED FROM
        ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
       BEHAVIOUR  checksumTPDUsDiscardedCounter-behaviour;

   REGISTERED AS     {nmsig-attribute 3} ;

 checksumTPDUsDiscardedCounter-behaviour   BEHAVIOUR
DEFINED AS
     This attribute specifies the number of TPDUs discarded due to a bad checksum.
 ;
```

A.6.4 NMSIG CO Transport Protocol Layer Entity Id

```
nmsig-coTransportProtocolLayerEntityId  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.Id   ;
    MATCHES FOR  Equality;
        BEHAVIOUR  coTransportProtocolLayerEntityId-behaviour;

REGISTERED AS     {nmsig-attribute 4} ;

 coTransportProtocolLayerEntityID-behaviour  BEHAVIOUR
DEFINED AS
     This is the distinguishing attribute for the managed object class connection oriented transport
     protocol layer entity.;
```

A.6.5 NMSIG Connectionless Network Protocol Layer Entity Id

```
nmsig-clNetworkProtocolLayerEntityId  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.Id ;
    MATCHES FOR  Equality ;
        BEHAVIOUR  clNetworkProtocolLayerEntityId-behaviour;

REGISTERED AS      {nmsig-attribute 5};

clNetworkProtocolLayerEntityId-behaviour   BEHAVIOUR
DEFINED AS
        This attribute is the distinguishing attribute for the managed object class
        clNetworkProtocolLayerEntity.;
```

A.6.6 NMSIG Contact Names

```
nmsig-contactNames   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.AnyName;
```

```
    MATCHES FOR  Set Comparison, Set Intersection;
          BEHAVIOUR  contactNames-behaviour   ;

REGISTERED AS     {nmsig-attribute 6};

contactNames-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies name(s) of one or more contacts.
        ;
```

A.6.7 NMSIG CPU Type

```
 nmsig-cPU-Type   ATTRIBUTE
      WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.CPU-Type;
  MATCHES FOR  Equality    ;
        BEHAVIOUR  cPU-Type-behaviour;

 REGISTERED AS   {nmsig-attribute 7};

   cPU-Type-behaviour  BEHAVIOUR
DEFINED AS
        This attribute specifies the type of the Central Processor Unit in a processing entity.;
```

A.6.8 NMSIG Entity Up Time

```
 nmsig-entityUpTime    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.EntityUpTime;
    MATCHES FOR   Equality, Ordering ;
           BEHAVIOUR  entityUpTime-behaviour;

REGISTERED AS     {nmsig-attribute 8} ;

   entityUpTime-behaviour  BEHAVIOUR
DEFINED AS
        This attribute specifies the time interval ( in seconds ) that has elapsed since the time that the
        value of the entity's operational state changed from 'disabled' to some other value, or since the
        time that the entity was created into a non disabled state.
        ;
```

A.6.9 NMSIG Equipment Id

```
nmsig-equipmentId   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Id;
    MATCHES FOR  Equality ;
           BEHAVIOUR  equipmentId-behaviour;
```

REGISTERED AS {nmsig-attribute 9} ;

equipmentId-behaviour BEHAVIOUR
DEFINED AS
This is the distinguishing attribute of the NMSIG equipment managed object class.;

A.6.10 NMSIG Equipment Purpose

nmsig-equipmentPurpose ATTRIBUTE
WITH ATTRIBUTE SYNTAX NMSIG-SYNTAX-1.EquipmentPurpose;
MATCHES FOR Equality ;
BEHAVIOUR equipmentPurpose-behaviour;

REGISTERED AS {nmsig-attribute 10};

equipmentPurpose-behaviour BEHAVIOUR
DEFINED AS
This attribute specifies what the equipment is used for (e.g. switching, processing, etc.).
;

A.6.11 NMSIG Inactivity Timeout

nmsig-inactivityTimeout ATTRIBUTE
WITH ATTRIBUTE SYNTAX NMSIG-SYNTAX-1.InactivityTimeout;
MATCHES FOR Equality, Ordering ;
BEHAVIOUR inactivityTimeout-behaviour;

REGISTERED AS {nmsig-attribute 11};

inactivityTimeout-behaviour BEHAVIOUR
DEFINED AS
This attribute specifies the maximum amount of time (in 1/100ths of a second) that the transport connection can remain up 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.6.12 NMSIG Local Network Address

nmsig-localNetworkAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX NMSIG-SYNTAX-1.Address;
MATCHES FOR Equality;
BEHAVIOUR localNetworkAddress-behaviour;

REGISTERED AS {nmsig-attribute 12};

localNetworkAddress-behaviour BEHAVIOUR
DEFINED AS
This attribute identifies the local network address of the transport connection (e.g. the local IP address for TCP or the local NSAP for OSI TP).;

A.6.13 NMSIG Local Network Addresses

```
nmsig-localNetworkAddresses       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.LocalNetworkAddresses;
    MATCHES FOR  Set Comparison, Set Intersection;
            BEHAVIOUR  localNetworkAddresses-behaviour;

REGISTERED AS      {nmsig-attribute 13};
```

localNetworkAddresses-behaviour BEHAVIOUR
DEFINED AS
This attribute specifies a set of local network addresses supported by a network protocol layer entity.;

A.6.14 NMSIG Local Transport Addresses

```
nmsig-localTransportAddresses    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.TransportAddresses;
    MATCHES FOR  Set Comparison, Set Intersection;
            BEHAVIOUR  localTransportAddresses-behaviour;

REGISTERED AS  {nmsig-attribute 14};
```

localTransportAddresses-behaviour BEHAVIOUR
DEFINED AS
This attribute specifies the set of transport addresses 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.;

A.6.15 NMSIG Local Transport Connection Endpoint

```
nmsig-localTransportConnectionEndpoint      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Address;
    MATCHES FOR  Equality ;
            BEHAVIOUR  localTransportConnectionEndpoint-behaviour;

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

localTransportConnectionEndpoint-behaviour BEHAVIOUR

```
DEFINED AS
        This attribute identifies the local transport connection endpoint (e.g. it represents the source port
        for TCP or the local t-selector for OSI TP).;
```

A.6.16 NMSIG Location Name

```
nmsig-locationName   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX    NMSIG-SYNTAX-1.AnyName;
    MATCHES FOR  Equality ;
           BEHAVIOUR  locationName-behaviour;

REGISTERED AS     {nmsig-attribute 16};

    locationName-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the name of a location (e.g. Hilo Hawaii USA);
```

A.6.17 NMSIG Manufacturer Info

```
nmsig-manufacturerInfo  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.ManufacturerInfo;
    MATCHES FOR  Equality ;
           BEHAVIOUR  manufacturerInfo-behaviour;

REGISTERED AS      {nmsig-attribute 17};

    manufacturerInfo-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies information about the manufacturer of the product that has implemented
        the underlying resource.;
```

A.6.18 NMSIG Max Connections

```
nmsig-maxConnections          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.MaxNumber;
    MATCHES FOR   Equality, Ordering;
           BEHAVIOUR  maxConnections-behaviour;

REGISTERED AS     {nmsig-attribute 18} ;

    maxConnections-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the maximum number of simultaneously open transport connections
        allowed by the transport protocol layer entity.;
```

A.6.19 NMSIG Max PDU Size

```
nmsig-maxPDUSize          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Length;
    MATCHES FOR   Equality, Ordering;
          BEHAVIOUR  maxPDUSize-behaviour;

REGISTERED AS     {nmsig-attribute 19} ;

    maxPDUSize-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the maximum length of a PDU that can be supported by the underlying
        resource
  ;
```

A.6.20 NMSIG Max Retransmissions

```
nmsig-maxRetransmissions  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.MaxNumber;
    MATCHES FOR   Equality, Ordering ;
          BEHAVIOUR  maxRetransmissions-behaviour;

REGISTERED AS     {nmsig-attribute 20} ;

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

A.6.21 NMSIG Memory Size

```
     nmsig-memorySize   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Amount;
    MATCHES FOR   Equality, Ordering ;
          BEHAVIOUR  memorySize-behaviour;

 REGISTERED AS   {nmsig-attribute 21};

    memorySize-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the amount of random access memory ( in kilobytes ) that is owned by
        a processing entity. ( 1 Kilobyte = 1024 bytes );
```

A.6.22 NMSIG Network Entity Type

```
    nmsig-networkEntityType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.NetworkEntityType;
  MATCHES FOR  Equality         ;
        BEHAVIOUR  networkEntityType-behaviour;

 REGISTERED AS      {nmsig-attribute 22};

    networkEntityType-behaviour   BEHAVIOUR
 DEFINED AS
        This attribute specifies the type of the network protocol layer  entity.;
```

A.6.23 NMSIG Network Id

```
nmsig-networkId   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Id;
    MATCHES FOR  Equality ;
           BEHAVIOUR  networkId-behaviour;

REGISTERED AS     {nmsig-attribute 23} ;

    networkId-behaviour   BEHAVIOUR
DEFINED AS
        This is the distinguishing attribute of the NMSIG network managed object class.;
```

A.6.24 NMSIG Network Purpose

```
nmsig-networkPurpose   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.NetworkPurpose;
    MATCHES FOR  Equality ;
           BEHAVIOUR  networkPurpose-behaviour;

REGISTERED AS     {nmsig-attribute 24} ;

    networkPurpose-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies what the network is used for ( e.g. manufacturing control,  airline
        reservation, etc. )
      ;
```

A.6.25 NMSIG NPDU Time To Live

```
nmsig-nPDUTimeToLive          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.MaxNumber;
```

```
    MATCHES FOR   Equality, Ordering;
          BEHAVIOUR  nPDUTimeToLive-behaviour;

REGISTERED AS      {nmsig-attribute 25};

   nPDUTimeToLive-behaviour   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.6.26 NMSIG Octets Retransmitted Error Counter

```
nmsig-octetsRetransmittedErrorCounter   ATTRIBUTE
         DERIVED FROM ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
          BEHAVIOUR  octetsRetransmittedErrorCounter-behaviour;

REGISTERED AS     {nmsig-attribute 26} ;

   octetsRetransmittedErrorCounter-behaviour   BEHAVIOUR
DEFINED AS
```

This attribute specifies the total number of octets that were retransmitted.;

A.6.27 NMSIG OS Info

```
 nmsig-osInfo   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.OsInfo;
    MATCHES FOR Set Comparison, Set Intersection;
          BEHAVIOUR  osInfo-behaviour;

 REGISTERED AS   {nmsig-attribute 27};

   osInfo-behaviour   BEHAVIOUR
DEFINED AS
```

This attribute specifies the names and releases of operating systems supported by the processing entity;

A.6.28 NMSIG Open Connections

```
nmsig-openConnections          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Number;
    MATCHES FOR   Equality, Ordering ;
          BEHAVIOUR  openConnections-behaviour;
```

```
REGISTERED AS     {nmsig-attribute 28} ;

    openConnections-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the number of currently established transport connections.;
```

A.6.29 NMSIG PDUs Discarded Counter

```
nmsig-PDUsDiscardedCounter          ATTRIBUTE
          DERIVED FROM ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
           BEHAVIOUR  pDUsDiscardedCounter-behaviour;

REGISTERED AS     {nmsig-attribute 29} ;

    pDUsDiscardedCounter-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the number of PDUs that were discarded by a network protocol layer
        entity.;
```

A.6.30 NMSIG PDUs Forwarded Counter

```
nmsig-PDUsForwardedCounter          ATTRIBUTE
          DERIVED FROM ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
           BEHAVIOUR  pDUsForwardedCounter-behaviour;

REGISTERED AS     {nmsig-attribute 30} ;

    pDUsForwardedCounter-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the number of PDUs forwarded by a network protocol layer entity.;
```

A.6.31 NMSIG PDUs Reassemble Fail Counter

```
nmsig-PDUsReasmblFailCounter          ATTRIBUTE
          DERIVED FROM ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
           BEHAVIOUR  pDUsReasmblFailCounter-behaviour;

REGISTERED AS     {nmsig-attribute 31} ;

    pDUsReasmblFailCounter-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the number of PDUs that could not be reassembled successfully by a
        network protocol layer entity.;
```

A.6.32 NMSIG PDUs Reassembled OK Counter

```
nmsig-PDUsReasmbldOKCounter          ATTRIBUTE
        DERIVED FROM ["Recommendation X.721"|"ISO/IEC DIS 10165-2" :] counter;
         BEHAVIOUR  pDUsReasmbldOKCounter-behaviour;

REGISTERED AS     {nmsig-attribute 32} ;

   pDUsReasmbldOKCounter-behaviour   BEHAVIOUR
DEFINED AS
      This attribute specifies the number of PDUs that were reassembled successfully by a network
      protocol layer entity.
      ;
```

A.6.33 NMSIG Peripheral Names

```
nmsig-peripheralNames  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.PeripheralNames;
   MATCHES FOR Set Comparison, Set Intersection  ;
         BEHAVIOUR  peripheralNames-behaviour;

REGISTERED AS     {nmsig-attribute 33};

   peripheralNames-behaviour   BEHAVIOUR
DEFINED AS
      This attribute specifies the names of auxiliary devices.;
```

A.6.34 NMSIG Product Label

```
nmsig-productLabel  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.ProductLabel;
    MATCHES FOR  Equality ;
          BEHAVIOUR  productLabel-behaviour;

REGISTERED AS      {nmsig-attribute 34};

   productLabel-behaviour   BEHAVIOUR
DEFINED AS
      This attribute specifies the product label of the product that has implemented the underlying
      resource.;
```

A.6.35 NMSIG Release

```
nmsig-release  ATTRIBUTE
```

```
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Release;
    MATCHES FOR  Equality ;
          BEHAVIOUR  release-behaviour;

REGISTERED AS      {nmsig-attribute 35};

   release-behaviour   BEHAVIOUR
DEFINED AS
      This attribute specifies the release number of the product that has implemented the underlying
      resource.;
```

A.6.36 NMSIG Remote Network Address

```
nmsig-remoteNetworkAddress       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Address;
    MATCHES FOR  Equality ;
          BEHAVIOUR  remoteNetworkAddress-behaviour;

REGISTERED AS      {nmsig-attribute 36};

   remoteNetworkAddress-behaviour   BEHAVIOUR
DEFINED AS
      This attribute identifies the remote network address of the transport connection (e.g. it
      represents the remote IP address for TCP or the remote NSAP for OSI TP).;
```

A.6.37 NMSIG Remote Transport Connection Endpoint

```
nmsig-remoteTransportConnectionEndpoint       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Address;
    MATCHES FOR  Equality ;
          BEHAVIOUR  remoteTransportConnectionEndpoint-behaviour;

REGISTERED AS      {nmsig-attribute 37};

   remoteTransportConnectionEndpoint-behaviour   BEHAVIOUR
DEFINED AS
      This attribute identifies the remote transport connection endpoint ( It represents the destination
      port for TCP or the remote t-selector for OSI TP).;
```

A.6.38 NMSIG Retransmission Timer Initial Value

```
nmsig-retransmissionTimerInitialValue  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Number;
    MATCHES FOR   Equality, Ordering;
```

```
            BEHAVIOUR  retransmissionTimerInitialValue-behaviour;

REGISTERED AS     {nmsig-attribute 38} ;

   retransmissionTimerInitialValue-behaviour   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.6.39 NMSIG Serial Number

```
nmsig-serialNumber  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.SerialNumber;
    MATCHES FOR  Equality ;
            BEHAVIOUR  serialNumber-behaviour;

REGISTERED AS      {nmsig-attribute 39};

   serialNumber-behaviour   BEHAVIOUR
DEFINED AS
       This attribute specifies the serial number of the product that has implemented the underlying
       resource.;
```

A.6.40 NMSIG System Id

```
nmsig-systemId  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Id;
    MATCHES FOR  Equality ;
            BEHAVIOUR  systemId-behaviour;

REGISTERED AS      {nmsig-attribute 40};

   systemId-behaviour   BEHAVIOUR
DEFINED AS
       This is the distinguishing attribute of the NMSIG computer system managed object class.;
```

A.6.41 NMSIG System Time

```
nmsig-systemTime  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GeneralizedTime;
    MATCHES FOR   Equality, Ordering;
            BEHAVIOUR  systemTime-behaviour;

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

```
    systemTime-behaviour   BEHAVIOUR
  DEFINED AS
        This attribute specifies the current time clocked at the computer system.;
```

A.6.42 NMSIG Transport Connection Id

```
  nmsig-transportConnectionId  ATTRIBUTE
     WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Id;
     MATCHES FOR  Equality ;
           BEHAVIOUR  transportConnectionId-behaviour;

  REGISTERED AS     {nmsig-attribute 42};

    transportConnectionId-behaviour   BEHAVIOUR
  DEFINED AS
        This attribute is the distinguishing attribute for the managed object class transportConnection.;
```

A.6.43 NMSIG Transport Connection Profile Id

```
  nmsig-transportConnectionProfileId  ATTRIBUTE
     WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Id;
     MATCHES FOR  Equality ;
           BEHAVIOUR  transportConnectionProfileId-behaviour;

  REGISTERED AS     {nmsig-attribute 43};

    transportConnectionProfileId-behaviour   BEHAVIOUR
  DEFINED AS
        This attribute is the distinguishing attribute for the managed object class nmsig-
        transportConnectionProfile.;
```

A.6.44 NMSIG Transport Connection Reference

```
  nmsig-transportConnectionReference       ATTRIBUTE
     WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Address;
     MATCHES FOR  Equality ;
           BEHAVIOUR  transportConnectionReference-behaviour;

  REGISTERED AS      {nmsig-attribute 44};

    transportConnectionReference-behaviour   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.6.45 NMSIG Transport Entity Type

```
    nmsig-transportEntityType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.TransportEntityType;
  MATCHES FOR  Equality ;
        BEHAVIOUR  transportEntityType-behaviour;

 REGISTERED AS      {nmsig-attribute 45};

   transportEntityType-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the type of the transport protocol layer entity.;
```

A.6.46 NMSIG User Friendly Label

```
nmsig-userFriendlyLabel  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.Label;
    MATCHES FOR  Equality;
           BEHAVIOUR  userFriendlyLabel-behaviour;

REGISTERED AS      {nmsig-attribute 46};

   userFriendlyLabel-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies a user friendly name.;
```

A.6.47 NMSIG Vendor Name

```
nmsig-vendorName   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   NMSIG-SYNTAX-1.AnyName;
    MATCHES FOR  Equality ;
           BEHAVIOUR  vendorName-behaviour;

REGISTERED AS     {nmsig-attribute 47};

   vendorName-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the name of a vendor.;
```

A.6.48 NMSIG Wrapped Counter

```
nmsig-wrappedCounter  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  NMSIG-SYNTAX-1.WrappedCounter;
    MATCHES FOR  Equality;
           BEHAVIOUR  wrappedCounter-behaviour;

REGISTERED AS      {nmsig-attribute 48};

    wrappedCounter-behaviour   BEHAVIOUR
DEFINED AS
        This attribute specifies the attribute Id and value of the counter attribute that wrapped.;
```

A.7 ATTRIBUTE GROUPS

This section provides definitions of attribute groups supported by managed object classes defined by the NMSIG.

A.8 PARAMETERS

This section provides definitions of parameters supported by managed object classes defined by the NMSIG.

A.9 ACTIONS

This section provides definitions of actions supported by managed object classes defined by the NMSIG.

A.10 NOTIFICATIONS

This section provides definitions of notifications emitted by managed object classes defined by the NMSIG.

A.10.1 NMSIG Counter Wrap

```
nmsig-counterWrap   NOTIFICATION
      BEHAVIOUR  counterWrap-behaviour;
      MODE  CONFIRMED AND UNCONFIRMED
      WITH INFORMATION SYNTAX   NMSIG-SYNTAX-1.WrapInfo
        AND ATTRIBUTE IDS  wrappedCounter  wrappedCounter;

REGISTERED AS  {notification};

counterWrap-behaviour   BEHAVIOUR
          DEFINED AS
               This notification indicates that a counter has wrapped.;
```

A.11 REFERENCES

This section lists the names of documents that were referenced in the earlier sections.

[DMI]

A.12 SYNTAX DEFINITIONS

This section contains an ASN.1 module that defines attribute and notification syntaxes referenced by the attribute and notification templates in Sections A.6 and A.8 respectively.

```
NMSIG-SYNTAX-1  {nmsig mil(2) nmsig-modules(0) syntax-1(0)}
        DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS Attribute FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3) }

       Address ::= OCTET STRING

       Amount ::= INTEGER

    AnyName ::=  SET OF ( CHOICE {  dn   DistinguishedName,
                          ps   PrintableString } )

    Cause  ::=  SEQUENCE {       INTEGER ( unknown (0),
                               user    (1),
                               provider (2) ),
                         INTEGER ( unknown (0),
                               excessiveIdle (1),
                               excessiveRtx  (2) )
                          }

    CPU-Type ::= PrintableString;

    EntityUpTime ::= INTEGER;

       EquipmentPurpose  ::= PrintableString

    Id ::= PrintableString

       InactivityTimeout ::= INTEGER

    Label  ::= PrintableString

       Length ::= INTEGER

    LocalNetworkAddresses  ::=  SET OF OCTET STRING

       ManufacturerInfo ::= PrintableString
```

```
    MaxNumber ::= INTEGER


NetworkEntityType ::=   INTEGER { other(0),
                         oSI CLNP (1),
                         internet IP (2) } (0..256)

    NetworkPurpose ::= PrintableString

    Number ::=  INTEGER;

OsInfo  ::=  SET  OF ( CHOICE { osName  [0] DistingishedName,
                        osSpec  [1] PrintableString } )

PeripheralNames ::=  SET OF AnyName

    ProductLabel ::= PrintableString

    Release ::= PrintableString

    SerialNumber ::= PrintableString

TransportAddresses ::=  SET  OF  SEQUENCE {
transportConnectionEndpoint OCTET STRING,
                networkAddress  OCTET STRING  }

TransportEntityType ::=  INTEGER { other(0),
                          oSI TP (1),
                          tCP (2),
                          sNA (3) } (0..256)

WrapInfo  ::=   SEQUENCE {
            wrappedCounter  Attribute  -- attribute ID and value of counter attribute that wrapped
              }

WrappedCounter ::=  Attribute

END -- End of NMSIG-SYNTAX-1 module
```

Annex B (informative)

NMSIG Object Identifiers

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 identifers are assigned under the { iso identified-organization oiw nmsig } node, labelled "nmsig".

Table B.1: Object Identifers Assigned Under "nmsig" Node

Identifier	Value	Reference
phase1	1	B.2
mil1	2	B.3

B.2 Phase 1 Object Identifiers

Several of the base standards referenced by Phase I agreements are at Draft International Standard (DIS) level. These draft standards specify a number of information objects, each accompanied by a tentative object identifier. However, actual object identifiers are not assigned and registered by ISO until the base standard reaches International Standard (IS) level.

Implementations require globally unamibiguous object identifiers to interoperate using Phase I agreements. Since these object identifiers are not yet specified by ISO, it is necessary for the NMSIG to assign and register identifiers needed to support Phase I agreements. All such object identifiers are assigned under a single "phase1" node of the NMSIG object identifier tree.

Object identifiers under the phase1 node are assigned such that the phase1 node replaces the corresponding { joint-iso-ccitt ms } node tentatively specified in the Draft International Standards. That is:

```
joint-iso-ccitt ── ms ─────┬─ smo ──────┬─ application-context ──────┬─ manager
                           │            │                            ├─ agent
                           │            │                            └─ manager-agent
                           │            │
                           │            └─ negotationAbstractSyntax ── version1
                           │
                           ├─ function ─┬─ part1 ── functionalUnitPackage
                           │            ├─ part4 ── functionalUnitPackage
                           │            └─ part5 ── functionalUnitPackage
                           │
                           └─ smi ──────── part2 ─┬─ managedObjectClass
                                                  ├─ nameBinding
                                                  ├─ package
                                                  ├─ attribute
                                                  ├─ attributeGroup
                                                  ├─ standardSpecificExtension
                                                  └─ notification
```

becomes

```
nmsig ──────────── phase1 ─┬─ smo ──────┬─ application-context ──────┬─ manager
                           │            │                            ├─ agent
                           │            │                            └─ manager-agent
                           │            │
                           │            └─ negotationAbstractSyntax ── version1
                           │
                           ├─ function ─┬─ part1 ── functionalUnitPackage
                           │            ├─ part4 ── functionalUnitPackage
                           │            └─ part5 ── functionalUnitPackage
                           │
                           └─ smi ──────── part2 ─┬─ managedObjectClass
                                                  ├─ nameBinding
                                                  ├─ package
                                                  ├─ attribute
                                                  ├─ attributeGroup
                                                  ├─ standardSpecificExtension
                                                  └─ notification
```

Using this methodology, the following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 } node.

Table B.2: Object Identifers Assigned Under "phase1" Node

Identifier	Value	Reference
smo	0	[GDMO] 8.4
function	2	[GDMO] 8.4

Identifier	Value	Reference
smi	3	[GDMO] 8.4

These object identifiers are to be used when referencing ISO DIS-specified Phase I information objects. During progression from DIS to IS, it is possible that some information object definitions and/or tentatively-specified object identifiers will change. As a result, final IS documents will contain a new set of information objects, each assigned its own ISO object identifer. This shall be the case even if no changes are made to the object definition during progression from DIS to IS. Phase1 object identifiers shall correspond in perpetuity to the referenced DIS text upon which Phase 1 stable agreements are based. However, to encourage timely migration, the use of phase1 object identifiers shall be deprecated 2 years after corresponding International Standard Profiles (ISPs) become available.

B.2.1 SMO Object Identifiers

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smo } node.

Table B.3: Object Identifers Assigned Under "smo" Node

Identifier	Value	Reference
application-context	0	[GDMO] 8.4
negotiationAbstractSyntax	1	[GDMO] 8.4

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smo application-context } node.

Table B.4: Object Identifers Assigned Under "application-context" Node

Identifier	Value	Reference
manager	0	[SMO] A.2.3
agent	1	[SMO] A.3.3
manager-agent	2	[SMO] A.4.3

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smo negotationAbstractSyntax } node.

Table B.5: Object Identifers Assigned Under "negotiationAbstractSyntax" Node

Identifier	Value	Reference
version1	1	[SMO] A.5.4

B.2.2 Function Object Identifiers

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 function } node.

Table B.6: Object Identifers Assigned Under "function" Node

Identifier	Value	Reference
part1	1	[GDMO] 8.4
part4	4	[GDMO] 8.4
part5	5	[GDMO] 8.4

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 function part1 } node.

Table B.7: Object Identifers Assigned Under "part1" Node

Identifier	Value	Reference
functionalUnitPackage	1	[OMF] 10

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 function part4 } node.

Table B.8: Object Identifers Assigned Under "part4" Node

Identifier	Value	Reference
functionalUnitPackage	1	[ARF] 11.3

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 function part5 } node.

Table B.9: Object Identifers Assigned Under "part5" Node

Identifier	Value	Reference
functionalUnitPackage	1	[ERMF] 11.3

B.2.3 SMI Object Identifiers

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smi } node.

Table B.10: Object Identifers Assigned Under "smi" Node

Identifier	Value	Reference
part2	2	[GDMO] 8.4

The following object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smi part2 } node.

Table B.11: Object Identifers Assigned Under "part2" Node

Identifier	Value	Reference
managedObjectClass	3	[GDMO] 8.4
nameBinding	6	[GDMO] 8.4
package	4	[GDMO] 8.4
attribute	7	[GDMO] 8.4
attributeGroup	8	[GDMO] 8.4
standardSpecificExtension	0	[GDMO] 8.4
notification	10	[GDMO] 8.4

Object identifiers are assigned under the { iso identified-organization oiw nmsig phase1 smi part2 } node by the following ASN.1 productions.

```
smi2MObjectClass OBJECT IDENTIFER ::= -- supersedes [DMI] 13.1
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) managedObjectClass(3) }
```

```
smi2NameBinding OBJECT IDENTIFER ::= -- supersedes [DMI] 13.1
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) nameBinding(6) }

smi2Package OBJECT IDENTIFIER ::= -- supersedes [DMI] 13.1
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) package(4) }

smi2AttributeID OBJECT IDENTIFIER ::= -- supersedes [DMI] 13.2
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) attribute(7) }

smi2AttributeGroup OBJECT IDENTIFIER ::= -- supersedes [DMI] 13.2
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) attributeGroup(8) }

arfProbableCause OBJECT IDENTIFIER ::= -- supersedes [DMI] 13.2
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) standardSpecificExtension (0) arf (0) }

smi2Notification OBJECT IDENTIFIER ::= -- supersedes [DMI] 13.3
    { iso(1) identified-organization(3) oiw(14) nmsig(2)
    phase1(1) smi(3) part2(2) notification(10) }
```

These ASN.1 productions effectively register, under the nmsig phase1 smi part2 node, all information objects specified in [DMI]. For example:

```
attributeValueChange NOTIFICATION
REGISTERED AS { smi2Notification 1 }
```

as defined in clause 12.1 of [DMI] is assigned the object identifier:

```
{ iso(1) identified-organization(3) oiw(14) nmsig(2)
phase1(1) smi(3) part2(2) notification(10) attributeValueChange(1) }
```

The majority of these information objects support Phase 1 agreements and shall be used as specified throughout Chapter 18. However, a small number of information objects specified in [DMI] support functionality beyond the scope of Phase 1 agreements. Although such objects are registered under the nmsig phase1 smi part2 node, their use is outside the scope of these agreements.

B.3 MIL Object Identifiers

These are the object identifiers referenced in Annex A –– Management Information Library.

All definitions that are registered under the mil1 node are based on the DIS version of the [GDMO] standard.

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

```
nmsig   OBJECT IDENTIFIER  ::=
            { iso identified-organizations(3) oiw(14) 2 }

mil1               OBJECT IDENTIFIER  ::=    { nmsig 2 }

nmsig-modules       OBJECT IDENTIFIER  ::=    { mil1 0 }

nmsig-objectClass   OBJECT IDENTIFIER  ::=    { mil1 1 }

nmsig-package       OBJECT IDENTIFIER  ::=    { mil1 2 }

nmsig-nameBindings  OBJECT IDENTIFIER  ::=    { mil1 3 }

nmsig-attribute     OBJECT IDENTIFIER  ::=    { mil1 4 }

nmsig-attributeGroups  OBJECT IDENTIFIER ::=  { mil1 5 }

nmsig-parameter     OBJECT IDENTIFIER  ::=    { mil1 6 }

nmsig-action        OBJECT IDENTIFIER  ::=    { mil1 7 }

nmsig-notification  OBJECT IDENTIFIER  ::=    { mil1 8 }
```

B.3.1 Object Class Object Identifiers

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

Table B.12: Object Identifiers Assigned Under "nmsig-objectClass" Node

Identifier	Value	Reference
nmsig-agent	1	A.4.1.1
nmsig-computerSystem	2	A.4.2.1
nmsig-coTransportProtocolLayerEntity	3	A.4.3.1
nmsig-clNetworkProtocolLayerEntity	4	A.4.4.1
nmsig-equipment	5	A.4.5.1
nmsig-network	6	A.4.6.1
nmsig-processingEntity	7	A.4.7.1
nmsig-transportConnection	8	A.4.8.1

Identifier	Value	Reference
nmsig-transportConnectionProfile	9	A.4.9.1
nmsig-transportConnectionRetransmissionProfile	10	A.4.10.1

B.3.2 Package Object Identifiers

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

Table B.13: Object Identifiers Assigned Under "nmsig-package" Node

Identifier	Value	Reference
nmsig-agent-Package	1	A.4.1.2
nmsig-computerSystem-Package	2	A.4.2.2
nmsig-coTransportProtocolLayerEntity-Package	3	A.4.3.2
nmsig-productInfo-Package	4	A.4.3.4
nmsig-clNetworkProtocolLayerEntity-Package	5	A.4.4.2
nmsig-clNetworkProtocolLayerEntityRedirection-Package	6	A.4.4.4
nmsig-equipment-Package	7	A.4.5.2
nmsig-network-Package	8	A.4.6.2
nmsig-processingEntity-Package	9	A.4.7.2
nmsig-transportConnection-Package	10	A.4.8.2
nmsig-transportConnectionRetransmission-Package	11	A.4.8.4
nmsig-transportConnectionProfile-Package	12	A.4.9.2
nmsig-transportConnectionRetransmissionProfile-Package	13	A.4.10.2

B.3.3 Name Bindings Object Identifiers

The following object identifiers are assigned under the { nmsig–nameBindings } node:

Table B.14: Object Identifiers Assigned Under "nmsig-nameBindings" Node

Identifier	Value	Reference
agent-root	1	A.5.1
computerSystem-network	2	A.5.2
computerSystem-computerSystem	3	A.5.2
computerSystem-root	4	A.5.2
coTransportProtocolLayerEntity-computerSystem	5	A.5.3
coTransportProtocolLayerEntity-system	6	A.5.3
coTransportProtocolLayerEntity-equipment	7	A.5.3
clNetworkProtocolLayerEntity-computerSystem	8	A.5.4
clNetworkProtocolLayerEntity-system	9	A.5.4
clNetworkProtocolLayerEntity-equipment	10	A.5.4
equipment-equipment	11	A.5.5
equipment-network	12	A.5.5
equipment-root	13	A.5.5
network-network	14	A.5.6
network-root	15	A.5.6
processingEntity-computerSystem	16	A.5.7
transportConnection-coTransportProtocolLayerEntity	17	A.5.8
transportConnectionProfile-coTransportProtocolLayerEntity	18	A.5.9
transportConnectionRetransmissionProfile-coTransportProtocolLayerEntity	19	A.5.10

B.3.4 Attribute Object Identifiers

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

Table B.15: Object Identifiers Assigned Under "nmsig-attribute" Node

Identifier	Value	Reference
nmsig-agentId	1	A.6.1
nmsig-cause	2	A.6.2
nmsig-checksumTPDUsDiscardedCounter	3	A.6.3
nmsig-coTransportProtocolLayerEntityId	4	A.6.4
nmsig-clNetworkProtocolLayerEntityId	5	A.6.5
nmsig-contactNames	6	A.6.6
nmsig-cPUType	7	A.6.7
nmsig-entityUpTime	8	A.6.8
nmsig-equipmentId	9	A.6.9
nmsig-equipmentPurpose	10	A.6.10
nmsig-inactivityTimeout	11	A.6.11
nmsig-localNetworkAddress	12	A.6.12
nmsig-localNetworkAddresses	13	A.6.13
nmsig-localTransportAddresses	14	A.6.14
nmsig-localTransportConnectionEndpoint	15	A.6.15
nmsig-locationName	16	A.6.16
nmsig-manufacturerInfo	17	A.6.17
nmsig-maxConnections	18	A.6.18
nmsig-maxPDUSize	19	A.6.19
nmsig-maxRetransmissions	20	A.6.20
nmsig-memorySize	21	A.6.21
nmsig-networkEntityType	22	A.6.22
nmsig-networkId	23	A.6.23
nmsig-networkPurpose	24	A.6.24
nmsig-nPDUTimeToLive	25	A.6.25

Identifier	Value	Reference
nmsig-octetsRetransmittedErrorCounter	26	A.6.26
nmsig-osInfo	27	A.6.27
nmsig-openConnections	28	A.6.28
nmsig-PDUsDiscardedErrorCounter	29	A.6.29
nmsig-PDUsForwardedCounter	30	A.6.30
nmsig-PDUsReasmblFailCounter	31	A.6.31
nmsig-PDUsReasmbldOKCounter	32	A.6.32
nmsig-peripheralNames	33	A.6.33
nmsig-productLabel	34	A.6.34
nmsig-release	35	A.6.35
nmsig-remoteNetworkAddress	36	A.6.36
nmsig-remoteTransportConnectionEndpoint	37	A.6.37
nmsig-retransmissionTimerInitialValue	38	A.6.38
nmsig-serialNumber	39	A.6.39
nmsig-systemId	40	A.6.40
nmsig-systemTime	41	A.6.41
nmsig-transportConnectionId	42	A.6.42
nmsig-transportConnectionProfileId	43	A.6.43
nmsig-transportConnectionReference	44	A.6.44
nmsig-transportEntityType	45	A.6.45
nmsig-userFriendlyLabel	46	A.6.46
nmsig-vendorName	47	A.6.47
nmsig-wrappedCounter	48	A.6.48

B.3.5 Notification Object Identifiers

The following object identifiers are assigned under the { nmsig–notification } node:

Table B.16: Object Identifiers Assigned Under "nmsig-notification" Node

Identifier	Value	Reference
nmsig-counterWrap	1	A.10.1