NISTIR 4448

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held September 14, 1990
National Institute of Standards and
Technology
Gaithersburg, MD 20899

Tim Boland, Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 4448

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held September 14, 1990
National Institute of Standards and
Technology
Gaithersburg, MD 20899

Tim Boland, Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

September 1990

Issued December 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Table of Contents

Table of Contents

1 GENERAL INFORMATION

1 PURPOSE OF THIS DOCUMENT

This document records working (not stable) implementation specification agreements of OSI protocols among the organizations participating in the NIST Workshop for implementors of OSI. This work is not currently considered advanced enough for use in product development or procurement reference. However, it is intended that this work be a basis for future stable agreements. It is possible that any material contained in this document may be declared stable in the future, and the material should be considered in this light. In the status sections of each chapter as appropriate, specific functionality may be flagged as being "likely" to become stable at the next workshop.

Only non-stable text is included in this document. Errata to Stable material, as well as new stable functionality, is presented as an aligned edition (in replacement page format) issued at the same time as this document.

As each protocol specification is completed (becomes technically stable), it is moved from this working document to the stable companion document as described below.

- o The companion document, "Stable Implementation Agreements for Open Systems Interconnection Protocols, Version 3 as of Septlember 1990" records mature agreements considered advanced enough for use in product development or procurement reference.

New text relating to any of the referenced subjects appears first in this working document. In general, new text must reside in this working document for at least one workshop period before being moved into the Stable Document.

Agreements text is either in this Working Document (not yet stable) or in the aligned Stable Document (has been declared stable). It is a goal that the same text not appear in the same position in both documents at once (except for section one).

The benefit of this document is that it gives the reader a glimpse of new functionality, for planning purposes. Together with the aligned, associated stable document, these two documents give the reader a complete picture of current OSI agreements in this forum.

An implementor should look at the aligned section in the Stable Document to get the true current status of stable material. In this Working Document, all references to the Stable Document are to V3 as of September 1990. Where appropriate, statements related to backward compatibility, interworking considerations, or agreement maintenance are given in this document. Architectural issues may also be considered as appropriate.

2 PURPOSE OF THE WORKSHOP

At the request of industry, the National Institute of Standards and Technology organized the NIST Workshop for Implementors of OSI to bring together future users and potential suppliers of OSI protocols. The Workshop accepts as input the specifications of emerging standards for protocols and produces as output agreements on the implementation and testing particulars of these protocols. This process is expected to expedite the development of OSI protocols and promote interoperability of independently manufactured data communications equipment.

3 WORKSHOP ORGANIZATION

See the aligned section of the Stable Implementation Agreements Document for information.

4 USE AND ENDORSEMENT BY OTHER ENTERPRISES

The Workshops are held for those organizations expressing an interest in implementing or procuring OSI protocols and open systems. However, there is no corporate commitment to implementations associated with Workshop participation.

The Agreements in this document were a basis for testing and product demonstrations in the Enterprise Networking Event in Baltimore, MD, June, 1988.

The agreements contained in earlier versions of this document were used for OSI demonstrations at the National Computer Conference in 1984 and at the AUTOFACT conference in 1985.

The agreements from several versions of this document have been adopted for use in implementations running on OSINET.

The MAP/TOP Steering Committee has endorsed these agreements and will "continue the use of the most current, applicable Implementors Workshop Agreements in all releases of the MAP and TOP specifications."

The COS Strategy Forum has "adopted a resolution stating that as a matter of policy COS should select as its sources of Implementation Agreements organizations or forums that are: (1) Broadly open, widely recognized OSI Workshops (NIST/OSI Workshops are first preference) ..."

The implementation specifications from the "Stable Implementation Agreements for Open System Interconnection Protocols" are referenced in Federal Information Processing Standard 146, "Government OSI Profile (GOSIP)."

5 RELATIONSHIP OF THE WORKSHOP TO THE NIST LABORATORIES

As resources permit, NIST, with voluntary assistance from industry, develops formal protocol specifications, reference implementations, tests and test systems for the protocols agreed to in the Workshops. This is work made available to the industry volunteers and to others making valid commitments to organized events and activities such as NCC, AUTOFACT, and OSINET. As soon as this work can be adequately documented, it is placed in the public domain through submission to the National Technical Information Service. Any organization may then obtain the work at nominal charge.

The NIST laboratories bear no other relationship to the Workshop.

6 STRUCTURE AND OPERATION OF THE WORKSHOP

6.1 Plenary

The main body of the Workshop is a plenary assembly. Any organization may participate. Representation is international. NIST prefers for the business of Workshops to be conducted informally, since there are no corresponding formal commitments within the Workshop by participants to implement the decisions reached. The guidelines followed are: 1) one vote per company or independent division, 2) only companies that regularly attend should vote, 3) only companies that plan to sell or buy a protocol should vote on its implementation decisions, 4) only companies knowledgeable of the issues should vote, and 5) no proxy votes are admissible. Other voting rules are contained in the draft Procedures Manual, Section 2.3.

6.2 Special Interest Groups

Within the Workshop there are Special Interest Groups (SIGs). The SIGs receive their instructions for their technical program of work from the plenary. The SIGs meet independently, usually during the Workshop. As technical work is completed by a SIG, it is presented to the plenary for disposition. Companies participating in a SIG are expected to participate in the plenary. Voting rules for SIGS are the same as voting rules for the plenary.

Special Interest Groups sometimes correspond with organizations performing related work, such as ANSI committees. Such correspondence should be sent through the plenary to the parent committee, such as ANSI X3T5 or ANSI X3S3. When SIG meetings take place between Workshops, the correspondence from these meetings should be addressed directly to the parent committee and copied to the Workshop plenary.

Following are procedures for cooperative work among Special Interest Groups.

- o Any SIG (SIG 1) or individual having issues to discuss with or requirements of another SIG (SIG 2) should bring the matter to the attention of the chairperson of that SIG (SIG 2).
- o The SIG 2 chairperson should bring the matter before SIG 2 for action.
- o SIG 2 should respond to the concerns or needs of SIG 1 or the individual in a timely manner.
- o If the matter cannot be satisfactorily resolved or if the request is outside the charter assigned to SIG 1, then it should be brought before the plenary.
- o SIGs are expected to complete work in a timely manner and bring the results before the plenary for disposition. However, the plenary may elect to act on any issue within the scope of the workshop at any time.

Following are the charters of the Special Interest Groups.

FTAM SIG

Scope

- o to develop stable FTAM Agreements between vendors and users for the implementation of interoperable products
- o in particular to maintain the FTAM Phase 2 and Phase 3 specifications with respect to experiences from implementations and from testing. It is a goal that FTAM Phase 3 will remain backward compatible with FTAM Phase 2.
- o to act as Registration Authority for OIW FTAM objects.
- o to define further FTAM functionality.
- o to conduct liaison with standardization bodies such as ISO SC 21 and ANSI X3T5.5.
- o to conduct liaison with and contribute to other bodies working on FTAM harmonization such as the Regional Workshops (EWOS, AOW) and the ISO SGFS to define Functional Standards

  and

- o to conduct liaison with vendor/user groups such as COS, MAP, TOP, and SPAG

High priority work items:

- o Maintain FTAM Phase 2 and Phase 3 Agreements
- o Maintain OIW FTAM object register
- o Contribute to development of FTAM ISPs
- o Specify use of general Character Set Agreements
- o Specify requirements of FTAM to a Directory Service
- o Specify use of Filestore Management functions

Low priority work items:

- o Specify use of Security functions
- o Specify use of Overlapped Access

X.400 (MESSAGE HANDLING SYSTEMS) SIG

Scope of Work:

- o To develop Stable MHS Agreements among Vendors and Users for the implementation of interoperable products.
- o To conduct Liaison with Standardization Bodies, such as X3V1 as ANSI TAG to ISO/IEC JTC1 SC18, U. S. CCITT Study Group D for input to Study Group VII/Q18, and U. S. CCITT Study Group A for input to Study Group I.
- o To Actively work with other Regional Bodies, primarily (EWOS, AOW) but including others, to define International Standardized Profiles (ISPs) for CCITT X.400 MHS, and ISO/IEC MOTIS.
- o To Review Abstract Tests for X.400 and MOTIS and provide feedback to appropriate bodies.

Current Work Items:

- o MHS use of X.500 Directory
- o Body Parts / Content Types
- o MHS Security Issures
- o Access Units
- o MHS Registration Issues
- o Maintain 1984 MHS Stable Agreements
- o Contribute to development of MHS ISPs
- o MHS routing.

Future Work Items for Next Year:

- o EDI over X.400 and MOTIS
- o Distribution Lists over X.400 and MOTIS.

LOWER LAYER SIG

The Lower Layer SIG will study OSI layers 1-4 and produce recommendations for implementations to support the projects undertaken by the workshop and the work of the other SIGs. Both connectionless and connection-oriented modes of operation will be studied. The SIG will accept direction from the plenary for work undertaken and the priority which it is assigned.

The objectives of the Lower Layer SIG are:

Study OSI layers 1-4 as directed by the plenary - such study is to include management objects, security, ISDN user-network interfaces for use in conjunction with OSI network services, routing exchange protocols, etc.

Produce and maintain recommendations for implementation of these layers,

- Where necessary, provide input to the relevant standards bodies concerning layers 1-4, in the proper manner, and

Review base standard abstract test suites with the goal of identifying the test cases required for the layer 1-4 Implementation Agreements. Develop test cases for Implementation Agreement functionality not present in the base standard (if any).

<u>OSI SECURITY ARCHITECTURE SIG</u>

GOAL: To develop an overall OSI Security Architecture which is consistent with the OSI reference model and which economically satisfies the primary security needs of both the commercial and Government sectors.

APPROACH: To define a security architecture encompassing the security addenda presently being specified at certain OSI layers, the required cryptographic algorithms and related key management functions, and the security management functions which must be performed between the layers and the peer entities defined in the OSI architecture.

OBJECTIVES:

- to develop agreements based on IS/DIS
- to develop/draft NWI proposals for submission to national bodies on areas not covered by existing standards work
- to draft contributions on proposed NWIs
- to register security objects
- to provide consultancy to other SIGs
- to act as a well-focused group
    - to propagate security information
    - to recommend and coordinate activities.

DIRECTORY SERVICES SIG

o To achieve the general goal of:

- The production and promotion of functional implementation agreements based on ISO/CCITT specifications for Directory Services in accordance with the goals and objectives of the OSI Implementors' Workshop

o by fulfilling the objectives of:

- providing and maintaining functional implementation agreements for Directory Services in the form of Stable and Working OIW Implementors' Agreements;
- serving in a leadership role in the development of an International Standardized Profile for Directory Services;
- Developing Agreements on security issues as related to Directory Services;
- serving in a consultative role to the other SIGs in the use of Directory Services by other OSI Applications; and
- registering Directory Services objects as necessary to accomplish the other objectives of the SIG.

VIRTUAL TERMINAL SIG

Scope

To develop agreements concerning implementation and testing of Virtual Terminal systems based on ISO 9040/9041 and their addenda. To monitor the X-window system and potentially develop implementors agreements for OSI compatibility.

Objectives

o Develop VTE-profiles to support diverse interactive applications and environments.

o Develop Control Objects which may be referenced and used within VTE-profiles.

o Register and maintain OIW VT objects.

o Conduct liaison with standards organizations, other regional workshops and vendor/user groups as necessary.

o Review and, if necessary, generate abstract test cases for VTE-profiles.

o Harmonize OIW VTE-profiles with those from other regional workshops.

o Adopt ISP format for OIW VTE-profiles under development.

- o Migrate existing OIW VTE-Profiles to ISP format.
- o Develop X-OSI Implementors' Agreement, if necessary.
- o Register and Maintain OIW X-OSI Objects.
- o Adopt ISP Format for OIW X-OSI Implementors' Agreements, if necessary.
- o Review and, if necessary, generate abstract test cases for X-windows.

High Priority

- o Maintain stabilized OIW VTE-profiles and Control Objects.
- o Develop fully general TELNET profile in ISP format.
- o Develop Scroll Profile in ISP format.

Low Priority

- o Develop abstract test cases.
- o Develop Page profile.
- o Migrate stable profiles to ISP format - Forms, TELNET, X.3, Transparent.

<u>UPPER LAYERS SIG</u>

The charter of the Upper Layers SIG is as follows.

Develop product level specifications for the implementation of:

- o Session service and protocol
- o Presentation service and protocol
- o ACSE service and protocol
- o Remote Operations Service Element (ROSE)
- o Reliable Transfer Service Element (RTSE)

In addition, the specifications to be developed by the Upper Layers SIG will address issues that are common to layers 5-7 such as addressing, registration, etc. This SIG will review output and proposals from other SIGs to ensure consistency with international standards regarding Upper Layer Architecture.

The specifications developed will be done to support the requirements of all ASE SIGs.

The objectives of the Upper Layers SIG are to:

- o Study OSI Session, Presentation, ACSE, ROSE, RTSE and CCR.
- o Produce and maintain recommendations for implementations of these layers,
- o Where necessary provide input to the relevant standards bodies concerning Session, Presentation, ACSE, ROSE, RTSE, and CCR.
- o React in a timely manner (i.e., to develop corresponding implementor's agreements) to technical changes in ISO documents.

The following are the guidelines under which the Upper Layers SIG will operate:

Align implementation agreements with other organizations such as EWOS, AOW, and ISO,

- o Develop implementor's agreements that promote the efficiency of protocol implementations.
- o Develop implementor's agreements that promote ease in the verification of interoperability, Develop necessary conformance statements.

<u>NETWORK MANAGEMENT SIG</u>

Will use phased workload approach to accommodate volume of emerging OSI management-related standards,

The SIG will:

- o Agree upon NIST Implementors OSI systems management reference model
- o Develop product level specifications for implementations, relating to common services/protocols for exchanging management information between OSI nodes
- o Develop product level specifications for implementations relating to specific management services for exchanging fault management (FM), Security Management (SM), Configuration Management (CM), Accounting Management (AM), and Performance Management (PM) information between OSI nodes
- o Initiate and coordinate with appropriate layer SIGs product level specifications of layer-specific management information to support FM, SM, CM, AM, and PM.

As necessary, the SIG will:

- o Establish liaisons with various standards bodies
- o Provide feedback for additional/enhanced services and protocols for OSI management

OFFICE DOCUMENT ARCHITECTURE

Scope

To develop agreements concerning implementation and testing of Office Document Architecture (ODA) systems based on ISO 8613, its addenda and related international standards.

Objectives

- o Develop ODA document application profiles to support a diverse set of applications and environments;
- o Register and maintain ODA document application profiles;
- o Conduct liaison with standards organizations, other groups developing ODA document application profiles, vendor/user groups and testing authorities as necessary;
- o Review and, if necessary, generate abstract test cases for ODA document application profiles;
- o Harmonize OIW ODA document application profiles with those from other international groups; and
- o Participate, as necessary, in the ISO ISP processing of FOD-type profiles.

High Priority

- o Develop and maintain OIW ODA document application profiles;
- o Harmonize OIW ODA document application profiles with other international groups; and
- o Assist in the progression of OIW ODA document application profiles through the ISO ISP process.

Low Priority

- o Develop abstract test cases;
- o Integrate addenda and extensions to the base standard into OIW ODA document application profiles; and
- o Develop awareness of ODA in vendor and user groups.

Note: The Registration SIG has effectively completed its work. The charter items below may be removed in the future.

REGISTRATION SIG

The NIST OSI Workshop Registration Authority Special Interest Group (RA SIG) will deal with OSI Registration for the following areas:

A. Registration of NIST OSI Workshop-Specified Objects.

o The NIST OSI Workshop RAD SIG will define the procedures for the operation of the NIST Registration Authority (i.e., NIST).

1. Define policies and procedures for the registration of objects defined by the NIST OSI Workshop,

2. Take account of currently existing OSI Workshop registration work,

3. Establish policies for the publication and promulgation of registered objects;

4. Liaise with other OSI Workshop SIGs, appropriate standards bodies (e.g., ANSI) and other appropriate organizations.

B. Support for ANSI (U.S.) Registration activities

Promote the registration of MHS Private and Administrative Management Domain Names, Network-Layer-Addresses, and other Administrative Objects by ANSI or a surrogate appointed by ANSI. If ANSI feels that it cannot serve as the Registration Authority or delegate its authority to another organization, then the NIST OSI Workshop RA SIG should actively support the search for another organization to carry out this work.

This SIG will conduct a self-assessment, three NIST OSI Workshop Plenary Meetings after the Charter is approved, to determine if it has fulfilled its mission. Based on this assessment, the SIG will either be disbanded or continue. This procedure will continue until the SIG is disbanded.

TRANSACTION PROCESSING SIG

o Produce TR10000-format OSI TP Profile,

o Describe TP's use of other profile services: ACSE, CCR, Pres., Dir.,

o Produce CCR profile covering TP requiremnts,

o Liaise with other internal and external organizations as required,

o Communicate with EWOS and AOW to reach goal of an aligned profile, and

o Act as registration authority for OIW TP objects, as necessary.

MANUFACTURING MESSAGE SPECIFICATION (MMS) SIG

Scope

To create an open forum for discussion and agreements pertaining to MMS and issues related to MMS.

Objectives

o To produce agreements for implementations of MMS (ISO 9506)

- To produce implementation agreements for IS implementations which enable existing DIS based implementations (such as specified in the MAP 3.0 specification) with minimal modifications to interoperate with IS implementations.

- To produce implementation agreements on MMS Companion Standards (as recognized by ISO TC184/SC5/WG2) after those have reached ISO DIS or equivalent status.

- Develop Conformance requirements

- Develop recommendations on MMS testing

As Necessary

- Respond to defect reports as accepted

- Provide feedback on Addendum material

- To produce implementation agreements on any ISO DIS (or higher level) or equivalent document defining alternate mappings of MMS to an OSI or other international standards based manufacturing communications architecture such as might be progressed from IEC SE 65

- Provide input on ISP for MMS when the ISO process for it is defined

<u>High Priority Work Items</u>

- Define a subset of MMS (ISO-9506) suitable for initial implementations

- Produce a set of implementation agreements appropriate to that initial subset of MMS encompassing the objectives

- Study ISO test methodologies and produce recommendations for MMS test implementations. If necessary, provide input on MMS specific requirements for the ISO test methodologies

- Provide input to ISO on Abstract Test Cases to facilitate conformance and interoperability testing on the initial subset

- Provide input to ISO on the elaboration of service procedures for error conditions and on the relation of the use of specific error codes to these error conditions for the initial subset.

<u>Low Priority Work Items</u>

Study and comment on DP level or equivalent documents relating to MMS activities defined in the objectives

Develop subsequent subsets of MMS

- Produce a set of implementors agreements for the subsequent subsets

- Provide input on Test Cases for the subsequent subsets

- o Provide input on errors for the subsequent subsets

- o Provide input to ISO on MMS ASE specific management entities.

REMOTE DATABASE ACCESS SIG

Scope:

For all RDA Implementations based on ISO 9579:

- o Develop Implementors' agreements;

- o Provide input to national and international standards organizations on RDA related standards and profiles;

- o Coordinate with other organizations on matters relevant to RDA.

Objectives:

- o Use ISO 9579 Generic RDA and the ISO SQL Specialization as a basis for Implementors' Agreements on the RDA SQL ASE and its application contexts;

- o Provide input to ANSI and ISO on the specification of an RDA ISP.

High Priority Work Items

1. To develop a work plan for RDA Implementors' Agreements with an associated time schedule, using the following tasks as a basis:

   a. review ULA agreements affecting RDA implementations,

   b. specify limits on encodings in RDA pdus,

   c. specify minimum conformance requirements for RDA implementations,

   d. identify and describe recommended practices in the implementation of RDA services and protocols,

   e. identify implementor defined items in ISO 9075 (SQL) affecting interoperability in an OSI environment,

   f. identify implementor defined items in ISO 9579 (RDA) affecting interoperability,

   g. identify RDA implementation requirements for CCR and TP,

   h. harmonize ULA requirements with SQL requirements with respect to handling of variant character sets in RDA.

Low Priority Work Items

1. Future RDA specializations, if any.

7 POINTS OF CONTACT

OSI Workshop -Chairman	Tim Boland	NIST	(301) 975-3664
OSI Workshop - Registration	Brenda Gray	NIST	(301) 975-3664
Directory Services - Vice Chair	You-Bong Weon-Yoon	AT&T Bell Labs	(201) 522-5073
FTAM SIG	Darryl Roberts	Unisys NCG	(805) 499-6698
Lower Layers SIG	Fred Burg	AT&T	(201) 949-0919
Manufacturing Message Secification (MMS) SIG	Herbert Falk	SISCO	(313) 774-0070
Network Management SIG - Co-Chairs	Paul Brusil George Mouradian	Mitre AT&T Bell Labs	(617) 271-7632 (201) 949-7671
ODA SIG	Frank Dawson	IBM	(214) 556-5052
Remote Database Access SIG	Peter Eng	IMB Canada	(416) 448-3087
Security SIG	James Galvin	Trusted Info. Systems	(301) 854-6889
Technical Liaison Committee	Einar Stefferud	NMA-Northrop	(714) 841-3711
Transaction Processing SIG Acting Chair	Andrew P. Schwartz	IBM Corporation	(415) 855-4766
Upper Layers SIG	Mark Thomas	AT&T Bell Labs	(201) 522-6671
Virtual Terminal SIG	Luke Lucas	Control Data Corporation	(612) 482-2874
X.400 SIG	Barbara Nelson	Retix	(213) 399-1611
MAP	Gary Workman	GM	(313) 947-0599
TOP	Laurie Bride	BCS	(206) 763-5719
Government OSI Profile	Jerry Mulvenna	NIST	(301) 975-3631

8 PROFILE CONFORMANCE

See Stable Implementation agreements, Version 3, as of September 14, 1990.

Table of Contents

List of Figures

2 SUBNETWORKS

Editor's Note: All references to Stable Agreements in this Section are to Version 3 dated September 1990.

1 INTRODUCTION

(Refer to Stable Implementation Agreements Document)

2 SCOPE AND FIELD OF APPLICATION

(Refer to Stable Implementation Agreements Document)

3 STATUS

This material is current as of September 14, 1990.

Editor's Note: The FDDI material in particular has been identified as a candidate for stability in December 1990.

4 ERRATA

Errata are reflected in replacement pages of Version 3, Stable Document, dated September 1990.

5 LOCAL AREA NETWORKS

(Refer to Stable Implementation Agreements Document)

5.1 IEEE 802.2 Logical Link Control

(Refer to Stable Implementation Agreements Document)

5.2 IEEE 802.3 CSMA/CD Access Method

(Refer to Stable Implementation Agreements Document)

5.3 IEEE 802.4 Token Bus Access Method

(Refer to Stable Implementation Agreements Document)

5.4 IEEE 802.5 Token Ring Access Method

(Refer to Stable Implementation Agreements Document)

5.5 Fiber Distributed Data Interface (FDDI)

5.5.1 Token Ring Media Access Control (MAC, X3.139-1987)

The following are implementation agreements with respect to FDDI MAC.

1 The address length shall be 48 bits.

2 There shall be some manual or programmatic means of resetting stations and concentrators to the values specified as defaults herein or in X3.139-1987.

3 The default value of T_Max shall be at least 165 milliseconds and not more than 200 milliseconds.

4 The default value of T_Req shall be equal to either T_MAX or T_Req_Max[1] whichever is less.

5 All FDDI stations shall process Info_Fields of 0 to 4478 bytes. The frame is defined as follows:

[1] T_Req_Max is defined in the Ring Management (RMT) section of Station Management. It is used in the resolution of duplicate address problems which prevent ring initialization. Stations which have detected that they are duplicates during ring initialization take action to make sure that they lose the Claim process to other stations having a T_Req value less the T_Req_Max. T_Req_Max is specified in SMT to have a value $\geq$ 167.8 millisecond.

P	SD	FC	DA	SA	Info	FCS	ED	FS

Figure 1 - FDDI FRAME FORMAT

P: Preamble
- 16 Idle Symbols for Transmitting
- >=6 Idle Symbols for Copying
- >=2 Idle Symbols for Repeating

SD: Starting Delimiter (2 Symbols, JK)
FC: Frame Control (2 Symbols)
DA: Destination Address (12 Symbols)
SA: Source Address (12 Symbols)
INFO: Information Field (=<8956 Symbols)
FCS: Frame Check Sequence (8 Symbols)
ED: Ending Delimiter (1 Symbol)
FS: Frame Status (3 Symbols)

6 Stations shall not use restricted token service.

5.5.2 Token Ring Physical Level (PHY,X3.148-1988)

The following implementation agreement is with respect to the FDDI PHY specifications.

1 The average delay, that is the time between when a station receives a Starting Delimiter (JK symbol pair) beginning a valid frame until it repeats that Starting Delimiter, when that Starting Delimiter is preceded by a sequence of a valid frame followed by 50 Idle Symbols shall not exceed:

- one microsecond in a station, and

- one microsecond times the number of ports in a concentrator, in addition to the delays contributed by the active slaves of the concentrator.

The measurement method described above allows a consistent repeatable measurement, however it does not measure maximum possible delay. When the delay is one microsecond as measured above, the maximum effective delay contribution component which can result is 1.164 microseconds. This number, not one microsecond, should be used per PHY to compute maximum possible network delay.

5.5.3 Physical Layer Media Dependent (PMD, X3.166-1989)

The following implementation agreements are with respect to the FDDI PMD specification.

1 Stations shall repeat all valid packets under all signal conditions specified in section 5.2 of X3.166-1989, "Active Input Interface", with a bit error rate (BER) of not more than 2.5 x 10^{-10}.

2 Stations shall repeat all valid packets under all signal conditions specified in section 5.2, "Active Input Interface", except that the Minimum Average Power shall be -29 dBm (2 dB above the specified minimum), with a BER of not more than 10^{-12}.

6 X.25 WIDE AREA NETWORKS

6.1 Introduction

(Refer to the Stable Implementation Agreements Document).

6.2 ISO 7776

(Refer to the Stable Implementation Agreements Document).

6.3 ISO 8208

(Refer to the Stable Implementation Agreements Document).

7 INTEGRATED SERVICES DIGITAL NETWORKS (ISDN)

7.1 Introduction

(Refer to the Stable Implementation Agreements Document).

7.2 Implementation Agreements

(Refer to the Stable Implementation Agreements Document).

7.2.1 Physical Layer, Basic Access at "U"

(Refer to the Stable Implementation Agreements Document).

7.2.2 Physical Layer, Basic Access at S and T

(Refer to the Stable Implementation Agreements Document).

7.2.3 Physical Layer, Primary Rate at "U"

(Refer to the Stable Implementation Agreements Document).

7.2.4 Data Link Layer, D-Channel

(Refer to the Stable Implementation Agreements Document).

7.2.5 Signaling

(Refer to the Stable Implementation Agreements Document).

7.2.6 Data Link Layer B-Channel

(Refer to the Stable Implementation Agreements Document).

7.2.7 Packet Layer

(Refer to the Stable Implementation Agreements Document).

ANNEX A

(Refer to the Stable Implementation Agreements Document.)

A.1 Data Link Layer, D-Channel

(Refer to the Stable Implementation Agreements Document.)

A.2 Signaling

(Refer to the Stable Implementation Agreements Document.)

Table of Contents

3 NETWORK LAYER

Editor's Note: All references to Stable Agreements in this Section are to Version 3 dated September 1990.

1 INTRODUCTION

(Refer to the Stable Agreements Document)

2 SCOPE AND FIELD OF APPLICATION

(Refer to the Stable Agreements Document)

3 STATUS

This material is current as of September 14, 1990.

Editor's Note: The priority material (Sections 3.5.1 and 3.11) and the addressing material (Section 3.7) should be examined closely for possible stability in December 1990.

4 ERRATA

Errata are reflected in pages of Version 3, Stable Document, dated September 1990.

5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS)

5.1 ISO 8473

1. Subsets of the protocol:

(Refer to the Stable Implementation Agreements Document).

2. Mandatory Functions:

(Refer to the Stable Implementation Agreements Document).

3. Optional Functions:

o (Refer to the Stable Implementations Agreements document).

o Intermediate systems implementing priority shall do so as described below. For End system network entities the implementation of priority is optional, but if implemented it shall also be done as described below.

1 NPDUs shall be scheduled based on the priority functions of ISO 8473. The scheduling algorithm for achieving this priority function is left as a local matter. It is required that the following constraints be met as described below.

- An NPDU of lower priority shall not overtake an NPDU of higher priority in an intermediate system (i.e., exit an IS ahead of a higher priority NPDU arriving before it).

- A minimum flow shall be provided for lower priority PDUs.[1]

2 According to ISO 8473, the priority level is a binary number with a range of 0000 0000 (lowest priority) to 0000 1111 (highest priority level). Within this range, the four abstract values corresponding to the four levels defined in section 3.11 shall be encoded as follows:

- "high reserved" priority will be encoded with value 14 (0000 0000 0000 1110),

- "high" priority will be encoded with value 10 (0000 0000 0000 1010),

- "normal" priority will be encoded with value 5 (0000 0000 0000 0101), and

- "low" priority will be encoded with value "zero" (0000 0000 0000 0000)

For a receiving network entity, a value lower than 5 shall be considered as "low"; a value lower than 10 and higher than 5 shall be considered as "normal", and a value lower than 14 and higher than 10 shall be considered as "high".

3 Network entities supporting priority shall process PDUs in which the priority parameter is absent as either "low", "normal", or "high" according to a locally configurable parameter. This is to ensure that NPDUs not containing the priority parameter can be processed by intermediate systems in a defined manner with respect to those which do contain the priority parameter.

4 IEEE 802.4 and IEEE 802.5 local area networks as well as some X.25 networks implementations have the ability to support subnetwork priorities. When available, a subnetwork priority function should be utilized in support of the priority requested of the network layer. The mapping of network layer priority levels onto subnetwork priority levels is a local configuration matter.

[1] The scheduling algorithm by which this is accomplished is for further study.

5.2 Provision of CLNS over Local Area Networks

(Refer to the Stable Agreements Document)

5.3 Provision of CLNS over X.25 Subnetworks

(Refer to the Stable Agreements Document)

5.4 Provision of CLNS over ISDN

(Refer to the Stable Implementation Agreements document).

5.4.1 CLNP Utilizing X.25 Services

(Refer to the Stable Implementations Agreements document).

5.5 Provision of CLNS over Point-to-Point Links

(To be based on ISO 8880)

6 CONNECTION-MODE NETWORK SERVICE

6.1 Mandatory Method of Providing CONS

6.1.1 General

(Refer to the Stable Implementation Agreements document).

6.1.2 X.25 WAN

(Refer to the Stable Implementation Agreements document).

6.1.3 LANs

(Refer to the Stable Implementation Agreements document).

6.1.4 ISDN

(Refer to the Stable Implementation Agreements document).

6.1.5 PRIORITY

Priority for CONS will be addressed with the implementation of X.25-1988 in a future version of these agreements.

6.2 Additional Option: Provision of CONS over X.25 1980 Subnetworks

(Refer to the Stable Implementation Agreements Document)

6.3 Agreements on Protocols

(Refer to the Stable Implementation Agreements Document)

6.3.1 ISO 8878

(Refer to the Stable Implementation Agreements Document.)

6.3.2 Subnetwork Dependent Convergence Protocol (ISO 8878/Annex A)

(Refer to the Stable Implementation Agreements Document)

6.4 Interworking

(Refer to the Stable Implementation Agreements Document.)

7 ADDRESSING

- Refer to the Stable Implementations Agreements Document

o Within routing domains intending to operate using the IS -IS Intradomain Routing Protocol defined in ISO/IEC JTC 1/SC 6 N4945, it is recommended that the DSP have a binary abstract syntax and that the last nine octets are structured as follows:

2 octets	6 octets	1 octet
AREA	ID	N-Selector

where the AREA field identifies a unique subdomain of the routing domain, the ID field identifies a unique system within an area, and an N-SELECTOR identifies a user of the Network Layer Service.

See the OSI Routing Framework document (ISO/TR 9575) for definitions of the above terms and concepts.

The above recommendation may be applicable in other routing environments.

8 ROUTING

8.1 ISO 9542 End System to Intermediate System Routing

(Refer to the Stable Implementation Agreements Document.)

10. ISO 8473 PDUs multicast as a result of the Query Configuration function shall use the Network Layer Protocol ID (NLPID) assigned to ISO 8473.

11. An ISO 8473 PDU received as a result of another ES having performed the Query Configuration function shall be processed as follows:

    - If the ISO 8473 PDU is addressed to one of the NSAPs present in the ES, the End System shall process the PDU according to the applicable clauses of ISO 8473 and invoke the Configuration Response Function (clause 6.6 of ISO 9542)

    - If the ISO 8473 PDU is not addressed to one of the NSAPs present in the ES, the End System shall discard the PDU without generating as ISO 8473 Error Report

12. For purposes of address matching and SNPA extraction, the first octet of the option parameter value of an address (clause 7.4.5) or SNPA Mask (clause 7.4.6) shall be aligned with the first octet (AFI) of the encoded trial NSAP Address.

The following items represent proposed solutions to defects in ISO 9542. These solutions are being progressed as defect reports to ISO 9542. These items will be deleted when the corresponding defect report is approved.

> An End System may choose to ignore an RD PDU received for a destination to which the ES has not sent traffic for some period of time. An ES must record redirection information only for those other systems with which it is in active communication.
>
> A holding time value of zero is permitted. When configuration and/or redirection information with a zero holding time is received, prior information shall be replaced, thus causing the system to set its holding timer to zero and discard the corresponding information.
>
> If one or more ISs suggested an ESCT, the minimum of the non-zero suggested values replaces the current value of the ES's CT.

8.2 DIS 10030 End System to Intermediate System Routing

The protocol used to provide End System to Intermediate System routing in support of the CONS (refer to section 3.6) shall be DIS 10030.

The following agreements apply to the use of DIS 10030:

1. A management mechanism capable of adding and deleting entries in the Routing Information Base (RIB) of both SNAREs and End Systems is recommended. When using the management mechanism to add an entry it should not be timed out, and the entry should be write protected from alteration by the DIS 10030 protocol.

8.3 Intra-Domain Intermediate to Intermediate Systems Routing

The protocol used to provide Intermediate System to Intermediate System routing in support of the CLNS (refer to section 3.5) among systems in a single routing domain shall be DP 10589

The following agreements apply to the use of DP 10589:

1. A management mechanism capable of configuring the Identifier, Characteristic, and Status attributes of the managed objects of clause 11 shall be provided.

8.4 Inter-Domain Intermediate Systems to Intermediate Systems Routing

An Administrative Authority shall determine the procedures and policies that govern the exchange of routing information with other routing domains.

Intermediate systems shall provide management mechanisms to configure the required inter-domain routing information.

9 PROCEDURES FOR OSI NETWORK SERVICE/PROTOCOL IDENTIFICATION

9.1 General

(Refer to the Stable Implementation Agreements document).

9.2 Processing of Protocol Identifiers

(Refer to the Stable Implementation Agreements document).

9.2.1 Originating NPDUs

(Refer to the Stable Implementation Agreements document).

9.2.2 Destination System Processing

(Refer to the Stable Implementation Agreements document).

9.2.3 Further Processing in Originating End System

(Refer to the Stable Implementation Agreements document).

9.3 Applicable Protocol Identifiers

(Refer to the Stable Implementation Agreements document.)

10 MIGRATION CONSIDERATIONS

This section considers problems arising from evolving OSI standards and implementations based on earlier versions of OSI standards.

10.1 X.25-1980

(Refer to the Stable Agreements Document)

11 USE OF PRIORITY[2]

11.1 Introduction

Within the OSI environment, Quality of Service (QoS) parameters are intended to influence the qualitative behavior of the various OSI Layer entities. QoS is described in terms of parameters related to performance, accuracy, and reliability (e.g. delay, throughput, priority, error rate, security, failure probability, and etc.).

[2] This section provides initial proposals on the use of priority. The proposal requires further technical review before considering it as having support as an implementation agreement. Refer to the following documents for further technical information:

LLSIG 88-64 LLSIG 88-120 LLSIG 88-122

QoS covers a broad spectrum of issues. As a first step, these agreements address the efficient sharing of Layer 1, 2, & 3 transmission resources by making use of the priority parameter. To accomplish this, implementation agreements and encodings are provided for Network and Transport Layer protocols. The implication of these agreement for upper layer protocols is limited to the conveyance of priority information in both directions between an application entity and the service boundary for the Transport Layer.

The implementation of priority as defined herein is optional for intermediate systems and end systems, but if implemented shall be as defined in the layer specific agreements (for Network Layer see section 3.5.1; for Transport Layer see section 4.5.1.2.6, and for Upper Layers the section will be included at a later date).

11.2 Overview

The purpose of the priority parameter, in the context of the lower layers, is to influence the scheduling of the transmission of data on subnetworks, in CONS as well as CLNS environments (end systems as well as intermediate systems). The priority parameter as defined is to be used by OSI Applications to control the "priority of data". Within the lower layers this translates into a contention for transmission resources, which has a direct impact on performance.

In order to implement practical mechanisms for scheduling the transmission of data units while maintaining the usefulness of priority, the specification of priority levels is limited to four; one corresponding to each of the four service classes:

- low priority
- normal priority
- high priority
- high reserved priority

The high reserved priority level is intended primarily for OSI network management purposes. The three lower priority levels are intended for information exchange by users.

These four priority levels are used, from an applications point of view, in the various communications lower layers (Transport, Network and Data Link) to provide a consistent mapping of "abstract priority levels" in and n-service onto the n-1 service and when available, priority parameter values in the layer protocol. In the upper layers (ASCE, Presentation and Session) local mechanisms are expected to be provided to application layer ASEs with a means for conveying priority information in both directions through the communication upper layers.

For example, this implies that an application request for a high priority service will be conveyed through association/presentation/session and will result in a high priority data transport connection and either high priority data CLNP PDUs (CLNS case) or a high priority data network connection/X.25 virtual call (CONS case).

12 CONFORMANCE

(Agreements to be added at a later date)

13 ANNEX A

Editor's Note: This material was moved to the Stable Document in June 1990. It was not considered an implementor agreement prior to June 1990.

Table of Contents

4 TRANSPORT LAYER

Editor's Note: All references to Stable Agreements in this Section are to Version 3 dated September 1990.

1 INTRODUCTION

(Refer to Stable Implementation Agreements Document)

2 SCOPE AND FIELD OF APPLICATION

(Refer to the Stable Implementation Agreements document).

3 STATUS

This material is current as of September 1990.

4 ERRATA

Errata are reflected in pages of Version 3, Stable Document, dated September 1990.

4.1 ISO/CCITT Defect Reports

This section lists the defect reports from ISO which are currently recognized to be valid for the purpose of NIST conformance.

5 PROVISION OF CONNECTION MODE TRANSPORT SERVICES

(Refer to the Stable Implementation Agreements document).

5.1 Transport Class 4

5.1.1 Transport Class 4 Overview

(Refer to the Stable Implementation Agreements document).

5.1.2 Protocol Agreements

5.1.2.1 General Rules

(Refer to the Stable Implementation Agreements Document.)

It is recommended that the capability of request acknowledgements be supported and proposed in CR TPDUs.

If request acknowledgements are supported, then if the implementation delays acknowledgements it shall:

a) request use of request acknowledgements in the CR TPDU

b) accept the use of request acknowledgements in the CC TPDU if it was proposed in the CR TPDU.

It is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU. If a CR TPDU is received with the "preferred" parameter and the preferred maximum TPDU size parameter is supported, the preferred parameter shall be returned in the CC TPDU and the existing TPDU size parameter in the CR TPDU shall be ignored.

It is recommended that inactivity timer values be exchanged during connection establishment. This may be mandatory in the future.

If the "exchange of inacitivity timers" capability is supported, the implementation shall send its minumum inactivity timer in the CR TPDU. If a CR TPDU is received with this timer value and the capability is supported, the responding CC TPDU shall contain the inactivity time.

If the Inactivity time is received and the capability is supported, the following shall be used as an upper bound for w:

$(IR-E_{LR})/N \geq W \qquad N \geq 2$

5.1.2.2 Transport Class 4 Service Access Points or Selectors

(Refer to the Stable Implementation Agreements Document.)

5.1.2.3 Retransmission Timer

Editor's Note: The following text will be added to the stable agreement after the existing text. The existing text will be labelled "Example One" and this text will be labelled "Example Two". The following sentence will also be added to the introduction paragraph: "Example one represents a simple retransmission strategy and example two is particularly suitable for networks subject to high traffic loads".

As network load increases, the variability of round-trip delay also increases. In environments where load fluctuates widely, it is therefore useful to estimate the variability of round-trip delay measurements and use this estimate in the calculation of retransmission timer values. An estimate of the variability of round-trip delay measurements can be efficiently calculated as an exponentially weighted average of the differences between round-trip delay measurements and the average round-trip delay. This represents the mean deviation of the round-trip delays, which is a useful approximation of the standard deviation and can be much more efficiently computed. The formula is

$$D \leftarrow D + (1 - a)(|S - E| - D)$$

where D is the estimate of variability in round-trip delays. S, E, and a are as defined for the preceding formula. As before the value of a must be between 0 and 1 and the choice of a value of $1 - 2^{-N}$ allows for efficient update of the average. The value of a for the variability estimation, though, does not need to be the same as that used for the round-trip delay estimate. A smaller value for a is useful in the variability estimation to cause a more rapid response to changes in round-trip delays. D can then be used to calculating retransmission timer values according to the formula:

$$T1 \leftarrow E + AR + kD$$

where $T1$ is the retransmission timer value, E is the estimated average round-trip delay, AR is the value of the acknowledgement timer parameter received from the remote transport service provider during connection establishment, and k is a locally administered factor. Since D approximates the standard deviation of the round-trip delays, but is greater than or equal to the standard deviation, round-trip delays within k standard deviations of the mean would be accounted for by the retransmission timer value (eg. $k = 2$, if round-trip delays were normally distributed, would account for 95% of the variability).

Round-trip time measurements based on acknowledgement of any retransmitted data should not be used to update the round-trip delay estimate or the estimate of variability. Such measurements are not reliable since it is ambiguous which transmission of the data is being acknowledged.

One strategy for handling a retransmission timeout is to retransmit the PDU and reset the timer with a value that is twice the previous value. In this case, a new roundtrip delay estimate and estimate of variability should be calculated only when an acknowledgement of data is received where none of the acknowledged data has been retransmitted. This calculation uses the new round-trip delay measurement and the last estimate before the retransmission timeout(s).

5.1.2.4 Keep-Alive Function

(Refer to the Stable Implementation Agreements Document.)

5.1.2.5 Congestion Avoidance Policies

(Refer to the Stable Implementation Agreements Document).

5.1.2.6 Use of Priority[1]

For end systems, the implementation of priority is optional, but if implemented, one of the four values defined in section 3.11 shall always be used in an instance of communications. In other words an explicit priority parameter shall be sent.

Additional requirements of systems implementing priority are defined below.

1 When Transport is implemented over a CLNS Network entity, each data TPDU and corresponding NSDU shall be assigned a priority level derived from the Transport connection priority level, except as excluded in item 5b and 5d below[2].

2 A local mechanism shall be provided to convey priority information to the Network service. If appropriate, simultaneous Transport service request can be managed on a priority basis within the Transport Layer.

3 The four abstract values corresponding to the four levels defined in 3.11 shall be encoded as follows:[3]

- o "high reserved" priority will be encoded with value "zero" (0000 0000 0000 0000), and
- o "high" priority will be encoded with value 5 (0000 0000 0000 0101),
- o "normal" priority will be encoded with value 10 (0000 0000 0000 1010),
- o "low" priority will be encoded with value 14 (0000 0000 0000 1110)

4 Other values should be interpreted as follows: a value lower than 5 and higher than 0 shall be interpreted as "high", a value lower than 10 and higher that 5 shall be interpreted as "normal", and a value higher than 10 shall be interpreted as "low".

5 The exchange of priority parameters by Transport entities is performed as described below[4].

a If priority is implemented in the end system, a priority value corresponding to one of the four abstract levels defined in section 3.11 will be conveyed down to the Transport entity and shall be encoded and sent in the CR TPDU as the priority level "desired" for the Transport connection.

[1] Refer to clause 3.11 for an overview on the use of priority.

[2] The approach to assigning priority to an NSDU is for further study.

[3] This encoding has been chosen to be consistent with ISO 8073, The results is a reverse encoding from that for ISO 8473.

[4] ISO 8073 does not define or support a sound negotiation mechanism at this time; the following process will serve to allow a priority level to be established for a TC.

b A receiving Transport entity supporting priority management shall either accept the priority level proposed in the CR TPDU or select a lower level. The CR shall not be rejected solely because of the "desired" priority level. The selected priority level shall be encoded and returned to the calling Transport entity in the CC TPDU. The TC priority is also passed to the local session entity with the T-Connect indication primitive and is eventually conveyed to the ASE, which can reject the association if the priority is unacceptable.

If the receiving Transport entity supports priority but receives a CR TPDU without the priority parameter, it shall associate a default priority level with the Transport connection for the purposes of managing the Transport connections which may be under its control. This default level shall not be encoded and placed in the corresponding CC TPDU and shall not result in any priority information being associated with NSDUs being passed to the Network entity supporting the Transport connection. The default shall be either "low", "normal", or "high" according to the locally configurable parameter.

c A receiving Transport entity not supporting priority management shall ignore the parameter in the CR TPDU.

d When the initiating Transport entity receives the CC TPDU containing the priority parameter, it establishes the priority for the Transport connection based on the level received and conveys this to the session entity with the T-Connect confirm primitive. If the priority parameter does not appear in the CC TPDU, the initiating Transport entity shall assume the remote Transport entity does not support priority and will therefore assign a default priority level to the Transport connection for the purposes of managing the Transport connection with respect to the other simultaneous Transport connections which may be under its control. However, this default shall not result in any priority information being associated with NSDUs being passed to the Network entity supporting the Transport connection. The default shall be either "low", "normal", or "high" according to a locally configurable parameter.

5.2 Transport Class 0

(Refer to Stable Implementation Agreements Document)

5.2.1 Transport Class 0 Overview

(Refer to Stable Implementation Agreements Document)

5.2.2 Protocol Agreements

5.2.2.1 General Rules

It is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU. If a CR TPDU is received with the "preferred" parameter and the preferred maximum TPDU size parameter is supported, the preferred parameter shall be returned in the CC TPDU and the existing TPDU size parameter in the CR TPDU shall be ignored.

5.2.2.2 Transport Class 0 Service Access Points

(Refer to Stable Implementation Agreements Document)

5.2.3 Rules for Negotiation

(Refer to Stable Implementation Agreements Document.)

5.3 Transport Class 2

(Refer to Stable Implementation Agreements Document.)

5.3.1 Transport Class 2 Overview

(Refer to Stable Implementation Agreements Document.)

5.3.2 Protocol Agreements

It is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU. If a CR TPDU is received with the "preferred" parameter and the preferred maximum TPDU size parameter is supported, the preferred parameter shall be returned in the CC TPDU and the existing TPDU size parameter in the CR TPDU shall be ignored.

6 PROVISION OF CONNECTIONLESS TRANSPORT SERVICE

(Refer to Stable Implementation Agreements Document.)

7 TRANSPORT PROTOCOL IDENTIFICATION

(Refer to the Stable Implementation Agreements document.)

Table of Contents

5 UPPER LAYERS

Editor's Note: All references to Stable Agreements in this section are to Version 3 dated September 1990.

1 INTRODUCTION

(Refer to Stable Agreements Document)

1.1 References

Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element - Addendum 1: Peer-Entity Authentication During Association Establishment, ISO 8649/DAD1 (ISO/IEC JTC1/SC21 N3771)

Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element - Addendum 1: Peer-Entity Authentication During Association Establishment, ISO 8650/DAD1 (ISO/IEC JTC1/SC21 N3772)

2 SCOPE AND FIELD OF APPLICATION

(Refer to Stable Agreements Document)

3 STATUS

This version of the upper layer agreements is under development.

4 ERRATA

4.1 ISO Defect Solutions

4.2 Session Defect Solutions Correcting CCITT X.215 and X.225

(Refer to Stable Agreements Document)

5 ASSOCIATION CONTROL SERVICE ELEMENT

5.1 Introduction

(Refer to Stable Agreements Document)

5.2 Services

(Refer to Stable Agreements Document)

5.3 Protocol Agreements

5.3.1 Application Context

(Refer to Stable Agreements Document)

5.3.2 AE Title

(Refer to Stable Agreements Document)

5.3.3 Result Parameter

If the result parameter of the AARE PDU contains the value accepted, then the result-source-diagnostic parameter shall contain the value null.

5.3.4 Peer Entity Authentication

If supported, peer-entity authentication during association establishment shall be implemented as specified in Addendum 1 to ISO 8650 (ISO 8650/DAD1).

5.4 ASN.1 Encoding Rules

(Refer to Stable Agreements Document)

5.5 Connectionless

(Refer to Stable Agreements Document)

6 ROSE

(Refer to Stable Agreements Document)

7 RTSE

(Refer to Stable Agreements Document)

8 PRESENTATION

8.1 Introduction

(Refer to Stable Agreements Document)

8.2 Service

(Refer to Stable Agreements Document)

8.3 Protocol Agreements

8.3.1 Transfer Syntaxes

(Refer to Stable Agreements Document)

8.3.2 Presentation Context Identifier

(Refer to Stable Agreements Document)

8.3.3 Default Context

(Refer to Stable Agreements Document)

8.3.4 P-Selectors

(Refer to Stable Agreements Document)

8.3.5 Provider Abort Parameters

(Refer to Stable Agreements Document)

8.3.6 Provider Aborts and Session Version

(Refer to Stable Agreements Document)

8.3.7 CPC-Type

(Refer to Stable Agreements Document)

8.3.8 Presentation-context-definition-result-list

(Refer to Stable Agreements Document)

8.3.9 RS-PPDU

(Refer to Stable Agreements Document)

8.4 Presentation ASN.1 Encoding Rules

8.4.1 Invalid Encoding

(Refer to Stable Agreements Document)

8.5 General

8.5.1 Presentation Data Value (PDV)

(Refer to Stable Agreements Document)

8.6 Connection Oriented

(Refer to Stable Agreements Document)

8.7 Connectionless

(Refer to Stable Agreements Document)

9 SESSION

9.1 Introduction

(Refer to Stable Agreements Document)

9.2 Services

(Refer to Stable Agreements Document)

9.3 Protocol Agreements

9.3.1 Concatenation

(Refer to Stable Agreements Document)

9.3.2 Segmenting

(Refer to Stable Agreements Document)

9.3.3 Reuse of Transport Connection

(Refer to Stable Agreements Document)

9.3.4 Use of Transport Expedited Data

(Refer to Stable Agreements Document)

9.3.5 Use of Session Version Number

(Refer to Stable Agreements Document)

9.3.6 Receipt of Invalid SPDUs

(Refer to Stable Agreements Document)

9.3.7 Invalid SPM Intersectlons

(Refer to Stable Agreements Document)

9.3.8 S-Selectors

(Refer to Stable Agreements Document)

9.4 Connectionless

(Refer to Stable Agreements Document)

10 UNIVERSAL ASN.1 ENCODING RULES

10.1 TAGS

(Refer to Stable Agreements Document)

10.2 Definite Length

(Refer to Stable Agreements Document)

10.3 EXTERNAL

(Refer to Stable Agreements Document)

10.4 Integer

(Refer to Stable Agreements Document)

10.5 String Types

(Refer to Stable Agreements Document)

10.6 Bit String

(Refer to Stable Agreements Document)

11 CHARACTER SETS

(Refer to part 21 -- a new chapter expressly for character sets.)

12 CONFORMANCE

(Refer to Stable Agreements Document)

12.1 Specific ASE Requirements

(Refer to Stable Agreements Document)

12.1.1 FTAM

12.1.1.1 Phase 2

(Refer to Stable Agreements Document)

12.1.2 MHS

12.1.2.1 Phase 1 (1984 X.400)

(Refer to Stable Agreements Document)

12.1.2.2 Phase 2, Protocol P1 (1988 X.400)

(Refer to Stable Agreements Document)

12.1.2.3 Phase 2, Protocol P7 (1988 X.400)

(Refer to Stable Agreements Document)

12.1.2.4 Phase 2, Protocol P3 (1988 X.400)

(Refer to Stable Agreements Document)

12.1.3 DS

12.1.3.1 Phase 1

(Refer to Stable Agreements Document)

12.1.4 Virtual Terminal

12.1.4.1 Phase 1a

(Refer to Stable Agreements Document)

12.1.4.2 Phase 1b

(Refer to Stable Agreements Document)

12.1.5 MMS

ACSE Requirements:
all

Application Context:

o "ISO MMS"

- implies use of ACSE and MMS ASE

Presentation Requirements:

Presentation Functional Units:

o kernel

Presentation Requirements:

o At least 2

Abstract Syntaxes:

o "mms-abstract-syntax-major-version 1"

Associated Transfer Syntax:

o "Basic Encoding of a single ASN.1 type"

Session Requirements:

Session Functional Units:

o kernel

o duplex

Version Number: 2

Maximum size of User Data parameter field:

10,240

12.1.6 Transaction Processing

ACSE Requirements:
all

Application Context:

The application context is user-defined.

Presentation Requirements:

Presentation Functional Units:

o kernel

Presentation Contexts:

o At least 3 must be supported if the commit functional unit of TP is not supported.
o At least 4 must be supported if the commit functional unit of TP is supported.

Abstract Syntaxes:

o "ISO 8650-ACSE1"
{ joint-iso-ccitt(2) association-control(2) abstract-syntax(1) apdus(0) version1(1) }

Associated Transfer Syntax:

o "Basic Encoding of a single ASN.1 type"
{ joint-iso-ccitt(2) asn1(1) basic-encoding(1) }

- "ISO 10026-TP"
  { joint-iso-ccitt(2) transaction-processing(?) abstract-syntax(2) tp-apdus(1) }

Associated Transfer Syntax:

- "Basic Encoding of a single ASN.1 type"
  { joint-iso-ccitt(2) asn1(1) basic-encoding(1) }

- If required, "ISO 9804-CCR"
  (TBD)

- At least one user-defined abstract syntax.

Session Requirements:

Session Functional Units:

- kernel

- duplex

- Others as required by CCR (TBD) if the commit functional unit of TP is supported.

  Version Number: 2

  Maximum size of User Data parameter field: 10,240

Annex A: Recommended Practices

(Refer to Stable Agreements Document)

Annex B: Object Identifier Register

B.1 Register Index

(Refer to Stable Agreements Document)

B. 2 Object Identifier Descriptions

(Refer to Stable Agreements Document)

Annex C: Backward Compatibility

Issue	Version & Section Changed	Backward Compatibility
Restrictions on minimum number of octets implementations shall be able to receive.	V1E2 5.5.3.2	Interworking problems may occur, since implementations could send more than 128 octets. [An implementation that conforms to versions previous to V1E2 as an initiator and V3E1 as a responder will be able to interoperate.]
Agreements on AE Title, AP Title, and AE Qualifier changed.	V1E3 section 5.5.3.3 & V1E4 section 5.5.3.3	Interworking problems may occur between implementations that expect different forms of AP Title and AE Qualifier to be used. [Implementations that accept any form of these parameters will interwork with initiators that conform to earlier versions.]
Restrictions on encoding of "Presentation Context Identifier."	V2E1 section 5.8.3.3	Interworking problems may occur since implementations could encode negative numbers. [An implementation that conforms to versions previous to V2E1 as a responder and V3E1 as an initiator will be able to interoperate.]
Mode selector as first element in set	V1E4 section 5.6.3.4	This will cause interworking problems for those implementations that don't encode "mode selector" as the first element in the set. [An implementation that conforms to versions previous to V1E4 as an initiator and V3E1 as a responder will be able to interoperate.]

Issue	Version & Section Changed	Backward Compatibility
Restrictions on encoding of "protocol version" and "presentatation requirements."	V2E1 section 5.8.4.2	This will cause interworking problems for those implementations expecting "protocol version" and "presentation requirements" to be encoded in the primitive form. [An implementation that conforms to versions previous to V2E1 as an initiator and V3E1 as a responder will be able to interoperate.]
Restrictions on encoding of "presentation selector."	V2E1 section 5.8.4.3	This will cause interworking problems for those implementations expecting "presentation selector" to be encoded in the primitive form. [An implementation that conforms to versions previous to V2E1 as an initiator and V3E1 as a responder will be able to interoperate with either version.]
Use of default values for Minor syncpoint changed.	V2E3 section 5.11.1.1.1	No backwards compatibility
Addition and deletions of abstract syntaxes.	V2E1 section 5.11.1.3.1	No backwards compatibility
Value for session functional unit "resynchronize" changed.	V2E4 section 5.11.1.4.1	No backwards compatibility
Restrictions on inclusion of "Transfer-syntax-name" in CP PPDU and CPC type.	V3E1 section 5.8.6	Interworking problems will occur for those implementations that expect "Transfer-syntax-name" parameter to be present in the PDV-List even though one transfer syntax was negotiated. [An implementation conforming to V3E1 as an initiator and versions previous to V3E1 as a responder will be able to interoperate.]

Issue	Version & Section Changed	Backward Compatibility
Enclding restrictions on ASN.1 INTEGER type describing PCI.	V3E1 section 5.10.4	Interworking problems will occur since implementations conforming to previous versions could encode PCI integer lengths greater than 4. [Responders that accept integers describing PCI that are encoded in greater than 4 octets and Initiators that conform to V3E1 will be able to interoperate.]
Encoding restrictions on BIT STRING, OCTET STRING, and CHARACTER STRING.	V3E1 section 5.10.5	Implementations that conform to previous versions can expect these strings to have nested constructed encodings and therefore interworking problems will occur. [Responders that accept nested constructed encodings and Initiators that conform to V3E1 will be able to interoperate.]
No extra trailing bits allowed in BIT STRING.	V3E1 section 5.10.6	Interworking problems will occur when implementations that conform to previous versions send extra trailing bits. [Responders accepting extra trailing bits and Initiators that conform to V3E1 will be able to interoperate.]
Restriction on usage of "token item field" and "user data."	V3E1 section 5.9.3.1	Interworking problems will occur since implementations that conform to V1E1 do not expect the "token item field" to be encoded when a category 0 SPDU is concatenated to a category 2 SPDU.
Restrictions on CPC-type values when multiple transfer syntaxes are proposed.	V2E2 section 5.8.3.9	Interworking problems may occur between initiators that send CPC-type values and receivers that do not examine them.

Issue	Version & Section Changed Occurred	Backward Compatibility
References to ISO 8649 and ISO 8650 changed.	V1E3 section "References."	Interworking problems will occur for those implementations that conform to ISO DIS 8649 and 8650. V1E3 references IS versions of 8649 and 8650.
References to ISO 8326, ISO 8327, ISO 8822, and ISO 8823 changed.	V1E4 section References.	Interworking problems will occur for those implementations that conform to 8326/DAD2, 8327/DAD2, DIS 8822, and DIS 8823. V1E4 referenced 8326/AD2, 8327/AD2, IS 8822, and IS 8823.
AE Title changed according to Amendment 1 to ISO 8650.	V3E1 section 5.5.3.2	Interworking problems will occur between initiators that use AE-title-form 1 and responders that accept only AE-Title-form 2.
Restrictions on usage of "direct references" in ABRT APDU.	V3E1 section 5.5.4	Interworking problems will occur for those implementations that expect the "direct reference" parameter to be included in the ABRT APDU. [An implementation that conforms to V3E1 as an initiator and versions previous to V3E1 as a responder will be able to interoperate.]

Table of Contents

6 Registration Authority Procedures for the OSI Implementors Workshop (OIW)

For current Registration Authority information for Workshop--Defined Objects, consult the aligned chapter of Version 3, Stable Implementation Agreements dated September 1990.

Table of Contents

7 Stable Message Handling Systems

Editor's Note: For current stable MHS agreements, consult the aligned section in the Stable Implementation Agreements document. This section serves as a reference or pointer to Stable Agreements contained in Version 3 dated September 1990.

Table of Contents

List of Figures

List of Tables

Part 8 Message Handling Systems

9.1 Use of O/R Addresses for Routing

Procurers are responsible for understanding the implications of routing requirements and capabilities.

9.2 ORAddress Attribute List Equivalence Rules

Two ORAddresses are equivalent if each contains the same set of attributes and each attribute compares in type and value.

The following equivalence rules apply when comparing a provided ORAddress with a collection of known ORAddresses. For example, in order to perform delivery of a message to a recipient, the MTA must unambiguously match the ORAddress contained in the message with the known ORAddresses. See X.402 (1988), section 18.4, for the base standard attribute equivalence rules. The following additional rules must also be applied by the delivering (or non-delivering) MTA:

a) If the provided ORAddress is an unambiguous underspecification of a known ORAddress, the ORAddresses are equivalent. For example, if the initials were omitted, the ORAddress would still be equivalent. Under-specification means that some attributes that are not present in the provided ORAddress are present in the known ORAddresses. Under-specification does not mean partial value (e.g., substring) equivalence when the same set of attributes are present in the ORAddresses.

b) Over-specified ORAddresses are not equivalent. Over-specification means that more attributes are present in the provided ORAddress than are present in the known ORAddresses.

c) An ADMD or PRMD name that is all numeric but encoded as Printable String is considered to be equivalent to the same ADMD or PRMD name, respectively, with the same numeric values encoded as Numeric String.

NOTES

1 An X.500 Directory service may or may not support these matching rules for equivalence.

2 Operational equivalence between T.61 and Printable String is for further study.

9.4 MHS Use of Directory

9.4.1 Introduction

The MHS standards recognize the need of MHS users for a number of directory service elements. Directory service elements are intended to assist users, their UAs, and MTAs in obtaining information for use in submittion, delivery, and the transfer of messages.

NOTE - The MTS may also use the directory service elements to obtain information, for example, to be used in the routing of messages. This application of the directory service is not defined by the base standards and is therefore not addressed by this Agreement.

9.4.2 Functional Configuration

Two MHS functional entities, the IPM UA and MTA, may access the Directory service using the Directory User Agent (DUA). The interface between the UA and DUA, or MTA and DUA is local and not defined. The interaction between the DUA and Directory System Agent (DSA) is specified in Chapter 11. A collocated DUA and DSA is also permitted.

9.4.3 Functionality

Examples of functional usages of directories have been identified for UAs and the MTAs in conjunction with their DUAs. These are:

a) UA Specific Functionality:

1) Verify the existence of a Directory Name.

2) Given a partial name, return a list of possibilities.

3) Search the Directory for entries containing a specified attribute type and value and return the Distinguished Names of the matching entries.

4) Return the O/R Address(es) that correspond to a Directory Name.

5) Determine whether a Directory Name presented denotes a user or a Distribution List.

6) Return the members of a Distribution List.

7) Return the capabilities of the entity referred to by a Directory Name.

8) Maintenance functions to keep the directory up-to-date, e.g. register and change credentials.

b) MTA Specific Functionality:

1) Authentication.

2) Return the O/R Address(es) that correspond to a Directory Name.

3) Determine whether a Directory Name presented denotes a user or a Distribution List.

4) Return the members of a Distribution List.

5) Return the capabilities of the entity referred to by a Directory Name.

6) Maintenance functions to keep the directory up-to-date.

7) Search the Directory for entries containing a specified attribute type and value and return the Distinguished Names of the matching entries.

In addition to functionality, a number of operational aspects must be considered. These include user-friendliness, flexibility, availability, expandability and reliability.

9.4.4 Naming and Attributes

Since user-friendliness is of primary importance in a messaging system, the naming conventions used in building the Directory Information Tree (DIT) will impact the ability of a user to make intelligent guesses for Directory Names.

It is recommended that the naming guidelines and DIT structures defined in Annex B of Recommendation X.521/ISO 9594-7 be used as the basis for MHS Directory Names. Annex C of Recommendation X.402/ISO 10021-2 specifies further the MHS specific object classes. The naming for MHS specific object classes are recommended as follows:

a) The naming for mhs-message-store, mhs-message-transfer-agent, and mhs-user-agent is that of Application Entity in the DIT.

b) The naming attribute for mhs-distribution-list is commonName. The organization, organizationalUnit, organizationalRole, organizationalPerson, locality, or groupOfNames can be immediate superior to entries of object class mhs-distribution-list.

c) The naming for mhs-user is that of organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality.

NOTE - The mhs-user object class is a generic object class which may be used in conjunction with another standard object class for the purpose of adding MHS information attributes, such as ORAddresses, to a Directory entry. The means to associate attributes of a generic object class to an entry (or to different entries) named by a standard object class(es) is by defining a new (un-)registered object class, whose superclass(es) is that of the naming object class(es), and of the generic object class. E.g., to associate mhs-user attributes in the organizationalPerson entry, a new unregistered object class can be defined as shown in figure 8.

```
real-user-entry  ::=  OBJECT CLASS
                      SUBCLASS OF organizationalPerson,
                                  mhs-user
```

Figure 8 - Example of Unregistered Object Class Definition

The MHS object classes, attributes, and attribute syntaxes that need to be supported by the Directory are as specified in Annex C of Recommendation X.402/ISO 10021-2.

In addition, the object classes organization, organizationalUnit, organizationalRole, organizationalPerson, locality, groupOfNames, residentialPerson, and country and their attributes and associated syntaxes as defined in X.520 (ISO 9594, Part 6) and X.521 (ISO 9594, Part 7) are required to support the MHS.

9.4.5 Elements of Service

This clause specifies the requirements for support of Elements of Service for conformance to the Use of Directory Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in clause 5.2.

Support for Elements of Service is specified both for the MT Service (table 14) and for the IPM Service (table 15).

Table 14 - Use of Directory: MT Elements of Service

Element of Service	Origination	Reception	Relay
Designation of Recipient by Directory Name	M	M	-

Table 15 - Use of Directory: IPM Elements of Service

Element of Service	Origination	Reception
Designation of Recipient by Directory Name	M	-

9.4.6 Directory Services

These Implementors Agreements require the Directory services as defined in table 16. Indicated are the Directory services required to support the needs of the MHS UA/MTA and MHS Administrator.

Table 16 - Directory Service Support Requirements

Directory Service	MHS UA/MTA	MHS Admin
Bind and Unbind	M	M
Read	M	M
Compare	M	M
Abandon	M	M
List	M	M
Search	M	M
Add Entry		M
Remove Entry		M
Modify Entry	M	M
Modify RDN		M

9.4.7 OIW X.400 Base Directory Implementors Agreements

This clause defines the X.400 base Directory Implementors Agreements. Its structure and content are based on the Implementors Agreements template suggested in chapter 11.

9.4.7.1 Other Profiles Supported

The OIW X.400 Base Directory Implementors Agreements requires the support of OIW Directory Common Application Directory Implementors Agreements as defined in chapter 11.

9.4.7.2 Standard Application Specific Attributes and Attribute Sets

The standard application specific attributes and attributes sets supported by these Implementors Agreements are listed in table 17. For each attribute and attribute set, a reference is provided to the standard where it is defined.

Table 17 - Standard Attributes and Attribute Sets

Attribute / Attribute Set	References
mhs-deliverable-content-length	X.402/IS 10021-2
mhs-deliverable-content-types	X.402/IS 10021-2
mhs-deliverable-eits	X.402/IS 10021-2
mhs-dl-members	X.402/IS 10021-2
mhs-dl-submit-permissions	X.402/IS 10021-2
mhs-message-store	X.402/IS 10021-2
mhs-or-addresses	X.402/IS 10021-2
mhs-preferred-delivery-methods	X.402/IS 10021-2
mhs-supported-automatic-actions	X.402/IS 10021-2
mhs-supported-content-types	X.402/IS 10021-2
mhs-supported-optional-attributes	X.402/IS 10021-2

9.4.7.3 Standard Application Specific Object Classes

The standard application specific object classes supported by these Implementors Agreements are listed in table 18. For each object class, a reference is provided to the standard where it is defined.

Table 18 - Standard Object Classes

Object Class	References
mhs-distribution-list	X.402/IS 10021-2
mhs-message-store	X.402/IS 10021-2
mhs-message-transfer-agent	X.402/IS 10021-2
mhs-user	X.402/IS 10021-2
mhs-user-agent	X.402/IS 10021-2

9.4.7.4 OIW Application Specific Attributes and Attribute Sets

There are no application specific attributes or attribute sets defined by these Implementors Agreements.

9.4.7.5 OIW Application Specific Object Classes

There are no application specific object classes defined by these Implementors Agreements.

9.4.7.6 Structure Rules

This clause defines the naming and structure rules for the MHS object classes which are subclasses of top.

9.4.7.6.1 MHS Distribution List

Attribute commonName is used for naming.

Themhs-distribution-list, organization, organizationalUnit, organizationalRole, organizationalPerson, locality, or groupOfNames can be immediately superior to entries of object class mhs-distribution-list.

9.4.7.6.2 MHS User

The naming for mhs-user is that of organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality.

The organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality object classes can be combined with the mhs-user object class to form a new composite object class.

11 MHS Security

11.1 Overview

The secure functional group is specified as six security classes which are incremental subsets of the security features available in the base standard. They are denoted as S0, S0A, S1, S2A, S2B, and S3.

S0: This security class gathers together security functions applicable only between MTS-Users. Consequently, security mechanisms are implemented within the MTS-User. An MTA is required to support the syntax of the security services on submission, as the "Kernel" supports the syntax on relay and delivery. The MTA is not expected to understand the semantics of the security services.

S0A: This security class requires the services of class S0 and also mandates the support of content confidentiality.

S1: This security class is designed to meet the required assurance level of a domain and requires secure functionality with the MTS-User and MTS. The MTS secure functionality is only required to achieve secure access management. As with S0, most of the security mechanisms are implemented within an MTS-User.

It primarily provides integrity and authentication between MTS-Users. However, MTAs are expected to support digital signatures for peer to peer authentication, security labelling and security contexts.

S2A: This security class is a superset of S1, adding confidentiality when explicitly required, by encrypting the message content.

S2B: This security class is a superset of S1, adding security functional within MTAs and the MTS. The main security function added within this group is authentication within the MTS, and, as a consequence, due to the non-repudiable nature of the keys used for authentication, non-repudiation is also added.

S3: This security class is a superset of both S2A and S2B, and mandates the provision of the security functions of both of these security classes.

Symmetric or asymmetric techniques (or a combination thereof) may be used within each security class and are identified by the registered algorithm identifier.

Various levels of assurance in trusted COMPUSEC functionality may be used within each security class. This is outside the scope of this Implementors Agreement.

A full rationale for each of the security classes and a broader discussion of security considerations are provided in annex E.

Table 19 provides an overview of the requirements made by the security classes on the MTS-User and MTA. The table entries are descriptive, and are not intended to refer to security service elements.

Table 19 - Overview of Security Requirements for Each Security Class.

Class	Requirements	
	MTS-User	MTA
Kernel		Submission, delivery, and relay of EoS
S0	Content Integrity, Proof of Delivery, Message Origin Authentication (UA to UA)	Kernel
S0A	S0 plus Content Confidentiality	Kernel
S1	S0 plus Message security label, Message security context, Security Management Services	Peer entity authentication, Security context, Security Management Services, and Message Security Label
S2A	S1 plus Content confidentiality	S1
S2B	S1 plus Message Origin Authentication Check, Probe Origin Authentication Check, Report Origin Authentication Check, Proof of Submission, and, Non-repudiation	S1 plus Message Origin Authentication Check, Prove Origin Authentication Check, Report Origin Authentication Check, Proof of Submission, and, Non-repudiation
S3	S2A plus S2B	S2A plus S2B

The incremental functionality of the security classes can be represented diagrammatically as shown in figure 9.

```
S0  - S0A
|
S1  - S2A
|
S2B - S3
```

Figure 9 - Incremental Functionality of the Security Classes

11.2 Common Requirements

11.2.1 Interworking Between Security Classes

Interworking between implementations supporting different security classes can be achieved up to the highest common class supported. As specified in the base standard, the label of the message, probe or report must be checked against the security context by any implementation claiming conformance to classes S1, S2A, S2B, and S3.

> NOTE - Interworking can be limited to messages of only one security class by defining a security context consisting of labels with security policy identifiers of only that security class.

The security policy identifier in the message is used to indicate the security class in force. These security policy identifiers are defined in table 20.

Only the security policy object identifiers is required for conformance to this profile, but more specific policy identifiers may be registered for private secure interworking.

The security classes S1, S2A, S2B, and S3 use security labels and secure access management. These services may be used to facilitate secure interworking between the security classes. This concept may be used and extended within private secure domains, such domains may define their own secuity labels and register private policy identifiers. Private secure domains may define security classes which have no incremental or hierarchical relationship.

An alternative to registering private policy identifiers could be to use a particular value of security category to denote private extensions to the security class defined in this profile. This permits, for example, traffic over an ADMD using only the policy identifiers defined in this profile.

11.2.2 Comparision of Security Labels

The Security Context service is provided by comparing the message security label to the labels which make up the security context associated with the entity to which the message is to be transferred. Access is permitted if the message security label is dominated by at least one of the labels which make up the security context. To establish dominance, the Security Policy Identifier in the two labels must be present and must be the same. A message containing an unrecognized security policy identifier shall be rejected.

For the delivering and submitting MTAs only, the Security Classification and Security Category must also be compared to the security context, using rules established by the policy under which that MTA is operating.

The message security label may be placed in the per-message extensions or in the message token. Since the token is a per-recipient element (and the label may be encrypted within the token), the Security Context service and Origin Authentication services shall use the label (if any) specified in the per-message extensions. The label carried in the token has significance only between the originator and recipient UAs.

11.2.3 Application Context

Unless either a sufficiently reliable subnetwork is used, or transport class 4 is selected, it is strongly recommended that a "reliable" application context be used for P3 and P7 (i.e., using RTSE).

When providing the peer entity authentication service, it is recommended that MTAs should not use the "association-recovery" procedure of RTSE (section 7.8.3 of X.228). MTAs in the role of sender should not invoke this procedure and MTAs in the role of receiver should not accept RT-OPEN requests asking for recovery.

> **NOTE -** It is permissible for the sending MTA to perform the "activity resumption" (section 7.8.1 of X.228) on an existing, authenticated RTSE association owned by this MTA.

11.3 Description of Security Classes

The sections to follow describe the security classes within the secure functional group. For each security class, there is a description of the security functionalities provided, followed by a table which gives the classification for each of the security services required by that class. Where the classification of a security service does not change for a higher security class, then that security service is not repeated in the table for the higher security class.

Figure 10 explains the column headings used in the security class tables.

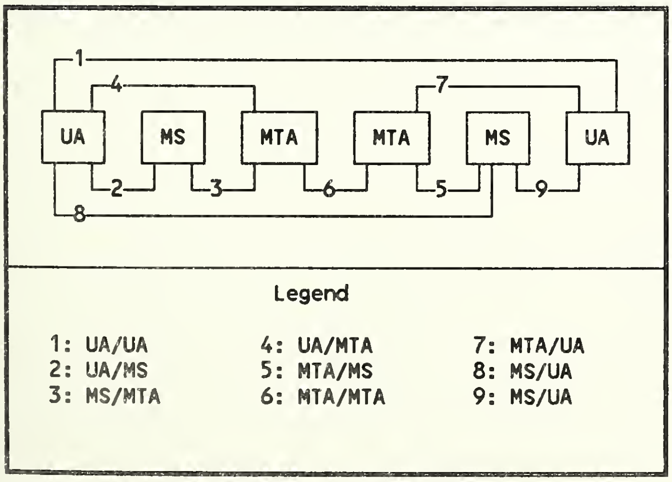


Figure 10 - Security Interfaces

11.4 Security Class 0 (S0)

11.4.1 Security Functionality

Security measures shall be provided by the MHS implementation in order to provide the following:

a) Integrity of message content;

b) Authentication of the MTS-User who originated the message;

c) Authentication of the MTS-User whom the message was delivered to.

This security class mandates the above services are provided by an MTS-User.

There are no requirements placed on the MTA.

11.4.2 Security Services for S0

Security class 0 (S0) mandates the security services listed table 20, which may be based on symmetric techniques, asymmetric techniques, or a combination of both.

Table 20 - Security Class 0 (S0)

Security Service	UA/ UA	UA/ MS	MS/ MTA	UA/ MTA	MTA/ MS	MTA/ MTA	MTA/ UA	MS/ UA	MS/ UA
Origin Authentication									
Message Origin Authentication[1]	M	I	-	I	-	-	-	-	-
Probe Origin Authentication	-	I[6]	-[6]	I	-	-	-	-	-
Report Origin Authentication	-	-	-	-	I	I	I	-	-
Proof of Submission	-	-	-	-	-	-	I	-	-
Proof of Delivery	M	-	-	-	-	-	-	M	-
Secure Access Management									
Peer Entity Authentication[2,7]	-	O	O	O	O	O	O	-	O
Security Context	-	O	O	O	O	O	O	-	O
Data Confidentiality Connection									
Connection Confidentiality[8]	-	I	I	I	I	I	I	-	I
Content Confidentiality	I	-	-	-	-	-	-	-	-
Message Flow Confidentiality	I	-	-	-	-	-	-	-	-
Data Integrity Services									
Connection Integrity[8]	-	I	I	I	I	I	I	-	I
Content Integrity	M	-	-	-	-	-	-	-	-
Message Sequence Integrity[4]	O	-	-	-	-	-	-	-	-
Non-Repudiation									
Non-Repudiation of Origin[1,5]	O	-	-	I	-	-	-	-	-
Non-Repudiation of Submission	-	-	-	-	-	-	I	-	-
Non-Repudiation of Delivery[5,10]	O	-	-	-	-	-	-	O[6]	-
Message Security Labelling[2,3]	O	O	O	O	O	O	O	O	O
Security Management Services									
Change Credentials	-	O	-	O	O	I[9]	O	-	-
Register	-	O	-	O	-	-	-	-	-
MS-Register	-	O	-	-	-	-	-	-	-
Data Confidentiality Connection									
Content Confidentiality	M	-	-	-	-	-	-	-	-

Table 20 - Security Class 0 (S0) (concluded)

Notes

1 Only provided to the message recipient.

2 Using either symmetric or asymmetric algorithms as identified by the algorithm identifier in the applicable protocol element.

3 When security labelling is used, the security policy identifier be included.

4 Allocation and management of sequence numbers is outside the of this Implementors Agreement (as it is subject to bilateral agreements).

5 Using either a trusted notory (symmetric) or using certificates tokens which are not repudiable (asymmetric).

6 Corrects the table 7 of X.402 in the base standard.

7 Authentication between indicated objects is a local issue.

8 Refer to section 10 of X.402 and IS 7498-2.

9 These services are expected to be provided by non-standard management services and are therefore outside the scope of this Implementors Agreement.

10 Non-Repudiation of Delivery can only be provided when the proof-of-delivery service is used.

11.5 Security Class 0A (S0A)

11.5.1 Security Functionality

Security measures shall be provided by the MHS Implementation in order to provide the following:

a) Security Functionality defined in security class S0; and,

b) Content Confidentiality.

11.5.2 Security Services for S0A

Security class 0A (S0A) mandates the security services of class S1 those listed in table 21, which may be based on symmetric or asmmetric techniques.

Table 21 - Security Class 0A (S0A)

Security Service	UA/ UA	UA/ MS	MS/ MTA	UA/ MTA	MTA/ MS	MTA/ MTA	MTA/ UA	MS/ UA	MS/ UA
Data Confidentiality Connection Content Confidentiality	M	-	-	-	-	-	-	-	-

11.6 Security Class 1 (S1)

11.6.1 Security Functionality

Security measures shall be provided by the MHS implementation in order to provide the following:

a) Authenticaton of MTA, MS, and UA;

b) Confidentiality of connections between MTA, MS, and UA;

c) Integrity of message content;

d) Authentication of message originator;

e) Authentication of message delivery (Proof of delivery);

f) MLS-features of MTA, MS, and UA;

g) MLS-separation of message probes and reports; and,

h) MLS-mediation by secure access measures.

NOTES

1 The level of assurance of the MLS trusted components is subject to bilateral agreement.

2 The level of accountability provided is subject to bilateral agreement.

11.6.2 Security Services for S1

Security class 1 (S1) mandates the security services listed in table 22, which may be based on asymmetric techniques, symmetric techniques, or a combination of both.

Table 22 - Security Class 1 (S1)

Security Service	UA/ UA	UA/ MS	MS/ MTA	UA/ MTA	MTA/ MS	MTA/ MTA	MTA/ UA	MS/ UA	MS/ UA
Origin Authentication									
Message Origin Authentication[2]	M[1]	I	-	I	-	-	-	-	-
Secure Access Management									
Peer Entity Authentication[3,4]	-	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	-	M[1]
Security Context	-	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	-	M[1]
Data Integrity Services									
Content Integrity	M[1]	-	-	-	-	-	-	-	-
Message Security Labelling[3]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]
Security Management Services									
Change Credentials	-	M	-	M	M	I[5]	M	-	-
Register	-	M	-	M	-	-	-	-	-
MS-Register	-	M	-	-	-	-	-	-	-

Notes

1 Shall always be used.

2 Only provided to the message recipient.

3 Using either symmetric or asymmetric algorithms as identified by the algorithm identifier in the applicabel protocol element.

4 Authentication between collocated objects is a local issue.

5 These services are expected to be provided by non-standard management services and are therefore outside the scope of this Implementors Agreement.

6 Corrects the corresponding table in the base standard (table 7 of X.402).

11.7 Security Class 2A (S2A)

11.7.1 Security Functionality

Security measures shall be provided by the MHS implementation in order to provide the following:

a) Security functionality defined for security class S1; and,

b) Content Confidentiality.

11.7.2 Security Services for S2A

Security class 2A (S2A) mandates the security services of class S1 those listed in table 23, which may be based on symmetric or asmmetric techniques.

Table 23 - Security Class 2A (S2A)

Security Service	UA/ UA	UA/ MS	MS/ MTA	UA/ MTA	MTA/ MS	MTA/ MTA	MTA/ UA	MS/ UA	MS/ UA
Data Confidentiality Connection Content Confidentiality	M	-	-	-	-	-	-	-	-

11.8 Security Class 2B (S2B)

11.8.1 Security Functionality

Security measures shall be provided by the MHS implementation in order to provide the following:

a) Security functionality defined for security class S1; and,

b) Plus authentication and non-repudiation of messages, probes, and reports.

11.8.2 Security Service for S2B

Security class 2B (S2B) mandates the security services of class S1 and those listed in table 24, based on asymmetric techniques.

Table 24 - Security Class 2B (S2B)

Security Service	UA/ UA	UA/ MS	MS/ MTA	UA/ MTA	MTA/ MS	MTA/ MTA	MTA/ UA	MS/ UA	MS/ UA
Origin Authentication									
Message Origin Authentication[3]	M[1]	M[1]	-	M[1]	-	-	-	-	-
Probe Origin Authentication	-	M[4]	-	M[1]	-	-	-	-	-
Report Origin Authentication	-	-	-	-	M[1]	M[1]	M[1]	-	-
Proof of Submission	-	-	-	-	-	-	-	M	-
Non-Repudiation									
Non-Repudiation of Origin	M[5]	-	-	M[2]	-	-	-	-	-
Non-Repudiation of Submission	-	-	-	-	-	-	M[2]	-	-
Non-Repudiation of Delivery	M[5]	-	-	-	-	-	-	M[2]	-

Notes

1 Shall always be used.

2 Using an asymmetric mechanism (i.e., certificates and tokens whic are not repudiable for authentication within MTAs and the MTS.

3 Using the Message Origin Authentication Check as detailed in the base standard.

4 Shall always be used, and corrects table 7 in X.402.

5 Using either a trusted notory (symmetric) or using certificates tokens which are not repudiable (asymmetric).

11.9 Security Class 3 (S3)

11.9.1 Security Functionality

Security measures shall be provided by the MHS implementation in order to provide the following:

a) Security functionality defined for security classes S2A and S2B.

11.9.2 Security Services for S3

Security class 3 (S3) mandates the services of classes S2A and S2B and are defined by the combination of tables 23 and 24.

A.5 Classification of the P1 Protocol Elements for Security Classes

The protocol element classifications used in tables 36 and 37 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 32. Thus, table 36 shows the additional support required in P1 to conform to security class S1. Table 37 indicates the additional support required to support security class S2B (above and beyond that for security class S1).

NOTES

1 There are no additional classifications for security class S0.

2 The addition of mandatory content confidentiality does not affect the P1 protocol.

Table 36 - Conformance Classification of the P1 Protocol Elements for Security Class S1

MTS Transfer Protocol (P1) for Security Class S1				Part 1 of 2
MT Kernel Static Support by MTS Class	B/C	A		
Protocol Element	O/R	O/R	Dyn	Comments/References
MTABind				
ARGUMENT				
<SET>				
initiator-credentials			M	
simple	O/O	O/O	X	
strong	M/M	M/M	M	
bind-token	M/M	M/M	M	
certificate	O/O	O/O		
security-context	M/M	M/M	M	
RESULT				
<SET>				
responder-credentials			M	
simple	O/O	O/O	X	
strong	M/M	M/M	M	
bind-token	M/M	M/M	M	
certificate	O/O	O/O		
MessageTransferEnvelope				
extensions				
message-security-label	M/M	M/M	M	
ReportTransferEnvelope				
extensions				
message-security-label	M/M	M/M	M	
per-recipient-fields				
extensions				
message-token	O/O	O/O	M	
asymmetric-token				
signature-algorithm-identifier	M/M	M/M	M	
name	M/M	M/M	M	
time	M/M	M/M	M	
signed-data			M	
message-security-label	M/M	M/M	M	
encryption-algorithm-identifier	M/M	M/M		
encrypted-data	M/M	M/M		
message-security-label	M/M	M/M		
content-integrity-key	M/M	M/M		

Table 36 - Conformance Classification of the P1 Protocol Elements for Security Class S1 (concluded)

MTS Transfer Protocol (P1) for Security Class S1				Part 2 of 2
MT Kernel Static Support by MTS Class				
Protocol Element	B/C O/R	A O/R	Dyn	Comments/References
bind-token				
asymmetric-token				See Note 1
signature-algorithm-identifier	M/M	M/M	M	
name	M/M	M/M	M	
time	M/M	M/M	M	
signed-data	M/M	M/M	M	
encryption-algorithm-identifier	M/M	M/M		
encrypted-data	M/M	M/M		
message-security-label	M/M	M/M		
content-integrity-key	M/M	M/M		
message-security-label				
security-policy-identifier	M/M	M/M	M	
security-classification	M/M	M/M		
privacy	O/O	O/O		
security-categories	M/M	M/M		

Notes

1 In line with the CCITT MHS Implementors' Guide, the asymetric token can be used by symetric and asymetric techniques as identified by the algorithm identifier.

Table 37 - Conformance Classification of the P1 Protocol Elements for Security Class S2B

MTS Transfer Protocol (P1) for Security Class S2B				Part 1 of 2
MT Kernel Static Support by MTS Class				
	B/C	A		
Protocol Element	O/R	O/R	Dyn	Comments/References
MessageTransferEnvelope				
extension				
originator-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
message-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
ProbeTransferEnvelope				
extensions				
originator-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
probe-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
ReportTransferEnvelope				
extensions				
reporting-MTA-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
report-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
per-recipient	M/M	M/M		
actual-recipient-name	M/M	M/M		
originally-intended-recipient-name	O/O	O/O		
delivery	O/O	O/O		
message-delivery-time	M/M	M/M		
type-of-MTS-user	M/M	M/M		
recipient-certificate	M/M	M/M		
proof-of-delivery	M/M	M/M		
non-delivery	O/O	O/O		
non-delivery-reason-code	M/M	M/M		
non-delivery-diagnostic-code	O/O	O/O		

Table 37 - Conformance Classification of the P1 Protocol Elements for Security Class S2B (concluded)

MTS Transfer Protocol (P1) for Security Class S2B				Part 2 of 2
MT Kernel Static Support by MTS Class				
Protocol Element	B/C O/R	A O/R	Dyn	Comments/References
Certificate				
version	M/M	M/M		
serialNumber	M/M	M/M		
signature	M/M	M/M		
algorithm	M/M	M/M		
parameters	O/O	O/O		
issurer	M/M	M/M		
validity	M/M	M/M		
notBefore	M/M	M/M		
notAfter	M/M	M/M		
subject	M/M	M/M		
subjectPublicKeyInfo	M/M	M/M		
algorithm	M/M	M/M		
subjectPublicKey	M/M	M/M		

A.6 Classification of the P3 Protocol Elements for Security Classes

The protocol element classifications in tables 38, 39, and 40 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 34. Thus, table 38 shows the additional support required in P3 to conform to security class S0. Table 39 indicates the additional support required to support security class S1 (above and beyond that for security class S0). Table 40 indicates the additional support required to support security class S2B (above and beyond that for security class S1).

> **NOTE** - There are no dynamic conformance classifications required by security class S0 (table 38).

Table 38 - Conformance Classification of the P3 Protocol Elements for Security Class S0

MTS Access Protocol (P3) for Security Class S0					Part 1 of 2
Static Support by: IPM					
	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MessageDelivery					
RESULT					
proof-of-delivery	M/-	M/-	-/O		
MessageSubmissionEnvelope					
PerRecipientMessageSubmission Fields					
extensions					
message-token	M/-	M/-	-/O		
asymmetric-token	M/-	M/-	-/O		
signature-algorithm-identifier	M/-	M/-	-/O		
name	M/-	M/-	-/O		
time	M/-	M/-	-/O		
signed-data	M/-	M/-	-/O		
content-confidentiality-algorithm-identifier	O/-	O/-	-/O		
content-integrity-check	M/-	M/-	-/O		
message-security-label	O/-	O/-	-/O		
proof-of-delivery-request	M/-	M/-	-/O		
message-sequence-number	O/-	O/-	-/O		
encrpytion-algorithm-identifier	O/-	O/-	-/O		
encrypted-data	O/-	O/-	-/O		
content-confidentiality-key	O/-	O/-	-/O		
content-integrity-check	M/-	M/-	-/O		
message-security-label	O/-	O/-	-/O		
content-integrity-key	O/-	O/-	-/O		
message-sequence-number	M/-	M/-	-/O		
content-integrity-check	M/-	M/-	-/O		
proof-of-delivery-request	M/-	M/-	-/O		

Table 38 - Conformance Classification of the P3 Protocol Elements for Security Class S0 (concluded)

MTS Access Protocol (P3) for Security Class S0					Part 2 of 2
Static Support by: IPM	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MessageDeliveryEnvelope					
other-fields					
extensions					
message-token	-/M	-/M	O/-		
asymmetric-token	-/M	-/M	O/-		
signature-algorithm-identifier	-/M	-/M	O/-		
name	-/M	-/M	O/-		
time	-/M	-/M	O/-		
signed-data	-/M	-/M	O/-		
content-confidentiality-algorithm-identifier	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		
message-security-label	-/O	-/O	O/-		
proof-of-delivery-request	-/M	-/M	O/-		
message-sequence-number	-/O	-/O	O/-		
encrpytion-algorithm-identifier	-/O	-/O	O/-		
encrypted-data	-/O	-/O	O/-		
content-confidentiality-key	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		
message-security-label	-/O	-/O	O/-		
content-integrity-key	-/O	-/O	O/-		
message-sequence-number	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		
proof-of-delivery-request	-/M	-/M	O/-		
ReportDeliveryEnvelope					
PerRecipientReportDeliveryFields					
extensions					
proof-of-delivery	-/M	-/O	O/-		

Table 39 - Conformance Classification of the P3 Protocol Elements for Security Class S1

MTS Access Protocol (P3) for Security Class S1					Part 1 of 3
Static Support by:	IPM				
	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MTSBind					MTS to MTS User
ARGUMENT					
initiator-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
security-context	-/M	-/M	M/-	M	
RESULT					
responder-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
MTSBind					MTS User to MTS
ARGUMENT					
initiator-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
security-context	M/-	M/-	-/M	M	
RESULT					
responder-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
SubmissionControl	-/M	M/M	M/-		
ARGUMENT					
controls					
permissible-security-context	-/M	-/M	M/-		
DeliveryControl	M/-	M/-	-/M		
ARGUMENT					
controls					
permissible-security-context	M/-	M/-	-/M		
Register					
ARGUMENT					
user-name	M/-	M/-	-/M		
labels-and-redirections					
user-security-label	M/-	M/-	-/M		

Table 39 - Conformance Classification of the P3 Protocol Elements for Security Class S1 (continued)

MTS Access Protocol (P3) for Security Class S1					Part 2 of 3
Static Support by: IPM	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
ChangeCredentials					MTS to MTSuser
ARGUMENT					
old-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
new-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
ChangeCredentials					MTSuser to MTS
ARGUMENT					
old-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
new-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
MessageSubmissionEnvelope					
extensions					
message-token	M/-	M/-	-/M	M	
content-integrity-check	M/-	M/-	-/M	M	
message-security-label	M/-	M/-	-/M	M	
MessageDeliveryEnvelope					
extensions					
message-security-label	-/M	-/M	M/-		
ReportDeliveryEnvelope					
extensions					
message-security-label	-/M	-/M	M/-	M	

Table 39 - Conformance Classification of the P3 Protocol Elements for Security Class S1 (concluded)

MTS Access Protocol (P3) for Security Class S1					Part 3 of 3
Static Support by:	IPM UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
bind-token					
asymmetric-token					
signature-algorithm-identifier	-/M	-/M	M/-	M	
name	-/M	-/M	M/-	M	
time	-/M	-/M	M/-	M	
signed-data	-/M	-/M	M/-	M	
encryption-algorithm-identifier	-/M	-/M	M/-		
encrypted-data	-/M	-/M	M/-		
message-security-label	-/M	-/M	M/-		
content-integrity-key	-/M	-/M	M/-		
message-token					
asymmetric-token					
signature-algorithm-identifier	M/-	M/-	-/M	M	
name	M/-	M/-	-/M	M	
time	M/-	M/-	-/M	M	
signed-data				M	
message-security-label	M/-	M/-	-/M	M	
content-integrity-check	M/-	M/-	-/M	M	
encryption-algorithm-identifier	M/-	M/-	-/M		
encrypted-data	M/-	M/-	-/M		
message-security-label	M/-	M/-	-/M		
content-integrity-key	M/-	M/-	-/M		
message-security-label					
security-policy-identifier	M/-	M/-	-/M	M	
security-classification	M/-	M/-	-/M		
security-categories	M/-	M/-	-/M		

Table 40 - Conformance Classification of the P3 Protocol Elements for Security Class S2B

MTS Access Protocol (P3) for Security Class S2B					Part 1 of 2
Static Support by: IPM	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MessageSubmission					
RESULT					
extensions					
originating-MTA-certificate	-/M	-/O	M/-		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
proof-of-submission	-/M	-/O	M/-		
MessageDelivery					
RESULT					
recipient-certificate	M/-	M/-	-/O		
certificate	M/-	M/-	-/M		
certification-path	M/-	M/-	-/M		
MessageSubmissionEnvelope					
extensions					
originator-certificate	M/-	O/-	-/M		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
message-origin-authentication-check	M/-	O/-	-/M	M	
algorithm-identifier	M/-	M/-	-/M		
content	M/-	M/-	-/M		
content-identifier	M/-	M/-	-/M		
message-security-label	M/-	M/-	-/M		
proof-of-submission-request	M/-	O/-	-/M		
ProbeSubmissionEnvelope					
extensions					
originator-certificate	M/-	O/-	-/M		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
probe-origin-authentication-check	M/-	O/-	-/M	M	
algorithm-identifier	M/-	M/-	-/M		
content-identifier	M/-	M/-	-/M		
message-security-label	M/-	M/-	-/M		

Table 40 - Conformance Classification of the P3 Protocol Elements for Security Class S2B (concluded)

MTS Access Protocol (P3) for Security Class S2B					Part 2 of 2
Static Support by: IPM	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MessageDeliveryEnvelope					
extensions					
originator-certificate	-/M	-/M	M/-		
certificate	-/M	-/M	M/-		
certification-path	-/M	-/M	M/-		
message-origin-authentication-check	-/M	-/M	M/-	M	
algorithm-identifier	-/M	-/M	M/-		
content	-/M	-/M	M/-		
content-identifier	-/M	-/M	M/-		
message-security-label	-/M	-/M	M/-		
ReportDeliveryEnvelope					
extensions					
reporting-MTA-certificate	-/M	-/O	M/-		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
report-origin-authentication-check	-/M	-/O	M/-		
PerRecipientReportDelivery-Fields					
extensions					
recipient-certificate	-/M	-/M	O/-		
certificate	-/M	-/M	M/-		
certification-path	-/M	-/M	M/-		
Certificate					
version	-/M	-/M	M/-		
serialNumber	-/M	-/M	M/-		
signature	-/M	-/M	M/-		
algorithm	-/M	-/M	M/-		
parameters	-/O	-/O	O/-		
issurer	-/M	-/M	M/-		
validity	-/M	-/M	M/-		
notBefore	-/M	-/M	M/-		
notAfter	-/M	-/M	M/-		
subject	-/M	-/M	M/-		
subjectPublicKeyInfo	-/M	-/M	M/-		
algorithm	-/M	-/M	M/-		
subjectPublicKey	-/M	-/M	M/-		

Table 41 presents the classification delta to classification tables 38, 3.39, and 40, for the addition of mandatory content confidentiality in the static conformance classification.

> NOTE - There are no dynamic conformance classification required by the addition of content confidentiality.

Table 41 - Conformance Classification of the P3 Protocol Elements for Security Classes S0A, S2A, or S3

MTS Access Protocol (P3) for Security Classes S0A, S2A, S3					Part 1 of 1
Static Support by:	IPM				
	UA	MS	MTA		
Protocol Element	O/R	O/R	O/R	Dyn	Comments/References
MessageSubmissionEnvelope					
extensions					
content-confidentiality-algorithm-identifier	M/-	O/-	-/O		
message-token					
asymmetric-token					
signed-data	M/-	-/-	-/-		
content-confidentiality-algorithm-identifier	M/-	-/-	-/-		
encrypted-data					
content-confidentiality-key	M/-	-/-	-/-		
MessageDeliveryEnvelope					
extensions					
message-token	-/M	-/M	O/-		
asymmetric-token					
signed-data	-/M	-/M	-/-		
content-confidentiality-algorithm-identifier	-/M	-/M	-/-		
encrypted-data					
content-confidentiality-key	-/M	-/M	-/-		
content-confidentiality-algorithm-identifier	-/M	-/M	O/-		

A.7 Classification of the P7 Protocol Elements for Security Classes

The protocol element classifications in table 41 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 35. Thus, table 41 shows the additional support required in P7 to conform to security class S1.

NOTES

1 There are no additional classifications for security classes S0 and S2B.

2 The addition of mandatory content confidentiality does not affect the P7 protocol.

Table 42 - Conformance Classification of the P7 Protocol Elements for Security Class S1

MS Access Protocol (P7) for Security Class S1				Part 1 of 1
Static Support by:	IPM			
	UA	MS		
Protocol Element	O/R	O/R	Dyn	Comments/References
MSBind				
ARGUMENT				
initiator-credentials			M	
simple	O/-	-/O	X	
strong	M/-	-/M	M	
bind-token	M/-	-/M	M	
certificate	O/-	-/O		
security-context	M/-	-/M	M	
RESULT				
responder-credentials			M	
simple	-/O	O/-	X	
strong	-/M	M/-	M	
bind-token	-/M	M/-	M	
certificate	-/O	O/-		
Register-MS				
ARGUMENT				
Register-MSArgument				
changeCredentials			M	
old-credentials	M/-	-/M	M	
simple	O/-	-/O	M	
strong	M/-	-/M	X	
bind-token	M/-	-/M	M	
certificate	O/-	-/O		
new-credentials	M/-	-/M	M	
simple	O/-	-/O	X	
strong	M/-	-/M	M	
bind-token	M/-	-/M	M	
certificate	O/-	-/O		
user-security-labels	M/-	-/M	M	
message-security-label				
security-policy-identifier	M/-	-/M	M	
security-classification	M/-	-/M		
privacy	O/-	-/O		
security-categories	M/-	-/M		

A.9 Classification of the IPM MS General Attributes for Security Classes

The classification of the attributes in table 44 is a delta to the MS General Attributes classified in table 36. Table 44 indicates the additional attributes that must be supported in the IPM MS for each of the security classes. There is no support required for security attributes in the basic MS.

Table 44 - IPM MS Security Attribute Support

Attribute	Security Class				
	S0	S0A	S1	S2A	S2B
content-confidentiality-algrithm-identifier	O	M	O	M	O
content-integrity-check	M	M	M	M	M
message-security-label	O	O	M	M	M
message-origin-authentication-check	M	M	M	M	M
message-token	M	M	M	M	M
origination-certificate	O	O	O	O	M
proof-of-delivery	M	M	M	M	M
reporting-mta-certificate	O	O	O	O	M
report-origin-authentication-check	O	O	O	O	M
security-classification	O	O	M	M	M

Annex C (informative)
Recommended Practices

C.6 Use of Externally Defined Body Part

The Externally Defined Body Part allows the identification of various types of information exchanged between UAs. This capability allows a UA on reception to recognize the appropriate application to process the information, e.g., to automatically invoke the application. It also allows the MTS to possibly convert from one type to another (e.g., to convert a word-processing document to ODA). For example, a spreadsheet application on a personal computer from vendor A by operating system X can send via X.400 to the UA on workstation from vendor B using operating system Y, and the latter will be able to recognize and possible invoke the appropriate spreadsheet application to process the data.

The Externally Defined Body Part definition is reproduced in figure 18.

```
ExternallyDefinedBodyPart   ::= SEQUENCE {
  parameters                    [0] ExternallyDefinedParameters  OPTIONAL,
  data                              ExternallyDefinedData  }

ExternallyDefinedParameters ::= EXTERNAL
ExternallyDefinedData       ::= EXTERNAL

EXTERNAL                    ::= [UNIVERSAL 8]  IMPLICIT SEQUENCE  {
  direct-reference              OBJECT IDENTIFIER  OPTIONAL,
  indirect-reference            INTEGER  OPTIONAL,
  data-value-descriptor         ObjectDescriptor  OPTIONAL,
  encoding                      CHOICE  {
    single-ASN1-type              [0]  ANY,
    octet-aligned                 [1]  IMPLICIT OCTET STRING,
    arbitrary                     [2]  IMPLICIT BIT STRING  }  }
```

Note - In the case of transfer of EXTERNAL in P2 BodyPart, the direct-reference component is mandatory and the indirect-reference and data-value-descriptor components must be absent.

Figure 18 - Externally Defined Body Part Definition

The following recommends how to use the components of this BodyPart:

a) *Use of Parameters component:* In simple cases, to ease the integration of applications to X.400 systems, the Parameters component need not be used.

b) *Use of Data component:* For each different format of data, different Object Identifiers for the Data component are recommended. If an application chooses to use ASN.1 to format the data to achieve a single representation across platforms, the single-ASN1-type encoding choice should be used. Otherwise:

1) The octet- (i.e., byte) aligned choice is used if the data format is octet-aligned; or,

2) The arbitrary choice is used if the data is bit-aligned.

c) *Assignment of Object Identifiers:* Object Identifiers need to be assigned for the EXTERNALs, and these identifiers for the Parameters and Data components should be different. The Object Identifier for an EXTERNAL also indicates the syntax of the data encoding, i.e., whether single-ASN1-type or octet-aligned or bit-aligned is being used.

NOTE - Whether the Object Identifier needs to be present in this case is for further study.

There are many ways to obtain object identifiers. One such way is described as follows:

a) The application provider obtains a unique Numeric Name form for their organization from ANSI, as described in ANSI ISSB 840 and ISSB 843, and appends this number form to {iso (1) member-body (2) US (840)} to form an object identifier denoting their organization.

b) The application provider (organization) allocates a series of numbers to identify the application data format; these numbers are appended to the object identifier constructed in step (i) to form an object identifier that is globally unique. It is recommended that the application provider (orgainization) use a hierarchical structure for identifying their data types to ease the administration of the identifiers.

For example, company PCSoftware Inc. obtains the organization number "999" from ANSI. The PCSoftware SpreadSheet file for MS-DOS might be assigned the following object identifier.

NOTE - ASN.1 notation is used. The numbers in parentheses form the identifier, the associated words describe the number.

{ iso (1) member-body (2) US (840) PCSoftware Inc. (999) MS-DOS-Application (1) SpreadSheet (3) Data (1) }

Since the SpreadSheet data format for MS-DOS does not follow ASN.1 encoding rules and is octet-aligned, the encoding choice for the Data EXTERNAL is octet-aligned.

It is also recognized that not all objects will be registered as indicated above. In order to transfer unregistered data, It is recommended that they be transferred in the BilaterallyDefinedBodyPart.

It is recommended that a 1988 system be able to support the Undefined Body Part (14), if it also supports Extended Body Parts. This is in order to facilitate inter-working with 1984 systems.

Annex D (informative)
List of ASN.1 Object Identifiers

D.3 Security Classes

The ASN.1 expressed in figure 19 defines the security Object Identifiers specified by these Implementation Agreements.

```
id-mhs-security              OBJECT IDENTIFIER ::= { iso (1)
  identified-organization (3) ewos (16) eg (2) mhs (4) security (4) }

id-policy-id                 OBJECT IDENTIFIER ::= { id-mhs-security 1 }
id-security-id               OBJECT IDENTIFIER ::= { id-mhs-security 2 }

-- Security Policy Object Identifiers --
-- Same as EWOS/ETSI --

security-class-0             OBJECT IDENTIFIER ::= { id-policy-id 0 }
security-class-0A            OBJECT IDENTIFIER ::= { id-policy-id 1 }
security-class-1             OBJECT IDENTIFIER ::= { id-policy-id 2 }
security-class-2A            OBJECT IDENTIFIER ::= { id-policy-id 3 }
security-class-2B            OBJECT IDENTIFIER ::= { id-policy-id 4 }
security-class-3             OBJECT IDENTIFIER ::= { id-policy-id 5 }

-- Security Category Object Identifiers --
-- Same as EWOS/ETSI --

private-id                   OBJECT IDENTIFIER ::= { id-security-id 0 }
confidence-id                OBJECT IDENTIFIER ::= { id-security-id 1 }
commercial-in-confidence-id  OBJECT IDENTIFIER ::= { id-security-id 2 }
management-in-confidence-id  OBJECT IDENTIFIER ::= { id-security-id 3 }
personal-in-confidence-id    OBJECT IDENTIFIER ::= { id-security-id 4 }
```

Figure 19 - Security Object Identifiers

Annex E (informative)
Secure Messaging Guidelines

E.1 Introduction

The purpose of the security functional group is to define an approach to the provision of secure messaging with Message Handling systems within the general framework of these Implementation Agreements.

E.2 Message Handling Vulnerabilities

The message handling vulnerabilities (threats) which can be protected using security measures are defined in Annex D (Security Threats) to Recommendation X.402 (1988):

a) Masquerading

b) Message sequencing

c) Modification of information

d) Denial of service

e) Repudiation

f) Leakage of information

Other specific threats exist if there is a failure to maintain information separation, which includes:

a) Manipulation

b) Misrouting

Some of these threats are defined in ISO standard IS 7498, OSI Reference Model, Part 2: Security Architecture. The ISO standard also specifies other threats, not all are relevant to message handling systems.

Annex D to CCITT Recommendation X.402 (1988) also indicates which MHS security services may provide protection against such threats.

Some threats to message handling systems cannot be easily prevented, merely detected, others are not appropriated for standardization.

E.3 General Principles

E.3.1 Security Policy

A general security policy can be defined as the set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. Thus a security policy defines an organization's overall approach to security and must cover all security aspects.

Security within an organization is not only the concern of message handling service and must be viewed in a more global and general sense. The wider aspects of a security policy would therefore include personnel security (such as the vesting and confidence placed in staff), end-user access control, physical, procedural and documentation security. These Implementors Agreements however is only concerned with Electronic Information Security (EIS), specifically in the areas of communications (COMSEC) and computer (COMPUSEC) as applicable to standardization of secure message handling system operating in a store and forward environment.

E.3.2 Security Classes

In the X.400 (1988) Recommendations, some threats are countered by EIS measures, these measures are realized by providing security services and implemented using security elements.

These Implementors Agreements groups together security features of a message handling system defined by the base standards into separate classes. A security class can be viewed as a tool which can be used to implement a security policy, and is not a security policy on its own right but a component of a security policy.

These Implementors Agreements includes a set of security classes, each class stipulates a set of mandatory and optional security services.

Mandatory security services within a security class can be selected by the subscriber or user, either on a per-message basis, or for an agreed contractual period of time. It is a local issue to determine when a mandatory security service is offered for user selection or when it is permanently invoked. Facilities and mechanisms to support the mandatory security services must always be provided within the security class, specifying the services as "mandatory".

E.3.3 Dynamic Behavior Requirements

The use of some security services is always required for certain security classes. This is specified in these Implementors Agreements by imposing additional dynamic requirements, to those specified in the base standards, ensuring that the corresponding protocol elements are always present.

Similarly, use of some security services are prohibited for certain security classes. This is specified in this profile by imposing additional dynamic requirements to those specified in the base standards, ensuring that the protocol elements are never present.

E.3.4 Encryption Techniques

The secure messaging facilities defined in the base standards are provided using three basic security techniques, namely:

a) Symmetric encryption

b) Asymmetric encryption

c) Trusted functionality (i.e., COMPUSEC measures).

The base standards permit the use of the techniques on an individual basis to provided security services or they can be combined in support of a security policy. This profile combines the techniques in order to provide a comprehensive set of security facilities, which are intended to counter various combinations of the vulnerabilities of a messaging service. In some cases the security services defined in the base standards can only be implemented using one of the techniques above, namely asymmetric encryption. However, the actual technique employed shall be dependent on the algorithms, and provided a security authority registers the algorithms for the domain.

It is the intention of these Implementors Agreements that implementations will not be restricted to asymmetric techniques. All the mandatory security services can be implemented using trusted functionality in combination with symmetric, asymmetric, or both the encryption techniques.

Although the base standards defines the syntax of an asymmetric token, these Implementors Agreements takes into account the ISO/CCITT Implementors' Guide, which permits the use of both asymmetric and symmetric techniques for both the signed and encrypted data.

Algorithms are assumed to be bilaterally agreed or registered by a registration authority. However, the algorithm-identifier must be unique and unambiguously define the algorithm.

E.3.5 Implementation Considerations

E.3.5.1 Peer Entity Authentication

Peer entity authentication is provided using the strong authentication mechanisms on the various Bind operations, using either asymmetric or symmetric techniques. The key management information necessary for symmetric peer entity authentication is outside the scope of these Implementors Agreements.

E.3.5.2 Confidentiality

Connection confidentiality is provided using the underlying OSI layers and is outside the scope of these Implementors Agreements. Mechanisms to support connection confidentiality are subject to bilateral agreement between peers (i.e., connection confidentiality may even be achieved by trusting the peer OSI connection).

Content Confidentiality may be achieved by either symmetric or asymmetric encryption techniques. It should be noted that use of asymmetric techniques precludes submission of messages to multiple recipients.

E.3.5.3 Integrity

Connection Integrity is provided using the underlying OSI layers and is outside the scope of these Implementors Agreements. Mechanisms to support Connection Integrity are subject to bilateral agreement between peers. It should be noted that the integrity of a connection can be increased by use of RTSE.

Content Integrity is achieved by computing a content integrity check as a function of the entire message content. When symmetric techniques are used to compute the content integrity check a secret key is required. This content integrity key may be confidentially sent to the message recipient using the message argument confidentiality security element in the message token (i.e., there may be other keys or parts of the key not sent by the originator with the message, but the key management of such external keys are outside the scope of these Implementors Agreements).

> **NOTE** - Content Integrity can also provide integrity of receipt and non-receipt notifications (IPNs) and can assist in the provision of "non-repudiation of receipt" since non-repudiation of delivery may be insufficient where delivery is to a Message Store.

E.3.5.4 Message Origin Authentication

End-to-end (i.e., UA to UA) Message Origin Authentication is automatically provided by content integrity. Security classes S2B and S3 provide additional protection by requiring support of origin authentication checks within the MTS.

E.3.5.5 Non-Repudiation

If asymmetric techniques are used for content integrity it can also provide non-repudiation of origin (UA to UA) depending on the level of trust placed in the certificate. If symmetric techniques are used, content integrity can also provide non-repudiation of origin, but only by using a trusted notary to validate the content integrity and provide trusted ...?

E.3.5.6 Secure Access Management

Secure Access Management can be implemented by a combination of Multi-Level Security (MLS) functionality by assurance of the various MHS components to support such functionality. MLS functionality is supported in the base standards by the use of security labels, security context and the security token and can be applied in a hierarchical and/or role manner depending on the policy requirements of a domain.

MLS assurance will generally also require other (COMPUSEC) measures and is outside the scope of the base standards and these Implementors Agreements. Reference should be made to the appropriate security authority and any applicable security evaluation criteria (e.g., U. S. DoD Orange Book, UK - Netherlands - Germany - France draft evaluation criteria).

E.3.5.7 Implications for the Use of Distribution Lists

Since the Distribution List expansion point must create all the per-recipient fields for the members of the Distribution List, it shall have the same security class as the originator and must have trusted functionality.

E.3.5.8 Implications on Redirection

The Security Functional Group has the effect of either requiring trust in any redirection facilities or prohibiting the use of redirection. If the Redirection facility is to be trusted, it must be subject to the security policy and obey the security labels as defined in the base standards.

E.3.5.9 Implications for 1984 Interworking

Interworking between implementations conforming to Security Functional Groups and 1984 systems is not supported. The Double Enveloping technique can be used to traverse an 1984 system.

E.3.5.10 Implications for Use of Directory

The X.400 security services use of the directory service does not require a trusted directory because the information that is retrieved is certified and can be validated independently of the directory.

Other threats (e.g., malicious corruption of directory information) may ?? from the broader use of the directory, however these are outside of the scope of the X.400 base standard and this Implementors Agreement.

Work continues within CCITT and ISO to impreve the security inherent in the Directory standards.

E.3.5.11 Implications for Conversion

Implementation of the Security Functional Group may additionally either require that any conversion facilities are trusted and can regenerate security elements (notably content integrity) or prohibit the use of conversion within the MTS altogether.

E.3.5.12 Accountability

Accountability depends on the identification and authentication of users, then subsequent records being kept on the actions taken by users. Therefore, accountability depends on all the relevant information being properly stored on recorded.

Accountability features provided by domains (or MTAs) are subject to bilateral agreement between domains (or MTAs) and may optionally provide non-repudiation services. Accountability features include pervasive mechanisms such as security logs, audit trails and archives, or they may be mechanisms supported by protocol. Protocol to support accountability will be subject to bilateral agreement.

E.3.5.13 Double Enveloping

Double enveloping can be used with each secure messaging security class. For each security class it is an optional extension to the security features which can be used to counter specific vulnerabilities. When double enveloping is used it shall be applied at the boundary of a domain, and obey the rules of an MTA at management domain boundaries. Figure 20 illustrates the technique.

```
OUTER ENVELOPE [envelope 2][inner envelope]
INNER ENVELOPE [envelope 1][content 1]
```

Figure 20 - Double Enveloping Technique

Address information in envelope 1 and 2 are not necessarily the same.

Trace information in envelope 1 and 2 are not necessarily the same.

The double envelope technique can be used in 1984 and 1988 MTS environments. When used in an 1988 environment, any security class can be applied to the outer envelope. It is recommended that the inner envelope is encrypted. When the double envelope technique is used as a secure relay path via an 1984 domain, any encryption of the outer content is subject to bilateral agreement.

Trace information is not passed between inner and outer envelopes. It is recommended that trace information on the outer envelope is always archived when the double envelope technique is used.

E.4 Security Class S0

E.4.1 Rationale

Security class S0 is confined to security functionality operating between MTS-Users on an end-to-end basis. It is designed to minimize the required functionality in the MTS to support submission of elements associated with these services. Security services which must be supported (i.e., must be made available) are those which are considered in any secure messaging environment, i.e.:

a) Content Integrity

b) Message Origin Authentication (end-to-end)

c) Proof of Delivery.

Other security services, such as Content Confidentiality, may optionally be supported.

E.4.2 Technical Implications

The technical implications for security class S0 are:

a) It is necessary to provide mechanisms in a UA which can generate the signed, signature and encrypted macros on message submission; and,

b) It is necessary to provide mechanisms in a UA which can handle the signed , signature and encrypted macros on message delivery.

E.5 Security Class S1

E.5.1 Rationale

The S1 security class is a superset of security class S0 and introduces the basic requirement for security functionality not only within the MTS-User but also within the MTS. This security functionality within the MTS is designed to support the enforcement of a security policy within a security domain. As a consequence, S1 enables trusted routing to be implemented.

NOTE - The level of trust in the route will depend on the level of trust in the security label and security context.

E.5.2 Technical Implications

The technical implications of security class S1 are:

a) It is necessary to provide mechanisms in a UA which can generate the signed, signature and encrypted macros on message submission.

b) It is necessary to provide mechanisms in a UA which can handle the signed, signature and encrypted macros on message delivery.

c) It is necessary to provide mechanisms in the MTA for registration, change-credentials and bind abstract operations (i.e., signed macro for bind).

d) It is necessary to provide mechanisms in the MS for MS-registration and MS-bind operation (i.e., signed macro for MS-Bind).

e) It is necessary to support message security labelling (the level of assurance is subject to individual security domain requirements).

f) It is necessary that reliable access is always supported.

g) It is necessary for the MTAs to check the existence of the security elements which are classified as "dynamic mandatory".

h) It is necessary to provide a trusted connection between peers to provide adequate confidentiality, integrity and peer entity authentication.

E.6 Security Class S2A

E.6.1 Rationale

Security class S2A is a superset of security class S1 and adds the requirement for support of end-to-end content confidentiality. The rationale for not requiring such support in security classes S0 and S1 is the processing overhead in encrypting the message content and reducing the complexity of the implementation.

The rationale why this is a superset of security class S1 is because the encryption techniques and mechanisms (i.e., algorithms, key lengths, key versions, etc.) are protected by Secure Access Management.

E.6.2 Technical Implications

The technical implications of security class S2A are the same as in security class S1, plus:

a) It is necessary to provide mechanisms in end systems which can use the encrypted macros for encryption and decryption the message content.

E.7 Security Class S2B

E.7.1 Rationale

Editor's Note - Text is missing for this section.

E.7.2 Technical Implications

Editor's Note - Text is missing for this section.

a) Connection confidentiality is only provided by this security class when the message-origin-authentication-check is computed using clear content to provide non-repudiation of origin security service (i.e., non-repudiation is provided only on the clear content of the message).

b) The irrevocable proof required to provide non-repudiation within the MTS is achieved by the management of asymmetric keys. The explicit definition of asymmetric key management is outside the scope of these Implementors Agreements.

E.8 Security Class S3

E.8.1 Rationale

Security class S3 is the sum of security classes S2A and S2B, providing support for all the security features of the base standards.

E.8.2 Technical Implications

The technical implications of security class S3 are as defined for both security classes S2A and S2B.

Table of Contents

9 STABLE FTAM PHASE 2

Editor's Note: For Stable FTAM Phase 2 Agreements, consult the aligned section in the Stable Implementation Agreements Document. This section serves as a reference or pointer to Stable Agreements contained in Version 3 dated September 1990.

Table of Contents

10 ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 3

Editor's Note: For current Stable FTAM Phase 3 Agreements, consult the aligned section in the Stable Implementation Agreements, Version 3 dated September 1990.

Table of Contents

11: DIRECTORY SERVICES PROTOCOLS

1 Introduction

Refer to subclause 11.1 of Stable Agreements Version 3 as of September 14, 1990.

2 Scope and Field of Application

Refer to subclause 11.2 of Stable Agreements Version 3 as of September 14, 1990.

3 Status

Refer to subclause 11.3 of Stable Agreements Version 3 as of September 14, 1990.

4 Use of the Directory

This clause will contain introductory text.

4.1 MHS

(TBD)

4.2 FTAM

(TBD)

5 Directory ASEs and Application Contexts

Refer to subclause 11.5 of Stable Agreements Version 3 as of September 14, 1990.

6 Schema

Refer to subclause 11.6 of Stable Agreements Version 3 as of September 14, 1990.

6.1 Support of Structures and Naming Rules

Refer to subclause 11.6.1 of Stable Agreements Version 3 as of September 14, 1990.

6.2 Support of Object Classes and Subclasses

Refer to subclause 11.6.2 of Stable Agreements Version 3 as of September 14, 1990.

6.3 Support of Attribute Types

Refer to subclause 11.6.3 of Stable Agreements Version 3 as of September 14, 1990.

6.4 Support of Attribute Syntaxes

Refer to subclause 11.6.4 of Stable Agreements Version 3 as of September 14, 1990.

6.5 Naming Contexts

Refer to subclause 11.6.5 of Stable Agreements Version 3 as of September 14, 1990.

6.6 Common Profiles

Refer to subclause 11.6.6 of Stable Agreements Version 3 as of September 14, 1990.

6.6.1 OIW Directory Common Application Directory Profile

Refer to subclause 11.6.6.1 of Stable Agreements Version 3 as of September 14, 1990.

6.6.1.1 Standard Application Specific Attributes and Attribute Sets

Refer to subclause 11.6.6.1.1 of Stable Agreements Version 3 as of September 14, 1990.

6.6.1.2 Standard Application Specific Object Classes

Refer to subclause 11.6.6.1.2 of Stable Agreements Version 3 as of September 14, 1990.

6.6.2 OIW Directory Strong Authentication Directory Profile

Refer to subclause 11.6.6.2 of Stable Agreements Version 3 as of September 14, 1990.

6.6.2.1 Other Profiles Supported

Refer to subclause 11.6.6.2.1 of Stable Agreements Version 3 as of September 14, 1990.

6.6.2.2 Standard Application Specific Object Classes

Refer to subclause 11.6.6.2.2 of Stable Agreements Version 3 as of September 14, 1990.

6.7 Restrictions on Object Class Definitions

Refer to subclause 11.6.7 of Stable Agreements Version 3 as of September 14, 1990.

7 Pragmatic Constraints

Refer to subclause 11.7 of Stable Agreements Version 3 as of September 14, 1990.

7.1 General Constraints

Refer to subclause 11.7.1 of Stable Agreements Version 3 as of September 14, 1990.

7.1.1 Character Sets

Refer to subclause 11.7.1.1 of Stable Agreements Version 3 as of September 14, 1990.

7.1.2 APDU Size Considerations

Refer to subclause 11.7.1.2 of Stable Agreements Version 3 as of September 14, 1990.

7.1.3 Service Control (SC) Considerations

Refer to subclause 11.7.1.3 of Stable Agreements Version 3 as of September 14, 1990.

7.1.4 Priority Service Control

Refer to subclause 11.7.1.4 of Stable Agreements Version 3 as of September 14, 1990.

7.2 Constraints on Operations

Refer to subclause 11.7.2 of Stable Agreements Version 3 as of September 14, 1990.

7.2.1 Filters

Refer to subclause 11.7.2.1 of Stable Agreements Version 3 as of September 14, 1990.

7.2.2 Errors

Refer to subclause 11.7.2.2 of Stable Agreements Version 3 as of September 14, 1990.

7.2.3 Error Reporting – Detection of Search Loop

Refer to subclause 11.7.2.3 of Stable Agreements Version 3 as of September 14, 1990.

7.3 Constraints Relevant to Specific Attribute Types

Refer to subclause 11.7.3 of Stable Agreements Version 3 as of September 14, 1990.

8 Conformance

Refer to subclause 11.8 of Stable Agreements Version 3 as of September 14, 1990.

8.1 DUA Conformance

Refer to subclause 11.8.1 of Stable Agreements Version 3 as of September 14, 1990.

8.2 DSA Conformance

Refer to subclause 11.8.2 of Stable Agreements Version 3 as of September 14, 1990.

8.3 DSA Conformance Classes

Refer to subclause 11.8.3 of Stable Agreements Version 3 as of September 14, 1990.

8.4 Authentication Conformance

Refer to subclause 11.8.4 of Stable Agreements Version 3 as of September 14, 1990.

8.5 Directory Service Conformance

Refer to subclause 11.8.5 of Stable Agreements Version 3 as of September 14, 1990.

8.6 The Directory Access Profile

Refer to subclause 11.8.6 of Stable Agreements Version 3 as of September 14, 1990.

8.7 The Directory System Profile

Refer to subclause 11.8.7 of Stable Agreements Version 3 as of September 14, 1990.

8.8 Digital Signature Protocol Conformance Profile

Refer to subclause 11.8.8 of Stable Agreements Version 3 as of September 14, 1990.

8.9 Strong Authentication Protocol Conformance Profile

Refer to subclause 11.8.9 of Stable Agreements Version 3 as of September 14, 1990.

9 Distributed Operations

Refer to subclause 11.9 of Stable Agreements Version 3 as of September 14, 1990.

9.1 Referrals and Chaining

Refer to subclause 11.9.1 of Stable Agreements Version 3 as of September 14, 1990.

9.2 Trace Information

Refer to subclause 11.9.2 of Stable Agreements Version 3 as of September 14, 1990.

10 Underlying Services

Refer to subclause 11.10 of Stable Agreements Version 3 as of September 14, 1990.

10.1 ROSE

Refer to subclause 11.10.1 of Stable Agreements Version 3 as of September 14, 1990.

10.2 Session

Refer to subclause 11.10.2 of Stable Agreements Version 3 as of September 14, 1990.

10.3 ACSE

Refer to subclause 11.10.3 of Stable Agreements Version 3 as of September 14, 1990.

11 Access Control

Refer to subclause 11.11 of Stable Agreements Version 3 as of September 14, 1990.

12 Test Considerations

Refer to subclause 11.12 of Stable Agreements Version 3 as of September 14, 1990.

12.1 Major Elements of Architecture

Refer to subclause 11.12.1 of Stable Agreements Version 3 as of September 14, 1990.

12.2 Search Operations

Refer to subclause 11.12.2 of Stable Agreements Version 3 as of September 14, 1990.

13 Errors

Refer to subclause 11.13 of Stable Agreements Version 3 as of September 14, 1990.

13.1 Permanent vs. Temporary Service Errors

Refer to subclause 11.13.1 of Stable Agreements Version 3 as of September 14, 1990.

13.2 Guidelines for Error Handling

Refer to subclause 11.13.2 of Stable Agreements Version 3 as of September 14, 1990.

13.2.1 Introduction

Refer to subclause 11.13.2.1 of Stable Agreements Version 3 as of September 14, 1990.

13.2.2 Symptoms

Refer to subclause 11.13.2.2 of Stable Agreements Version 3 as of September 14, 1990.

13.2.3 Situations

Refer to subclause 11.13.2.3 of Stable Agreements Version 3 as of September 14, 1990.

13.2.4 Error Actions

Refer to subclause 11.13.2.4 of Stable Agreements Version 3 as of September 14, 1990.

13.2.5 Reporting

Refer to subclause 11.13.2.5 of Stable Agreements Version 3 as of September 14, 1990.

14 Specific Authentication Schemes

Refer to subclause 11.14 of Stable Agreements Version 3 as of September 14, 1990.

14.1 Specific Strong Authentication Schemes

Refer to subclause 11.14.1 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1 ElGamal

Refer to subclause 11.14.1.1 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.1 References

Refer to subclause 11.14.1.1.1 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.2 Background

Refer to subclause 11.14.1.1.2 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.3 Digital Signature

Refer to subclause 11.14.1.1.3 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.4 Verification

Refer to subclause 11.14.1.1.4 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.5 Known Constraints on Parameters

Refer to subclause 11.14.1.1.5 of Stable Agreements Version 3 as of September 14, 1990.

14.1.1.6 Note on subjectPublicKey

Refer to subclause 11.14.1.1.6 of Stable Agreements Version 3 as of September 14, 1990.

14.1.2 One–Way Hash Functions

Refer to subclause 11.14.1.2 of Stable Agreements Version 3 as of September 14, 1990.

14.1.2.1 SQUARE–MOD–N Algorithm

Refer to subclause 11.14.1.2.1 of Stable Agreements Version 3 as of September 14, 1990.

14.1.2.2 MD2 Algorithm

Refer to subclause 11.14.1.2.2 of Stable Agreements Version 3 as of September 14, 1990.

14.1.2.3 Study of Other One–Way Hash Functions

The OIW Directory SIG is studying the applicability of alternative one–way hash functions. The most recent development in this area was the announcement by Ralph Merkle that 2–pass SNEFRU has been broken. Its use is therefore discouraged.

14.1.2.4 Use of One–Way Hash Functions in Forming Signatures

Refer to subclause 11.14.1.2.4 of Stable Agreements Version 3 as of September 14, 1990.

14.1.3 ASN.1 for Strong Authentication Algorithms

Refer to subclause 11.14.1.3 of Stable Agreements Version 3 as of September 14, 1990.

14.1.4 Note on the ENCRYPTED MACRO

Refer to subclause 11.14.1.4 of Stable Agreements Version 3 as of September 14, 1990.

14.2 Protected Simple Authentication

Refer to subclause 11.14.2 of Stable Agreements Version 3 as of September 14, 1990.

14.3 Simple Authentication

Refer to subclause 11.14.3 of Stable Agreements Version 3 as of September 14, 1990.

15 ANNEX A: MAINTENANCE OF ATTRIBUTE SYNTAXES

Refer to subclause 11.15 of Stable Agreements Version 3 as of September 14, 1990.

15.1 Introduction

Refer to subclause 11.15.1 of Stable Agreements Version 3 as of September 14, 1990.

15.2 General Rules

Refer to subclause 11.15.2 of Stable Agreements Version 3 as of September 14, 1990.

15.3 Checking Algorithms

Refer to subclause 11.15.3 of Stable Agreements Version 3 as of September 14, 1990.

15.3.1 distinguishedNameSyntax

Refer to subclause 11.15.3.1 of Stable Agreements Version 3 as of September 14, 1990.

15.3.2 integerSyntax

Refer to subclause 11.15.3.2 of Stable Agreements Version 3 as of September 14, 1990.

15.3.3 telephoneNumberSyntax

Refer to subclause 11.15.3.3 of Stable Agreements Version 3 as of September 14, 1990.

15.3.4 countryName

Refer to subclause 11.15.3.4 of Stable Agreements Version 3 as of September 14, 1990.

15.3.5 preferredDeliveryMethod

Refer to subclause 11.15.3.5 of Stable Agreements Version 3 as of September 14, 1990.

15.3.6 presentationAddress

Refer to subclause 11.15.3.6 of Stable Agreements Version 3 as of September 14, 1990.

15.4 Matching Algorithms

Refer to subclause 11.15.4 of Stable Agreements Version 3 as of September 14, 1990.

15.4.1 UTCTimeSyntax

Refer to subclause 11.15.4.1 of Stable Agreements Version 3 as of September 14, 1990.

15.4.2 distinguishedNameSyntax

Refer to subclause 11.15.4.2 of Stable Agreements Version 3 as of September 14, 1990.

15.4.3 caseIgnoreListSyntax

Refer to subclause 11.15.4.3 of Stable Agreements Version 3 as of September 14, 1990.

16 ANNEX B: GLOSSARY

Refer to subclause 11.16 of Stable Agreements Version 3 as of September 14, 1990.

17 ANNEX C: REQUIREMENTS FOR DISTRIBUTED OPERATIONS

Refer to subclause 11.17 of Stable Agreements Version 3 as of September 14, 1990.

17.1 General Requirements

Refer to subclause 11.17.1 of Stable Agreements Version 3 as of September 14, 1990.

17.2 Protocol Support

Refer to subclause 11.17.2 of Stable Agreements Version 3 as of September 14, 1990.

17.2.1 Usage of ChainingArguments

Refer to subclause 11.17.2.1 of Stable Agreements Version 3 as of September 14, 1990.

17.2.2 Usage of Chainging Results

Refer to subclause 11.17.2.2 of Stable Agreements Version 3 as of September 14, 1990.

18 ANNEX D: GUIDELINE FOR APPLICATIONS USING THE DIRECTORY

Refer to subclause 11.18 of Stable Agreements Version 3 as of September 14, 1990.

18.1 Tutorial

Refer to subclause 11.18.1 of Stable Agreements Version 3 as of September 14, 1990.

18.1.1 Overview

Refer to subclause 11.18.1.1 of Stable Agreements Version 3 as of September 14, 1990.

18.1.2 Use of the Directory Schema

Refer to subclause 11.18.1.2 of Stable Agreements Version 3 as of September 14, 1990.

18.1.2.1 Use of Existing Object Classes

Refer to subclause 11.18.1.2.1 of Stable Agreements Version 3 as of September 14, 1990.

18.1.2.2 Kinds of Object Classes

Refer to subclause 11.18.1.2.2 of Stable Agreements Version 3 as of September 14, 1990.

18.1.2.3 Use of Unregistered Object Classes

Refer to subclause 11.18.1.2.3 of Stable Agreements Version 3 as of September 14, 1990.

18.1.2.4 Side Effects of Creating Unregistered Object Classes

Refer to subclause 11.18.1.2.4 of Stable Agreements Version 3 as of September 14, 1990.

18.2 Creation of New Object Classes

Refer to subclause 11.18.2 of Stable Agreements Version 3 as of September 14, 1990.

18.2.1 Creation of New Subclasses

Refer to subclause 11.18.2.1 of Stable Agreements Version 3 as of September 14, 1990.

18.2.2 Creation of New Attributes

Refer to subclause 11.18.2.2 of Stable Agreements Version 3 as of September 14, 1990.

18.3 DIT Structure Rules

Refer to subclause 11.18.3 of Stable Agreements Version 3 as of September 14, 1990.

19 ANNEX E: TEMPLATE FOR AN APPLICATION SPECIFIC PROFILE FOR USE OF THE DIRECTORY

Refer to subclause 11.19 of Stable Agreements Version 3 as of September 14, 1990.

Table of Contents

12 STABLE SECURITY AGREEMENTS

Editor's Note: This section points to Stable Security Agreements which are contained in the aligned section of the Stable Implementation Agreements, Version 3 dated September 1990.

Table of Contents

1 Introduction

2 Scope

3 Normative References

3.1 ISO/IEC 9594-8 (CCITT X.509 Recommendation)

Information Technology - Open Systems Interconnection - The Directory -
Part 8: Authentication Framework.

3.2 ISO 8649: 1988/DAD 1

Service Definition for the Association Control Service Element, Addendum 1: Peer-Entity Authentication During Association Establishment, [SECSIG 90-02].

3.3 ISO 8650: 1988/DAD 1

Protocol Specification for the Association Control Service Element, Addendum 1: Peer-Entity Authentication During Association Establishment, [SECSIG 90-03].

3.4 ISO/IEC 9594-3 (CCITT X.511 Recommendation)

Information Technology - Open Systems Interconnection - The Directory -
Part 3: Abstract Service Definition.

3.5 ISO 10021-4 (CCITT X.411 Recommendation)

Information Processing Systems - Text Communication - MOTIS - Message Transfer System : Abstract Service Definition and Procedures.

4 Definitions

Editor's Note: This clause will contain all unique terms used in this part, to be determined.

Refer to ISO 7498/2 for definitions of security relevant terms. This base standard contains detailed descriptions of accepted security terms.

Refer to ISO TR-10000 for general ISO definitions used in this part.

The following security terms are not defined in ISO 7498/2:

- o Authentication

o Mechanism

Editor's Note: The above two terms will be defined as a work item.

5 Symbols and Abbreviations

6 Architectures

Editor's Note: Clause 6 below is evloving and is not considered stable enough to consider for insertion into the OIW Stable Implementation Agreements in the near future.

6.1 Introduction

Open Systems Security provides for secure distributed information processing in an environment which is heterogeneous in terms of technology and administration. For example, some environments may require protection from a minimal set of security threats while others require more complete protection.

An objective of the OIW Security SIG is to collaborate with other OIW SIGs in the development of security profiles based upon International Standards and Draft International Standards.*

The architectural objectives include:

a. Development of security profiles in collaboration with other OIW SIGs which support their communication architectures.

b. Agreement on and documentation of the security aspects of current OIW protocols.

c. Ensuring consistency in the use of security services and mechanisms in OIW Implementation Agreements.

Editor's Note: * This refers to the deliverable, stable text and is not to be taken as a constraint on documents to be considered by the group.

6.2 General OIW Application Environments

It is useful for the sake of simplification to look at the various OIW groups and to divide them into general categories so that a small set of general security profiles can be applied to similar application environments.

Generalized OIW application environments are given below:

a. Single Application Association (FTAM, VT, MMS, DS)

- Not an application relay
- Association is used to identify the parties in the communication (i.e. no intermediaries)
- Single application over the lifetime of the association

- Data exchanged can use security information which is dependent upon the application association.

b. Store and Forward (MHS)

- All store and forward is done in non-real time (application relay)

- Data exchanged includes complete security information which is not dependent on the application association.

c. Distributed Transactions (TP, RDA)

- Multiple applications over the lifetime of the association are possible
- Features delegation
- ACID properties are mandated (Atomicity, Consistency, Isolation and Durability)
- Need to authenticate at a finer granularity than the association

6.3 Security Profiles

This section is organized as follows:

1. Purpose of security profiles
2. Generic threat/security service table
3. Description of generic OIW security profiles

6.3.1 Purpose of Security Profiles

Editor's Note: Text TBD. We will further refine the profiles and how they define services and mechanisms in relation to threats.

6.3.2 Generic Threat/Security Service Table

Editor's Note: Threat/Security Service table to be developed for:

a. Mapping threats to services will be refined to be a one to many relationship.

b. Detailed threat description.

Editor's Note: Text references to 7498/2 and finer granularity on the threats to be added later.

6.3.3 Description of Generic OIW Security Profiles

o Profile 0 Null

o Profile 1 Basic

Authentication

The rationale for this set of profiles are that it is always an option to not support security at all. However, if security is to be supported, the minimum set of security services provided is authentication. The type of authentication will be a refinement by the specific application environment.
The remaining security services from 7498/2 shown in 13.6.3.2 may or may not be added into application specific security profiles. This is something that needs to be jointly determined between the Security SIG and the other OIW SIG involved.

6.4 Guidelines for OIW Application Profile Development

The following guidelines are provided for other OIW SIGs to use in the preliminary development of their own application specific security profile. It is intended that final completion of the security profiles should be done in a joint manner between the Security SIG and the other OIW SIGs.

The basic steps in the guidelines are as follows:

a. Start with the Security SIG Basic Security profile (13.6.3.3).

b. Perform application specific threat analysis. Map the result of this analysis to general security services.

c. Map general security services onto application specific security services (E.G. the threats identified for MHS in X.402 are mapped against MHS specific security services).

d. Editor's Note: Steps d and beyond are TBD. It will require further discussion to decide exactly how the application specific security profile is finally determined, how those profiles can be specified (security context, object identifier?) and how we will specify the mechanisms of choice for the implementation of the profile. Further discussion is needed on Security Policy. This is a priority work item.

Editor's Note: Proposed sections follow below: - text TBD

13.6.iSpecification of Security Profile
13.6.i.jSuggested Placement of Security Services
13.6.i.j.kSuggested Mechanisms for Security Services
13.6.i.j.k.lSuggested Placement of Mechanisms

7 Key Management

8 Lower Layers Security

9 Upper Layers Security

10 Message Handling System (MHS) Security

All current MHS security relevant text appears in part 8.

11 Directory Services Security

12 Network Management Security

This section outlines an approach to providing security services for OSI Network Management. The goals of this approach are to provide security in a manner that is simple and straight forward to implement, and to avoid any unnecessary computational and managerial overhead. The approach also takes into consideration the need for different levels of security services within different network management domains, and the near term requirement for interoperability of network management entities over disparate network types.

12.1 Threats

For the purpose of discussion, threats are divided into two categories: primary and secondary threats. Primary threats are those considered to be applicable to the full range of network management implementations, while secondary threats are considered to be applicable to the more limited range of highly secure implementations.

The primary threats to be protected against are the following:

a. The masquerading of a manager or agent entity.

b. The fabrication or modification of Common Management Information Protocol (CMIP) data units.

By countering primary threats, disruption of network management services by the casual user can be avoided.

The secondary threats to be protected against are the following:

a. All primary threats.

b. The disclosure of CMIP data units.

c. The replay, reflection, reordering, insertion, or deletion of CMIP data units.

12.2 Security Services

12.2.1 Basic Security Services

The security services required to counter primary threats are:

a. Peer Entity Authentication

b. Data Origin Authentication

c. Connectionless Integrity

Peer entity authentication is to occur during the establishment of an application association. If the association is successfully established, the underlying security mechanism provides information that is subsequently used in data origin authentication. There the information may be included in or, in some other way, transform the data units of subsequent exchanges so that they can be identified as originating from an authenticated entity. Both authentication security services are to be provided at the application level of protocol.

Connectionless integrity insures that data units originating from an authenticated source are not modifiable without detection. When combined with a strong data origin authentication mechanism, the ability to fabricate new data units is also countered. Connectionless integrity may be provided at either the application level of protocol or within one of the lower levels of protocol (i.e., transport or network). The former approach is described in this note and the decision of which to employ is left for further study.

12.2.2 Enhanced Security Services

The security services required to counter secondary threats are:

a. All basic security services with the possible exception of connectionless integrity.

b. Connectionless confidentiality.

c. Connection integrity with recovery.

Both connectionless confidentiality and connection integrity may be provided at either the application level of protocol or within one of the lower levels of protocol. The latter provision is assumed here. Enhanced security services are not discussed further in this note, but to be issued as a requirement for lower layer protocol and service standards, and according functional standards to be developed.

12.3 Security Mechanisms

12.3.1 Peer Entity Authentication

In order to simplify the management aspects associated with various phases of authentication procedures, the authentication scheme proposed is the same as that used for secure messages on the Internet [Ref. B26], which is compatible with the Directory Authentication Framework standard [Ref. 3.1]. The assumption is made that the certification authorities established for messaging would be usable and suitable for network management as well. It is also assumed that certificates will identify the owner, the owner's public key, dates of validity, and be signed by the certification authority, and that successful authentication results in the establishment of a cryptographic association.

One suitable location to convey authentication information is the association control service element (ACSE) authentication field, described in the addenda to the ACSE service definition and protocol specification covering peer-entity authentication during association establishment [Ref. 3.2, 3.3].

12.3.1.1 ACSE Authentication Field Usage

ACSE authentication extensions [Ref. 3.2, 3.3] support two-way authentication through the definition of a new functional unit. When this functional unit is employed, additional parameters are provided by the A-ASSOCIATE service to indicate this requirement and convey authentication information between entities. The ASN.1 definition for this information is given below.

```
        from ISO 8650: 1988/DAD1 [Ref. 3.3]

        Authentication ::= SEQUENCE {
                mechanism-name [0] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                authentication-value [1] CHOICE {
                        charstring [0] IMPLICIT GraphicString,
                        bitstring [1] IMPLICIT BITSTRING,
                        external [2] IMPLICIT EXTERNAL,
                        other [3] ANY DEFINED BY mechanism-name }  }
```

It is proposed that support of ACSE authentication functional unit be mandatory for network management. This may require that a mechanism-name be defined and registered for network management, and that the corresponding authentication-value (i.e., CHOICE other [3]) defined by that mechanism be conveyed. Since it is intended that the ASN.1 definitions of the authentication field arguments, the procedures for handling those arguments, and their mapping onto ACSE be consistent with other application layer protocols, this defined mechanism conceivably may be utilized by other application protocols.

12.3.1.2 Authentication Value Definition

The authentication scheme used for privacy enhancement of Internet electronic mail [Ref. B26] relies on a key management architecture based on the use of public key certificates. Certificates are issued by a certification authority (CA) and contain the public key of a principa, its identity, and other related information such as the serial number and validity period of the certificate, and the identity of the issuer. The CA acting on behalf of the IA applies a digital signature to a certificate that provides non-forgable assurance of the identity of a principle and binding it to the given public key. The according ASN.1 definitions are given below.

```
        from ISO 9495-8 [Ref. 3.1], Annex G:

        Certificate ::= SIGNED SEQUENCE {
                version  [0] Version DEFAULT v1988,
                serialNumber  CertificateSerialNumber,
                signature  AlgorithmIdentifier,
                issuer  Name,
                validity ,
                subject  Name,
                subjectPublicKeyInfo  }

        SIGNED MACRO ::=
                BEGIN
                TYPE NOTATION ::= type(ToBeSigned)
                VALUE NOTATION ::= value (VALUE
                        SEQUENCE{
                                ToBeSigned,
                                AlgorithmIdentifier,
                                ENCRYPTED OCTETSTRING } )
```

Credentials are used to establish the identity of a user and provide the means for meaningful utilization of public key certificates. In addition to conveying the certificate of a principa, they can protect against replay attacks and, in situations where a hierarchy of certification authorities exists, they can convey the chain of CA certificates that are needed by the recipient to verify the senders certificate. One definition of credentials taken from the abstract service definition of the directory bind operation [Ref. 3.4] is given below.

from ISO 9495-3 [Ref. 3.4], Annex A, and ISO 9495-8 [Ref. 3.1]:

```
        Credentials ::= CHOICE {
                        simple  [0] SimpleCredentials,
                        strong  [1] StrongCredentials,
                        externalProcedure [2] EXTERNAL }

        SimpleCredentials ::= SEQUENCE {
                        name [0] DistinguishedName,
                        validity [1] SET { time1 [0] UTCTime OPTIONAL,
                                           time2 [1] UTCTime OPTIONAL,
                                           random1 [2] BITSTRING
                  random2 [3] BITSTRING} OPTIONAL,
                        password [2] OCTETSTRING OPTIONAL }

        StrongCredentials ::= SET {
                        certification-path [0] CertificationPath OPTIONAL,
                        bind-token  [1] Token }

        Token ::= SIGNED SEQUENCE {
                        algorithm [0] AlgorithmIdentifier,
                        name [1] DistinguishedName,
                        time [2] UTCTime,
                        random [3] BITSTRING }

        CertificationPath ::= SEQUENCE {
                userCertificate Certificate,
                theCACertificates SEQUENCE OF CertificatePair OPTIONAL}

        CertificatePair ::= SEQUENCE {
```

```
forward [0] Certificate OPTIONAL,
reverse [1] Certificate OPTIONAL }
```

It is proposed that this definition be utilized by network management as the ACSE authentication value definition.

12.3.2 Connectionless Integrity

In order to identify whether changes to a data unit have occurred it is proposed that an integrity check value (ICV) be computed over the entire data unit and included in the protocol control information for that data unit. The specification and location for conveying this information is left for further study. Because of the envisaged relationship between the underlying mechanisms employed for data origination authentication and connectionless integrity, they are to be considered jointly.

12.3.3 Data Origination Authentication

The proposed security mechanism for data origination authentication is encipherment and intended to protect the ICV computed for connectionless integrity. Successful peer authentication results in the establishment of a cryptographic association between network management entities. The association allows the originator of a data unit to encrypt it or portions of it, and have the peer recipient verify origination through decryption. In order to minimize computational effort, it is proposed that only the integrity check value be enciphered (i.e., a signature) rather than the entire data unit.

This approach implies that data origination authentication information resides with the integrity check value, and that an according ASN.1 definition reflect any requirements of the signing algorithm or choice of algorithm. However, there appears to be no appropriate location in the application layer protocols employed by network management to convey such data origination authentication information. This issue is left for further study.

13 Annex A: ISPICS Requirements List

14 Annex B: Bibliography

B.1 ISO/IEC JTC1 SC21 N3614 Information Retrieval, Transfer, and Management for OSI

B.2 ISO/IEC DP 9796 Data Cryptographic Techniques

B.3 Secure Data Network System (SDNS): Key Management Profile - Communications Protocol Requirements (SDN-601/NIST IR 90-4262)

B.4 SDNS: Message Security Protocol (SDN-701/NIST IR 90-4250)

B.5 SDNS: Directory (SDN-702/NIST IR 90-4250)

B.6 ISO/IEC JTC1 SC21/WG1 N5002 Security ASE

B.7 Access Control Information Specification (ACIS)

B.8 SDNS: Key Management Protocol - Definition of Services Provided (SDN-902/NIST IR 90-4262)

B.9 SDNS: Key Management Protocol - Specification of the Protocol (SDN-903/NIST IR 90-4262)

B.10 ISO/IEC JTC1 SC21/WG1 N4110 Authentication ASE Exchange

B.11 SDNS: Security Protocol 3 (SDN-301/NIST IR 90-4250)

B.12 SDNS: Security Protocol 4 (SDN-401/NIST IR 90-4250)

B.13 SDNS: Key Management Protocol - SDNS Traffic Key (SDN-906/NIST IR 90-4262)

B.14 ISO/IEC JTC1 SC21/WG1 N5001 Upper Layers Security Model

B.15 ISO/IEC JTC1 SC21/WG1 F29 N5045 Access Control Framework

B.16 ISO/IEC JTC1 SC21/WG1 F30 Authentication Framework

B.17 ISO/IEC JTC1 SC21/WG1 F31 N5047 Integrity Framework

B.18 ISO/IEC JTC1 SC21/WG1 F32 N5046 Non-Repudiation

B.19 ISO/IEC JTC1 SC21/WG4 N3775 Security Audit Trail

B.20 ISO/IEC JTC1 SC21/WG1 N4110 Authentication ASE Exchange

B.21 ISO/IEC JTC1 SC21/WG7 N4022 Key Management Framework

B.22 ISO/IEC JTC1 SC21/WG1 N5048 Confidentiality Framework

B.23 ISO/IEC JTC1 SC21/WG1 N5049 Guide to OSI Security Standards

B.24 ISO/IEC JTC1 SC21/WG1 N5044 Security Framework Overview

B.25 RFC-1113, Privacy Enhancement for Internet Electronic Mail: Part I - Message Encipherment and Authentication Procedures.

B.26 RFC-1114, Privacy Enhancement for Internet Electronic Mail: Part II - Certificate-Based Key Management.

B.27 RFC-1115, Privacy Enhancement for Internet Electronic Mail: Part III - Algorithms, Modes, and Identifiers (August 1989).

15 Annex C: Status

16 Annex D: Errata

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	SECTION	NOTES
	TECHNICAL	WIA PART - 13	10	MOVED TO MHS
	TECHNICAL	WIA PART - 13	12.3.1	ADDED
	TECHNICAL	WIA PART - 13	3	ADDED
	TECHNICAL	WIA PART - 13	4	CHANGED, DELETED
	EDITORIAL	WIA PART - 13	ALL	NOTES ADDED, EDITING

17 Annex E: Security Labels

18 Annex F: Security-SIG Management Plan

NUMBER	NEXT MILESTONE	DATE
ISO/IEC JTC1 SC21 N3614		
ISO/IEC DP 9796		
SDN-601/NIST IR 90-4262		
SDN-701/NIST IR 90-4250		
SDN-702/NIST IR 90-4250		
ISO/IEC JTC1 SC21/WG1 N5002		
SDN-902/NIST IR 90-4262		
SDN-903/NIST IR 90-4262		
ISO/IEC JTC1 SC21/WG1 N4110		
SDN-301/NIST IR 90-4250		
SDN-401/NIST IR 90-4250		
SDN-906/NIST IR 90-4262		
ISO/IEC JTC1 SC21/WG1 N5001		
ISO/IEC JTC1 SC21/WG1 F29 N5045		
ISO/IEC JTC1 SC21/WG1 F30		
ISO/IEC JTC1 SC21/WG1 F31 N5047		
ISO/IEC JTC1 SC21/WG1 F32 N5046		
ISO/IEC JTC1 SC21/WG4 N3775		
ISO/IEC JTC1 SC21/WG1 N4110		
ISO/IEC JTC1 SC21/WG7 N4022		
ISO/IEC JTC1 SC21/WG1 N5048		
ISO/IEC JTC1 SC21/WG1 N5049		

Table of Contents

14 Part 14 - ISO Virtual Terminal Protocol

Editor's Note: References to Stable Agreements in this part refer to Version 3 dated September 1990.

1 Introduction

See Stable Agreements.

2 Scope and Field of Application

2.1 Phase Ia Agreements

See Stable Agreements.

2.2 Phase Ib Agreements

See Stable Agreements regarding Forms profile.

The Scroll profile is intended to support line-at-a-time applications and has colour and text attribute capabilities.

2.3 Phase II Agreements

See Stable Agreements regarding X.3 profile.

The Page profiles are intended for applications which require page-oriented operation.

3 Status

These agreements are being done in phases. Below is the current status of each phase.

3.1 Status of Phase Ia

The Phase Ia Agreements, which include the profiles for Telnet and Transparent operation, are complete and were stabilized in May, 1988. See Stable Agreements.

3.2 Status of Phase Ib

The Forms profile of Phase 1b was stabilized in December, 1988. Alignment with EWOS Forms profile was achieved in September, 1989. See Stable Agreements.

3.3 Status of Phase II

The Phase II agreements include profiles for Scroll, X.3 and Page operations and will be completed at an unspecified future date, except for X.3, as mentioned below.

The X.3 profile was stabilized in December, 1989. See Stable Agreements.

It is intended that Phase II agreements be compatible with Phase I agreements.

4 Errata

5 Conformance

See Stable Agreements.

6 Protocol

See Stable Agreements.

7 OIW Registered Control Objects

7.1 Sequenced Application (SA)

See Stable Agreements.

7.2 Unsequenced Application (UA)

See Stable Agreements.

7.3 Sequenced Terminal (ST)

See Stable Agreements.

7.4 Unsequenced Terminal (UT)

See Stable Agreements.

7.5 Termination Conditions CO (TC)

This CO is an instance of the standard type TCCO, as defined in ISO 9040. It is initially designed for use with the OIW Scroll VT profile, though as a registered CO it is available for use by other VT profiles.

In addition to the three standardized data elements, it provides a definition and update syntax for further types of Termination Condition. Each additional type is available for use in additional data elements of the CO. The number and type of such additional data elements is defined in the profile using this CO.

7.5.1 Entry Number

To be supplied by the Registration Authority.

7.5.2 Name of Sponsoring Body

NIST/OSI Workshop for Implementors of OSI, VTSIG.

7.5.3 Date

The date of submission of this proposal is September 15, 1989.

7.5.4 Identifier

oiw-vt-co-tcco-tc OBJECT IDENTIFIER ::= { oiw-vt-co-tcco tc(0) }

7.5.5 Descriptor Value

"OIW VT CO for Termination Conditions"

7.5.6 CO VTE-parameters

```
CO-structure    = ,      *(not defined in this registration, see note 1 in 14.7.5.8)*
CO-priority     = "normal"
        {
        CO-element-id  = 1,     *(termination length)*
        CO-category    = "integer",
        CO-size        = 65535 },
        {
```

```
        CO-element-id  = 2, *(time-out mantissa)*
        CO-category    = "integer",
        CO-size        = 65535 },
        {
        CO-element-id  = 3, *(time-out exponent)*
        CO-category    = "integer",
        CO-size        = 65535 },
```

(the following represents possibly multiple invocations of a generic data element type, according to the value of CO-structure for the instance of this CO.)

```
        FOR N=4 to CO-structure
        {
        CO-element-id  = N,     *(acts as integer identifier for the events in this element)*
        CO-category    = "transparent",
        CO-size        =        *(not defined in this registration, see note 2 in 14.7.5.8)* }
```

7.5.7 CO Values, Semantic and Update Syntax

The value fields for data elements 1,2 and 3 are defined in ISO 9040.

The value field for each additional data element is defined by the following ASN.1 construct which also defines the update syntax.

```
TermCondList  ::= SEQUENCE OF CHOICE {
              void                  [0] IMPLICIT NULL,
              x3ForwardingCond      [1] IMPLICIT INTEGER,
              stEventList           [2] IMPLICIT Range,
              anySTUpdate           [3] IMPLICIT NULL,
              stEventMasks          [4] IMPLICIT MaskValues,
              dOChars               [5] IMPLICIT DOCharacters }

Range         ::= SEQUENCE OF SEQUENCE {
                                    [1] IMPLICIT LogEvent,
                                    [2] IMPLICIT LogEvent OPTIONAL }
-- each pair represents an interval of values as defined for the value field of
--CO ST, see 14.7.3.7.  The second value in each pair shall not be smaller than
--the first value.  If the second value is omitted, the interval contains only
--the specified first value.

LogEvent      ::= INTEGER
-- values as defined for value field of CO ST, see 14.7.3.7.

MaskValues    ::= SEQUENCE OF SEQUENCE {
              mask                  [1] IMPLICIT LogEvent,
              value                 [2] IMPLICIT LogEvent }

DOCharacters  ::= SEQUENCE OF SEQUENCE {
                                    [1] IMPLICIT Repref,
                                    [2] IMPLICIT INTEGER,
                                    [3] IMPLICIT INTEGER OPTIONAL }

Repref          ::= INTEGER
-- index to the list of repertoires for the Display Object
```

7.5.8 Additional Information

Note: The value of CO-structure is defined in the profile to be the number of types of termination conditions available for use within the profile.

Note: The value of CO-size for each additional data element of this CO must be defined within the profile definition which uses those additional termination conditions.

7.5.9 Usage

Defined in profile.

8 OIW Defined VTE-Profiles

8.1 Telnet Profile

See Stable Agreements.

8.2 Transparent Profile

See Stable Agreements.

8.3 Forms Profile

See Stable Agreements.

8.4 X3 Profile

See Stable Agreements.

8.5 Scroll Profile

OIW VTE-Profile Scroll-1989 (r1,r2,...r9)

8.5.1 Introduction

This Scrolling A-mode VTE-profile is designed to support line-at-a-time interactions between a terminal and a host system, the type of operation typified by operating system command entry.

Scrolling is bi-directional, forward and backward.

The profile also provides a facility for switching local echo "on" or "off".

This VTE-Profile supports what is often referred to as "type-ahead", so input from the terminal user is available to the host application as soon as the application is ready for input, thus providing efficiency by minimizing communication delays.

This VTE-profile supports the definition of "input" termination events by the "Application VT-user" so the application can specify what events will cause "input" data to be forwarded to the "Application VT-user".

8.5.2 Association Requirements

8.5.2.1 Functional Units

The Urgent Data Functional Unit is optional, and will be used if available.

8.5.2.2 Mode

This profile operates in A-mode.

8.5.3 Profile Body

```
Display-objects =
{
        {
        display-object-name = DOA,
        DO-access = profile-argument-rl,
        dimension = "two",
                x-dimension =
                {
                        x-bound = profile-argument-r2,
                        x-addressing = "no-constraint",
                        x-absolute = "no",
                        x-window = x-bound
                },
                y-dimension =
                {
```

```
                y-bound = "unbounded",
                y-addressing = "no-constraint",
                y-absolute = "no",
                y-window = profile-argument-r10
        },

erasure-capability = "yes",

*( repertoire-capability is implied by the number of occurrences of profile-argument-r4 )*

repertoire-assignment = profile-argument-r4,

DO-emphasis = profile-argument-r5,

foreground-colour-capability = profile-argument-r6,
foreground-colour-assignment = profile-argument-r7,
background-colour-capability = profile-argument-r6,
background-colour-assignment = profile-argument-r8
},
{
display-object-name = DOB,
DO-access = opposite of profile-argument-rl,
dimension = "two",
        x-dimension =
        {
                x-bound = profile-argument-r2,
                x-addressing = "no-constraint",
                x-absolute = "no",
                x-window = x-bound
        },
        y-dimension =
        {
                y-bound = "unbounded",
                y-addressing = "higher only",
                y-absolute = "no",
                y-window = 1
        },
erasure capability = "yes",
*( repertoire-capability is implied by the number of occurrences of profile-argument-r4 )*

repertoire-assignment = profile-argument-r4,

DO-emphasis = profile-argument-r5,

foreground-colour-capability = profile-argument-r6,
foreground-colour-assignment = profile-argument-r7,
background-colour-capability = profile-argument-r6,
background-colour-assignment = profile-argument-r8
```

```
        }
},

Control-objects =
{
        {
        CO-name                 = E,     *(standard Echo CO)*
        CO-type-identifier      = vt-b-sco-echo,
        CO-access               = profile-argument-r1,
        CO-priority             = "normal",
        CO-trigger              = "selected",
        CO-category             = "boolean",
        CO-size                         = 1
        },
        IF r9 = "TE" THEN
        {
        CO-name                 = TE, *(Termination Event CO)*
        CO-type-identifier      = vt-b-sco-tco,
        CO-access               = opposite of profile-argument-r1,
        CO-priority             = "normal",
        CO-trigger              = "selected",
        CO-category             = "integer"
        },
                {
        CO-name                 = SA, *(NIST Registered CO)*
        CO-type-identifier      = nist-vt-co-misc-sa,
        CO-access               = profile-argument-r1,
        CO-priority             = "normal",
        CO-trigger              = "not selected",
        CO-category             = "integer",
        CO-size                 = 65535
        },
                        {
        CO-name                 = UA, *(NIST Registered CO)*
        CO-type-identifier      = nist-vt-co-misc-ua,
        CO-access               = profile-argument-r1,
        CO-priority             = "urgent",
        CO-category             = "integer",
        CO-size                 = 65535
        },
                        {
        CO-name                 = ST, *(NIST Registered CO)*
        CO-type-identifier      = nist-vt-co-misc-st,
        CO-access               = opposite of profile-argument-r1,
        CO-priority             = "normal",
        CO-category             = "integer",
        CO-size                 = 65535
        },
        {
        CO-name                 = UT, *(NIST Registered CO)*
        CO-type-identifier      = nist-vt-co-misc-ut,
        CO-access               = opposite of profile-argument-r1,
        CO-priority             = "urgent",
        CO-category             = "integer",
        CO-size                 = 65535
        },
        {
        CO-name                 = TC, *(Termination conditions CO)*
        CO-type-identifier      = nist-vt-co-tcco-tc,
        CO-structure            = N, *( defined with TCCO)*
        CO-access               = profile-argument-r1,
        CO-priority             = "normal",
                {
                CO-element-id   = 1, *(termination length)*
                CO-category     = "integer",
                CO-size         = 65535 },
                {
                CO-element-id   = 2, *(time-out mantissa)*
                CO-category     = "integer",
                CO-size         = 65535 },
                {
                CO-element-id   = 3, *(time-out exponent)*
                CO-category     = "integer",
                CO-size         = 65535 },
                {
                CO-element-id   = 4-N, *(from registered TCCO)*
                CO-category     = ???,
                CO-size         = ??? }
```

The NIST Workshop VT SIG is defining this registered TCCO. This TCCO is a reference to that registered control object.

```
        }
}

        Device-objects =
        {
                {
                device-name = DVA,    *("output" device object)*
                device-default-CO-access = profile-argument-rl,
                device-default-CO-initial-value = 1."true",
                device-display-object = DOA,
                device-minimum-X-array-length = profile-argument-r2,
                device-minimum-Y-array-length = profile-argument-r3,
                device-control-object = {SA,UA}
                },
                {
```

```
                device-name = DVB,    *("input" device object)*
                device-default-CO-access = opposite of profile-argument-r1,
                device-default-CO-initial-value = 1."true",
                device-display-object = DOB,
                device-minimum-X-array-length = profile-argument-r2,
                device-control-object = profile-argument-r9,
                device-control-object = {ST,UT},
                device-control-object = TE
                }
    },

    type-of-delivery-control = "simple-delivery-control".
```

8.5.4 Profile Argument Definitions

r1 - is mandatory and enables negotiation of which VT-user has update access to display object DOA. It takes values "WACI", "WACA". It implies the asymmetric roles of the VT-users as "Application VT-user" and "Terminal VT-user". If the value for DOA is "WACI", then the association initiator is the "Application VT-user"; if the value of DOA is "WACA", then the association initiator is the "Terminal VT-user". This profile argument is also used to determine which VT-user has access to other VT objects as described above. Reference in the profile definition to "opposite of profile- argument-r1" means that the alternative of the two possible values for profile- argument-r1 is to be used. This argument is identified by the identifier for DO-access for display object DOA.

r2 - is optional and enables negotiation of a value for the VTE-parameter x-bound for the display objects DOA and DOB. It takes an integer value greater than zero. This argument is identified by the identifier for x-bound for display object DOA. Default is 80.

r3 - is optional and enables the negotiation of a value for the VTE-parameter device-minimum-Y-array-length for device object DVA. It takes an integer value greater than zero; if absent, a device of any length will be satisfactory.

Note: Indicates screen length.

r4 - is optional and provides for the negotiation of value(s) for the VTE-parameter repertoire-assignment. The value of repertoire-capability is implied by the number of occurrences of this argument. Default is specified by 9040.

r5 - is optional and provides for the negotiation of a value for the VTE-parameter DO-emphasis. The default value is that given in ISO 9040, B.17.3. Refer to ISO 9040 B.17.4 for rules governing the selection of non-default values.

r6 - is optional and provides for the negotiation of value(s) for VTE-parameters foreground-colour-capability and background-colour-capability. Default is 8.

r7 - is optional and provides for the negotiation of a value for VTE-parameter foreground-colour-assignment. Default is {"white", "black", "red", "cyan", "blue", "yellow", "green", "magenta"}.

r8 - is optional and provides for the negotiation of a value for VTE-parameter background-colour-assignment. Default is {"black", "white", "cyan", "red", "yellow", "blue", "magenta","green"}.

r9 - is optional and enables negotiation of a termination control object. The value for this argument is the value of CO-name for the termination control object, i.e. "TE"; if absent, no termination control is defined.

r10 - is optional and provides for the negotiation of a value for the VTE-parameter y-window of the DOA Display Object. Default is 24.

8.5.5 Profile Dependent CO Information

This profile makes use of five NIST registered Control Objects, SA, UA, ST, UT and TCCO. The CO-access in each CO is defined within this profile.

8.5.6 Profile Notes

8.5.6.1 Definitive Notes

1. Only the first boolean of the default control object contained in each device object is defined. This boolean is defined as the "on/off" switch for the device where the value "true" ="on" and "false" = "off". These values were chosen so the initial value of the boolean, "true", means the device is initially "on" and data to/from the display objects is being mapped to the device.

2. Only one boolean is defined in the standard echo control object, E. The semantics of this boolean is defined such that "false" means "local echo off" and "true" means "local echo on"; these values were chosen so echoing is initially "off" (which would provide security when a password is entered at the start of a terminal session).

8.5.6.2 Informative Notes

1. This profile models a scrolling device which is capable of scrolling both forwards and backwards. The display pointer may be moved backwards to modify earlier lines. A typical use for this profile is for applications where type-ahead may be advantageous and control over local echo "on"/"off" is required, e.g. the type of application where a conventional teletypewriter device or 'teletype-compatible' video device having 'full duplex' capability is often used. Display object DOA referred to above is typically mapped to the display or printing device and display object DOB is typically mapped to the keyboard.

2. Use of A-mode enables "typed-ahead"into display object DOB, and such updates can be delivered immediately to the peer VT-user, potentially reducing transmission delays. Such delivery will be forced, and marked, by a termination condition or a VT-DELIVER. Type-ahead is at the discretion of the terminal user.

3. Display object DOB has an unbounded y-dimension so as to provide a blank line for each new line entered.

4. Line-at-a-time forward scrolling is mapped onto an update-window (value zero) which allows NO backward updates to preceding lines (x-arrays). The device-minimum-Y-array-length negotiated by profile-argument-r3 can be used to indicate the number of lines (x-arrays) which should remain visible to the human terminal user although specifically NOT available for update.

5. The ability to switch local echo "on" or "off" is always present; the ECHO control object is used for this purpose.

8.5.7 Specific Conformance Requirements

None.

9 Annex A

See Stable Agreements.

10 Annex B - Clarifications

10.1 Defaults

See Stable Agreements.

11 Annex C - Object Identifiers

See Stable Agreements for Object Identifiers assigned to objects in the Stable Agreements. Object Identifiers below have been assigned to objects for which work is still in progress.

Profiles defined by OIW VT SIG:

```
oiw-vt-pr-scroll-1989  OBJECT IDENTIFIER ::= { oiw-vt-pr        scroll-1989(3) }
```

Control Objects defined by OIW VT SIG:

```
oiw-vt-co-tcco-tc  OBJECT IDENTIFIER ::=        { oiw-vt-co-tcco  tc(0) }
```

PART 15: TRANSACTION PROCESSING September 1990 (Working)

Table of Contents

15 Introduction

Editor's Note: This section has been editorially changed to allow numbers for subelements.

Editor's Note: Changes were approved at the September 1990 Workshop and will be included later.

The NIST/OIW Transaction Processing (TP) Sig is developing implementation agreements for the TP model, service and protocol, ISO 10026 (parts 1,2 and 3).

A transaction, as defined in ISO 10026, is a set of related operations characterized by the ACID properties. The ACID properties are:

> *Atomicity*: a property of a set of related operations such that the operations are either all performed, or none of them are performed.
>
> *Consistency*: a property of a set of related operations such that the effect of the operations are performed accurately, correctly, and with validity, with respect to application semantics. Bound data is moved from one consistent state to another consistent state.
>
> *Isolation*: a property of a set of related operations such that the partial results of the operations are not accessible, except by operations of the set.
>
> *Durability*: a property of a completed set of related operations such that all the effects of the operation are not altered by any sort of failure.

1 Scope

These agreements will address the following areas:

1. Specification of functional unit profiles:
   A. Kernel
   B. Polarized Control
   C. Shared Control
   D. Handshake
   E. Commit
   F. Unchained Transactions
2. Agreements covering TP services and generation of TP protocol.
3. Agreements covering the use of the following OSI services by TP:
   A. ACSE for association management
   B. CCR for support of provider supported ACID properties
   C. Presentation service
   D. Directory services
4. Agreements with regard to implementation issues not specified in ISO 10026.
5. Statement of requirements to meet conformance to the agreements.
6. Additionally, the following interoperability issues will be addressed:
   A. TP usage by other OSI standards

B. Application context
C. Security

2 SPECIFICATION OF FUNCTIONAL UNITS

2.1 FUNCTIONAL UNITS

Kernel

Polarized Control

Shared Control

Handshake

Commit

Unchained Transactions

2.2 COMBINATIONS OF FUNCTIONAL UNITS

Application Transactions

Unchained Provider-supported Transactions

Chained Provider-supported Transactions

2.3 TP USE OF OSI SERVICES

2.3.1 ACSE - ASSOCIATION MANAGEMENT

2.3.2 CCR - PROVIDER ACID PROPERTIES

2.3.3 PRESENTATION SERVICES

2.3.4 DIRECTORY SERVICES

2.3.5 IMPLEMENTATIONS ISSUES NOT SPECIFIED IN ISO 10026

2.3.6 APPLICATION CONTEXT

2.3.7 SECURITY

2.3.8 RECOMMENDED PRACTICES

2.4 CONFORMANCE STATEMENT

3 OSI TRANSACTION PROCESSING PROTOCOL AGREEMENTS

The tables below detail the requirements included in the NIST OSI TP Implementation Agreement. The tables present the following information:

o Optional and Mandatory PDU fields and their ranges

o Optional and Mandatory ASE service primitive parameters and their ranges

All the tables are written in a PICS-like format. Each row contains a field or parameter followed by the standard's requirements for that item and then NIST's (Implementation Agreement) requirements. For PDU fields and service parameters, additional columns containing a range and notes are included.
Unless otherwise noted, the following column descriptions and keys apply to all tables:

FIELD/PARAMETER: The particular standard-defined field or parameter being described.

STND: The Transaction Processing standard's (ISO 10026) requirements for the item. This field will have one of the following values; their meaning is defined by the international standard.

M: Mandatory
C: Conditional
O: Optional
NU: Not Used

NIST: This implementation agreement's requirements for the item. This field will have one of the following values; their meaning is defined by the implementation agreement.

Y: Supported, this is a mandatory or optional feature in the base standard. Its syntax and semantics shall be implemented as specified in the base standard or the TP agreements by all implementations claiming conformance to the profile. It is not a requirement that the feature shall be used in all instances of communications, unless mandated by the base standard or stated otherwise in the TP agreement. Fully supported attributes will conform to at least the minimum range of values as defined in ISO 10026-3, unless stated otherwise in the TP agreement. Conformant implementations supporting optional features will be able to interoperate with those implementations which do not support the feature. The support of a feature can depend on the support of a class of features to which it belongs, e.g. parameter in a PDU, a PDU in a functional unit.

O: Optionally supported, is left to the implementation as to whether this feature is supported. If a parameter is optionally supported, then the syntax shall be supported, but it is left to each implementation whether the semantics are supported. The receiver of an unsupported optional parameter which is not subject to negotiation shall, at least, inform the sender by informative diagnostic,and interoperability will not be affected.

NIST RANGE: The allowable range of values for this parameter.

SOURCE: Who supplies data for the parameter. This field will have one of the following values:

TPPM: Transaction Processing Protocol Machine
REQ: Requesting TPSUI

SINK: Who uses the parameter. This field will have one of the following values:

TPPM: Transaction Processing Protocol Machine
IND: Receiving TPSUI
REQ: Requesting TPSUI

NOTES: Any additional comments applying to the parameter.

3.1 TP-BEGIN-DIALOGUE-RI

Sending, to begin a dialogue

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Initiating-TPSU-Title	O	O	TPPM	0..2**31-1	
Recipient-TPSU-Title	C	Y	Req	0..2**31-1	
Selected-Functional-Units	C	Y	Req		2
Commit	O	O			
Polarized-Control	O	O			
Handshake	O	O			
Unchained-Transactions	O	O			
Initial-Coordination-Level	C	Y	Req		
Invocation-data	O	O	Req		1
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

Sending, to begin a TP channel

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	
Channel-utilization	C	Y	TPPM		

Receiving, to begin a dialogue

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Initiating-TPSU-Title	O	Y	Ind	0..2**31-1	
Recipient-TPSU-Title	C	Y	Ind	0..2**31-1	
Selected-Functional-Units	C	Y	Ind		2
Commit	O	O			
Polarized-Control	O	O			
Handshake	O	O			
Unchained-Transactions	O	O			
Initial-Coordination-Level	C	Y	Ind		
Invocation-data	O	O	Ind		1
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

Receiving, to begin a TP channel

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	
Channel-Utilization	C	Y	TPPM		

Note:

1. May need to determine limits on the amount and type of data passed in this manner.
2. See section "Support of Functional Units" for minimum valid combinations of functional units.

3.2 TP-BEGIN-DIALOGUE-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

3.3 TP-REJECT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostic	C	Y			1, 4
User-data	O	O	Req		2, 3
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

TP-REJECT-RI, Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostic	C	Y			1, 4
User-data	O	O	Req		2, 3
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

Note:
1. **User/Provider division of values is unclear in standard's ASN.1.**
2. **May need to determine limits on the amount and type of data passed in this manner.**
3. **Parameter is present on provider rejects.**
4. **Parameter is present on user rejects.**

3.4 TP-BID-RI

No parameters

3.5 TP-BID-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Result	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Result	M	Y	TPPM		

3.6 TP-END-DIALOGUE-RI

No parameters

3.7 TP-U-ERROR-RI

No parameters

3.8 TP-U-ERROR-RC

No parameters

3.9 TP-P-ERROR-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Diagnostic	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Diagnostic	M	Y	Ind		

3.10 TP-ABORT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostics	C	Y	TPPM		1, 4
User-data	C	O	Req		2, 3

TP-ABORT-RI, Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	Ind		
Diagnostics	C	Y	Ind		1, 4
User-data	C	O	Ind		2, 3

Note:

1. May want to specify meanings for the reason codes, Permanent and Transient failure.
2. May need to determine limits on the amount and type of data passed in this manner. Text says parm is optional, ASN.1 says mandatory.
3. Parameter is present on provider abort.
4. Parameter is present on user abort.

3.11 TP-REQUEST-CONTROL-RI

No parameters

3.12 TP-GRANT-CONTROL-RI

No parameters

3.13 TP-HANDSHAKE-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Confirmation	C	Y	Req		

TP-HANDSHAKE-RI, *Receiving*

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	TPPM		
Confirmation	C	Y	TPPM		1

Note: 1. Parameter is present only on handshake when Shared Control functional unit is active.

3.14 TP-HANDSHAKE-RC

No parameters

3.15 TP-HANDSHAKE-AND-GRANT-CONTROL-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Confirmation	M	Y	Req		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Confirmation	M	Y	TPPM		

3.16 TP-HANDSHAKE-AND-GRANT-CONTROL-RC

No parameters

3.17 TP-DEFER-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
End-dialogue	O	Y	TPPM		1
Grant-control	O	Y	TPPM		1
Next-Transaction	O	Y	TPPM		1

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
End-dialogue	O	Y	TPPM		1
Grant-control	O	Y	TPPM		1
Next-Transaction	O	Y	TPPM		1

Note: **1. The field is mandatory only if required by supported functional units, else it is not used.**

3.18 TP-PREPARE-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Data-permitted	O		Req		

TP-PREPARE-RI, Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Data-permitted	O		Ind		

3.19 TP-UNCHAIN-RI

No parameters

3.20 TP-BEGIN-TRANSACTION-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Chain	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Chain	M	Y	TPPM		

3.21 TP-ASSOCIATION-ESTABLISHMENT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		
Contention winner assignment	M	Y	TPPM		
Bid-Mandatory	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		
Contention winner assignment	M	Y	TPPM		
Bid-Mandatory	M	Y	TPPM		

3.22 TP-ASSOCIATION-ESTABLISHMENT-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		

4 ACSE SERVICE PARAMETERS

This section shows TP's use of ACSE services and parameters.

4.1 A-ASSOCIATE

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Mode	M	Y		
Application Context Name	M	Y		
Calling AP Title	M(A)			
Calling AE Qualifier	M(A)			
Calling AP Invocation Identifier	M			
Calling AE Invocation Identifier	M			
Called AP Title	C(A)			
Called AE Qualifier	C(A)			
Called AP Invocation Identifier	C(B)			
Called AE Invocation Identifier	C(B)			
Responding AP Title	M(A)			
Responding AE Qualifier	M(A)			
Responding AP Invocation Identifier	M(A)			

PARAMETER	STND	NIST	NIST RANGE	NOTES
Responding AE Invocation Identifier	M(A)			
User Information	M	Y		
Result	M	Y		
Diagnostic	O	O		
Calling Presentation Address	M	Y		
Called Presentation Address	M	Y		
Responding Presentation Address	O	O		
Presentation Context Definition List	M	Y		
Presentation Context Definition Result List	O	O		
Default Presentation Context Name	O	NU		
Default Presentation Context Result	O	NU		
Quality of Service	M	Y		
Presentation Requirements	M	Y	Kernel only	
Session Requirements	M	Y	Kernel + Full Duplex + CCR requirements (if used)	

PARAMETER	STND	NIST	NIST RANGE	NOTES
Initial Synchronization point Serial Number	M(A)			
Initial Assignment of Tokens	M(A)			
Session-Connection Identifier	NU	NU		

Note: **(A) Only if CCR is used, else parameter is a user option**
(B) Parameter becomes mandatory if the association is being established for

A-ASSOCIATE
Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Mode	M	Y		
Application Context Name	M	Y		
Calling AP Title	M(A)			
Calling AE Qualifier	M(A)			
Calling AP Invocation Identifier	M			
Calling AE Invocation Identifier	M			
Called AP Title	C(A)			
Called AE Qualifier	C(A)			

PARAMETER	STND	NIST	NIST RANGE	NOTES
Called AP Invocation Identifier	C(B)			
Called AE Invocation Identifier	C(B)			
Responding AP Title	M(A)			
Responding AE Qualifier	M(A)			
Responding AP Invocation Identifier	M(A)			
Responding AE Invocation Identifier	M(A)			
User Information	M	Y		
Result	M	Y		
Result Source	M	Y		
Diagnostic	O	O		
Calling Presentation Address	M	Y		
Called Presentation Address	M	Y		
Responding Presentation Address	O	O		

PARAMETER	STND	NIST	NIST RANGE	NOTES
Presentation Context Definition List	M	Y		
Presentation Context Definition Result List	O	Y		
Default Presentation Context Name	O	NU		
Default Presentation Context Result	O	NU		
Quality of Service	M	Y		
Presentation Requirements	M	Y	Kernel only	
Session Requirements	M	Y	Kernel + Full Duplex + CCR requirements (if used)	
Initial Synchronization Point Serial Number	M(A)			
Initial Assignment of Tokens	M(A)			
Session-Connection Identifier	NU	NU		

Note: (A) Only if CCR is used, else parameter is a user option
(B) Parameter becomes mandatory if the association is being established for recovery purposes (channels)

4.2 A-RELEASE

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Reason	NU	NU		
User information	NU	NU		
Result	M	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Reason	NU	NU		
User information	NU	NU		
Result	M	Y		

4.3 A-ABORT

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User Information	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Abort Source	M	Y		
User information	NU	NU		

4.4 A-P-ABORT

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Provider Reason	O	O		

5 PRESENTATION SERVICE PARAMETERS

This section shows TP's use of Presentation services and parameters.

5.1 P-TOKEN-PLEASE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens				1
User-data	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens				1
User-data	NU	NU		

Editor's Note: 1. Why is there an inconsistency in the token parameter of P-Token-Please and P-Token-Give.

5.2 P-TOKEN-GIVE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens	M	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens	M			

5.3 P-DATA

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	M	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	M	Y		

6 CCR SERVICE PARAMETERS

This section shows TP's use of CCR services and parameters.

6.1 C-BEGIN

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Atomic Action Id.-Master's Name	M	Y		
Atomic Action Id.-Suffix	M	Y		1
Branch Id.-Superior's Name	M	Y		
Branch Id.-Suffix	M	Y		1
User Data	C	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Atomic Action Id.-Master's Name	M	Y		
Atomic Action Id.-Suffix	M	Y		1
Branch Id.-Superior's Name	M	Y		
Branch Id.-Suffix	M	Y		1
User Data	C	Y		

Note: 1. Must decide which CCR ASN.1 Choice to use

6.2 C-PREPARE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

6.3 C-READY

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	NU	NU		

6.4 C-COMMIT

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

C-COMMIT, Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

6.5 C-ROLLBACK

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

6.6 C-RECOVER

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Recovery State	M	Y		
Atomic Action Identifier	M	Y		
Branch Identifier	M	Y		
User-data	C	Y		

C-RECOVER, *Receiving (Indication/Confirmation)*

PARAMETER	STND	NIST	NIST RANGE	NOTES
Recovery State	M	Y		
Atomic Action Identifier	M	Y		
Branch Identifier	M	Y		
User-data	C	Y		

Table of Contents

16 OFFICE DOCUMENT ARCHITECTURE

There is international alignment work progressing between the OIW, EWOS, and AOW on the Level 3 DAP (based on Chapter 16 in the Stable Document). As these alignment changes are completed, appropriate changes will be included in a revised Chapter 16. The current intention is to rename Chapter 16 to "Office Document Architecture Level 3 DAP."

The intention is to declare this revised work stable in December 1990.

Table of Contents

List of Figures

List of Tables

Foreword

This part of the Working Implementation Agreements for Open Systems Interconnection Protocols was prepared by the ODA Special Interest Group within the NIST OSI Implementors Workshop. The text defines a document application profile for the Open Document Architecture base standard defined by ISO 8613.

Development of this document application profile has been done in liaison with several organizations. These include ODA expert groups within the:

o Asia-Oceania Workshop (AOW);

o CCITT Study Group VIII;

o European Workshop on Open Systems (EWOS).

The liaison between these organizations has occured within the meetings of the Profile Alignment Group for ODA (PAGODA). These meetings have focused on the development of a single set of Internationally aligned ODA document application profiles.

This text reflects the output of the September 1990 meeting of the NIST ODA SIG, as ratified by the Plenary session of NIST OIW. This document is intended to be technically equivalent to the Revision 1 of the PAGODA FOD26 ODA DAP, as defined by the September 1990 PAGODA Editors Group Meeting in Geneva. In addition, this document is a specification of a document application profile supporting both the ASN.1 (ODIF) and the SGML (ODL/SDIF) encodings of ODA. The format of the document is intended to be consistent with the revised format for NIST OIW Style Guide.

Changes from the previous version of these agreements are annotated using REDLINE and STRIKEOUT facilities. New text is annotated as example new text. Text to be removed is annotated as ~~example deleted text~~. The text in clause 7 has been updated to reflect the current status of the DAP Proforma and Notation. Due to the significant number of notation changes, the change control markup was not used in clause 7.

0 NIST Level 2 ODA DAP

This is the definition of a single specification for two ~~an~~ ODA document application profiles (DAP) named FOD26. The two DAPs differ only in the encoding of the data stream. One uses the ASN.1 based ODIF encoding. The other uses the the SGML/SDIF based ODIF encoding. When this document refers to *this profile*, it is referring to either of the document application profiles defined by this specification.

This profile is suitable for interchanging documents in formatted form, processable form or formatted processable form and has been defined in accordance with ISO 8613-1. The format of this profile is in accordance with the standardized proforma and notation defined in Draft Addendum to ISO 8613-1 Annex F (to be published).

1 Scope and field of application

This profile specifies interchange formats for the transfer of structured documents between equipment designed for word or document processing. Such documents may contain character, raster graphics and geometric graphics content.

The documents that can be interchanged using this profile range from simple documents to highly structured technical reports, articles and typeset documents such as brochures. This profile provides a comprehensive level of features for the transfer of documents between these systems.

This profile allows documents to be interchanged in the following forms:

o formatted form;

o processable form;

o formatted processable form.

The architecture levels defined for these three forms have matching functionalities so that the interchange formats of a document are convertible from a processable form to any other form.

This profile is independent of the processes carried out in an end system to create, edit or reproduce documents. It is also independent of the means to transfer documents which for example, may be by means of communication links or storage media.

2 References

ISO 8613-1 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 1: Introduction and General Principles (1989).

ISO 8613-2 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 2: Document Structures (1989).

ISO 8613-4 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 4: Document Profile (1989).

ISO 8613-5 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 5: Open Document Interchange Format (1989).

ISO 8613-6 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 6: Character Content Architectures (1989).

ISO 8613-7 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 7: Raster Graphics Content Architectures (1989).

ISO 8613-8 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format - Part 8: Geometric Graphics Content Architectures (1989).

ISO 8613-1 - Information Processing - Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 1: DAD - A Document Application Profile Proforma and Notation (to be published).

CCITT Recommendation T.4 - Standardization of group 3 facsimile apparatus for document transmission (1988).

CCITT Recommendation T.6 - Facsimile coding schemes and coding control functions for group 4 facsimile apparatus (1988).

ISO 8859-1 - Information Processing - 8-bit Single-byte coded graphic character sets - Part 1: Latin alphabet No. 1.

ISO 6937-2 - Information Processing - Coded character sets for text communication - Part 2: Latin alphabet and non-alphabetic characters.

ISO 2022 - Information Processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.

ISO 8824 - Information Processing Systems - Open Systems Interconnection - Astract Syntax Notation One (ASN.1).

ISO 8825 - Information Processing Systems - Open Systems Interconnection - Basic encoding rules for abstract syntax notation one (ASN.1).

ISO 8879 - Information processing - Text and office systems - Standard Generalized Markup Language (SGML).

ISO 9069 - Information processing - SGML support facilities - SGML Document Interchange Format (SDIF).

ISO 9070 - Information processing - SGML support facilities - Registration procedures for public owner identifiers.

ISO/TR 9573 - Information processing - SGML technical report - Techniques for using SGML.

ISP FOD11 - Office document format profile (to be published).

ISP FOD26 - Office document format profile (to be published).

ISP FOD36 - Office document format profile (to be published).

3 Definitions and terminology

3.1 Definitions

The definitions given in ISO 8613-1 are applicable to this profile.

3.2 Constituent names

Each constituent that may be included in a document that conforms to this profile has been given a unique name which serves to identify that constituent throughout this profile.

The convention is that full names are used (i.e., no abbreviations are used), two or more words in a name are concatenated and each word begins with a capital. Examples of constituent names used in this profile are BodyText, Footnote, RectoPage and ColumnFixed.

In clause 6 of this profile, each constituent provided by this profile is underlined once at the point in the text at which the purpose of that constituent is defined. This also serves to identify all the constituents provided by this profile.

The same constituent names are also used in the technical specification in clause 7 of this profile so that there is a one-to-one correspondence between the use of these names in clause 6 and 7.

Although the constituent names relate to the purpose of the constituents, the semantics of constituents must not be implied from the actual names that are used. Also, these names do not appear in an interchanged document but a mechanism for identifying constituents in an interchange document is provided (see 6.6.4). Thus in an application using this profile, the constituents may known to the user by different names.

4 Relationship with other profiles

This profile belongs to a series of hierarchically related profiles which include FOD11 and FOD36.

The features supported by this profile are a superset of the features supported by the profile FOD11 and thus all data streams that are conformant to FOD11 are also conformant to this profile.

Also the features supported by this profile are a subset of those supported by the profile FOD36 and thus all data streams that are conformant to this profile are also conformant to FOD36.

5 Conformance

In order to conform to this profile, a data stream representing a document must meet the requirements specified in clause 5.1.

This profile does not define implementation ~~or service~~ requirements. These requirements are defined in other profiles that make use of this profile.

5.1 Data stream conformance

The following requirements apply to the encoding of data streams which conform to this profile:

o The data stream shall be encoded in accordance with the ASN.1 encoding rules defined in ISO 8825 or the SGML encoding rules defined in ISO 8879;

o The data stream shall be structured in accordance with the interchange formats defined in clause 8 of this profile;

o The document, as represented by the data stream after resolution of any external references, shall be structured in accordance with one of the documents architecture classes as defined in clause 6.1 of this profile and shall contain all mandatory constituents specified for that class; other constituents may be included, provided that they are permitted for that class, as specified in clause 7;

o Each constituent shall contain all those attributes specified as required for that constituent in this profile; other attributes may be specified provided that they are permitted for that constituent;

o The attribute values specified shall be within the range of permissible values specified in this profile;

o The encoded document shall be constructed in accordance with the abstract document architecture defined in ISO 8613-2;

o The document shall be structured in accordance with the characteristics and constraints specified in clause 6 of this DAP and shall contain only those features defined in clause 6.

5.2 Implementation conformance

This clause states the requirements for implementations claiming conformance to this profile.

~~An implementation claiming to originate and/or receive data streams conforming to this profile must complete a GSS and/or RSS proforma as defined in Annex [X] of this ISP.~~

A conforming receiving implementation must be capable of receiving any data streams conforming to this profile and usually, but not always, involves recognizing and further processing the data stream elements. The explicit meaning of "receiving" is determined by a Receiving Support Statement (RSS) ~~defined in accordance with Annex [X] of this ISP~~.

A receiving system which claims conformance to this DAP must be capable of handling data streams which are conformant to DAPs that are subordinate to this DAP within the taxonomy described in clause 4.

6 Characteristics supported by this document application profile

This clause describes the characteristics of documents which can be represented by data streams conforming to this profile. This clause also describes how these characteristics are represented in terms of constituent constraints.

6.1 Overview

This profile supports the interchange of documents in three forms, namely:

o processable form, which facilitates the revision of a document by a recipient;

o formatted form, which facilitates the reproduction of a document as intended by the originator;

o formatted processable form, which facilitates the reproduction of a document as intended by the originator or facilitates the revision of a document.

The constituents that may make up these three forms of document are defined in clauses 6.1.1, 6.1.2 and 6.1.3. Constituents defined as *required* must occur in any document that conforms to this profile. Constituents listed as *optional* may or may not be present in the document depending on the requirements of the particular document.

The constituents that make up a complete document that is conformant to this profile include all those referenced and contained in, if any, external and resource documents (see clauses 6.6.1 and 6.6.2).

6.1.1 Formatted documents

Required constituents:

o a document profile;

o layout object descriptions representing a specific layout structure.

Optional constituents:

o layout object class descriptions representing a 'factor' generic layout structure;

o presentation styles;

o content portion descriptions.

6.1.2 Processable form documents

Required constituents:

o a document profile;

o logical object class descriptions representing a 'complete' generic logical structure;

o logical object descriptions representing a specific logical structure.

Optional constituents:

o layout object class descriptions representing a 'complete' generic layout structure;

- o layout styles;
- o presentation styles;
- o content portion descriptions.

In the case of processable form documents, when the generic layout structure is not present, addition restrictions are placed on the layout directives that may be included in layout styles. These restrictions are defined in clause 7 of this profile.

6.1.3 Formatted processable documents

Required constituents:

- o a document profile;
- o logical object class descriptions representing a 'complete' generic logical structure;
- o logical object descriptions representing a specific logical structure;
- o layout object class descriptions representing a 'complete' generic layout structure;
- o layout object descriptions representing a specific layout structure.

Optional constituents:

- o layout styles;
- o presentation styles;
- o content portion descriptions.

6.1.4 Generic documents

A generic document consists of one of the following sets of constituents:

either a)

- o a document profile;

o logical object class descriptions which represent a *complete* generic logical structure;

o layout styles whose presence are optional;

o presentation styles whose presence are optional;

o generic content portions whose presence are otpional.

or b)

o a document profile;

o layout object class descriptions which represent a *complete* generic layout structure or *factor set*;

o presentation styles whose presence are optional;

o generic content portions whose presence are optional.

or c)

o a document profile;

o logical object class descriptions which represent a *complete* generic logical structure;

o layout class descriptions which represent a *complete* generic layout structure;

o layout styles whose presence are optional;

o presentation styles whose presence are optional;

o generic content portions whose presence are optional.

6.2 Logical characteristics

This clause defines the logical constituent constraints provided by this profile to represent the characteristics of documents.

Different constituent constraints may be used to represent and distinguish parts of a document that have different logical characteristics. This clause describes the general characteristics and typical uses of the constituent constraints that are provided.

The descriptions of the logical characteristics represented by each of the constituent constraints is provided for guidance only. It is the responsibility of the user to determine how a document is to be represented using the constituents provided. Adherence to these guidelines may enhance the mutual understanding of a document by an originator and a recipient.

6.2.1 Overview of the logical structure

From the logical point of view, the document consists of two parts, namely a *body* part and a *common* part.

The body part represents main content of a document and is intended to be reproduced in the body area of the pages that make up the document. The body part must be included in all documents that are interchanged in accordance with this profile.

The common part represents common content that is to be placed in reserved header and footer areas on each page of a document. Header and footer content are independently optional and so may be included in an interchanged document only if required.

6.2.2 Body part of logical structure

6.2.2.1 DocumentLogicalRoot

DocumentLogicalRoot is a constituent constraint representing the top level in the document logical structure. Its immediate subordinates consist of a sequence of one or more constituent constraints of the type Passage.

The automatic numbering schemes that apply to constituent constraints of the types NumberedSegment and Footnote may be initialised on the DocumentLogicalRoot.

6.2.2.2 Passage

Passage is a constituent constraint that represents the first level of logical sub-division of a document. It may be used to indicate a logical grouping of subordinate parts of a document that are to be regarded as an entity for reading or that have common layout and presentation characteristics. For example:

o the contents to be placed on the title page of a report;

o the front matter in the table of contents or foreword;

o the main matter of the document;

o the back matter, consisting of appendices, glossary or index.

The automatic numbering schemes that apply to subordinate constituent constraints of the types NumberedSegment and Footnote may be initialised on a Passage.

The immediate subordinates of a Passage consist of an optional arbitrary ordered sequence of one or more of the following constituent constraint types:

o Paragraph;

o BodyGeometric;

o BodyRaster;

o BodyText.

These may be optionally followed by one or more constituent constraints of the type NumberedSegment.

A Passage must at least have one of the above constituent constraint types as a subordinate.

A document may contain several different class definitions of the type Passage, each of which defines the common characteristics of sets of Passages within the document such as their allowed subordinates or layout properties. For example, a class of Passages may be defined which always begin on a new page set.

6.2.2.3 NumberedSegment

NumberedSegment is a constituent constraint that represents a logical subdivision of a Passage or another higher level NumberedSegment. It is used to represent the grouping of parts of a document that are distinguished by an identifier. These parts may also have some common layout characteristics.

The automatic numbering schemes that apply to subordinate constituent constraints of the types NumberedSegment and Footnote may be initialised on a Passage.

The immediate subordinates of a NumberedSegment consist of the constituent constraint Number, whose presence is mandatory and serves to carry the identifier of the NumberedSegment. This is followed by an optional arbitrary ordered sequence of one or more of the following constituent constraint types:

- Paragraph;
- BodyGeometric;
- BodyRaster;
- BodyText.

These are optionally followed by a sequence of one or more constituent constraints of the type NumberedSegment. Hence a document may contain any number of nested levels of the constituent NumberedSegment.

A NumberedSegment is typically used to represent entities such as chapters, sections, nested sub-sections and appendices which contain an identifier that serves to distinguish that entity for human comprehension.

A document may contain any number of different class definitions of NumberedSegment which define the common characteristics of sets of NumberedSegments, such as their allowed subordinates and layout properties.

Class definitions of NumberedSegments cannot be recursively defined. That is, a NumberedSegment at one level cannot refer to a NumberedSegment at a higher level and there must be one NumberedSegment definition for each level of NumberedSegment in the specific logical structure of a document.

6.2.2.4 Number

Number is a constituent constraint that represents the identifier of a NumberedSegment to which it is subordinate. This identifier allows the NumberedSegment to be distinguished within the document for machine processing or human comprehension.

A Number is a basic logical constituent which contains a content generator which, when evaluated, produces the identifier referred to above. This evaluation takes place during the layout process.

The identifiers are structured and consist of sequence of one or more numerals that allow NumberedSegments at the same or different levels in a document structure to be uniquely distinguished. The numerals may be represented by Arabic or Roman numerals or by their alphabetic equivalent in lower or upper case characters (the number 1 is represented by *A* etc.). Each numeral in an identifier may be distinguished by means of *separators* characters such as spaces and full stops; a typical example is *6.2.3.4*.

Further details of the structure and generation of the identifiers are given in 6.6.7.

6.2.2.5 Paragraph

Paragraph is constituent constraint that is a subdivision of a Passage or NumberedSegment. It is typically used to represent the grouping of parts of a document that deals with a single theme or topic. These parts may consist of character, raster graphics and geometric graphics content.

The immediate subordinates of a Paragraph consist of an arbitrary ordered sequence of one or more of the following constituent constraints:

o BodyText;

o BodyRaster;

o BodyGeometric;

o Footnote.

Constituents of the type BodyText may be *concatenated* to form a continuous stream of character content which is laid out as a single unit. Sequences of constituents of the types BodyText and Footnote may be concatenated to represent a stream of character content with embedded footnotes. Multiple embedded footnotes, which may be consecutive without intervening text, may be included in the content. Alternatively, the character content may contain hard new line controls, which will cause parts of the content to be separated when laid out.

Another typical use of a Paragraph is to represent a group of document parts that have common layout characteristics. An example is a graphical illustration with associated text which is to be laid out in a particular frame.

6.2.2.6 BodyText, BodyRaster and BodyGeometric

BodyText, BodyRaster and BodyGeometric are constituent constraints which represent the lowest level of logical subdivision of a document. These constituent constraints are subdivisions of Passages, NumberedSegments and Paragraphs. They allow the layout and presentation requirements of different parts of a document to be specified.

These are basic logical constituents that directly refer to content portions that contain character, raster graphics and geometric graphics content respectively. BodyText may refer to one or more content portions each containing processable, formatted or formatted processable character content. BodyRaster and BodyGeometric may only refer to a single content portion containing formatted processable raster graphics content or formatted processable geometric graphics content respectively.

Constituents of these types in the generic logical structure may refer to generic content. This provides the means of defining common content within the body part of a document.

6.2.2.7 Footnote

Footnote is a constituent constraint that is a subdivision of a Paragraph and is used to represent footnotes within a document.

A footnote is an amount of content that is logically associated with a particular part of the document body but which is intended to be read and laid out separately from its associated part of the document. Typically, a footnote consists of a footnote identifier, which is embedded within the document body, and the footnote itself, which is laid out elsewhere.

A Footnote is a composite logical constituent whose immediate subordinates consist of the constituent constraint FootnoteReference, which represents the footnote identifier, followed by the constituent constraint FootnoteBody, which represents the footnote itself. Both of these subordinates are mandatory.

6.2.2.8 FootnoteReference

FootnoteReference is a constituent constraint that is used to represents an identifier that provides a footnote reference within the body of a document.

FootnoteReference is a basic logical constituent that contains a content generator which when evaluated produces the identifier referred to above. ~~This identifier may be automatically generated or may be manually inserted by the user. When automatically generated identifiers are used, the identifier consists of a single number which may be represented in the form of Arabic or Roman numerals or by its alphabetic equivalent. The initialisation of the identifiers takes place at the passage level. If the identifier is manually inserted, then it may consist of any character string.~~

This character string consists of a label with optional prefix and suffix character strings. The label is used to uniquely identify a particular footnote and may consist of a number which is represented in the form of Arabic or Roman numerals or by an alphabetic equivalent. The number may be automatically generated so that its value is incremented for each successive footnote. Alternatively, the label may consist of a user defined character string.

The format of the content generator referred to above is described in clause 6.6.8.

6.2.2.9 FootnoteBody

FootnoteBody is a constituent constraint which represents the content of a footnote.

FootnoteBody is a composite logical constituent whose subordinates consist of the constituent constraint FootnoteNumber, which is mandatory and represents the footnote identifier, followed by one or more constituent constraints of the type FootnoteText which represents the footnote content. The identifier referred to above is identical to the corresponding footnote identifier which is embedded in the content of the document body and represented by the constituent constraint FootnoteReference.

The constituents subordinate to FootnoteBody are intended to be laid out separately from the other parts of the document content. When a generic layout structure is specified for the document, these constituents are constrained to be laid out in a FootnoteArea frame (see 6.3.5.8).

6.2.2.10 FootnoteNumber

FootnoteNumber is a constituent constraint that represents the footnote identifier within the footnote body.

This identifier is identical to the content associated with the constituent constraint FootnoteReference but is intended to be laid out so that it immediately precedes the content of the footnote body.

FootnoteNumber is a basic logical constituent that contains a content generator which when evaluated produces the identifier referenced above. The format of this content generator is the same as the content generator that may be specified for the constituent constraint FootnoteReference.

It is required to specify the layout category name *Footnote* for this constituent; this ensures that this constituent is laid out in a FootnoteArea frame when a generic layout structure is specified within the document.

6.2.2.11 FootnoteText

FootnoteText is a constituent constraint that is used to represent the footnote content. It is the lowest logical subdivision of a FootnoteBody.

FootnoteText is a basic logical constituent that references one or more content portions each containing processable, formatted or formatted processable character content.

It is required to specify the layout category name *Footnote* for this constituent; this ensures that this constituent is laid out in a FootnoteArea frame when a generic layout structure is specified within the document.

6.2.3 Common content part of the logical structure

6.2.3.1 CommonContent

CommonContent is a constituent constraint that represents common content that is to be laid out in the header and footer areas of the pages of a document. Common content may consist of any combination of character, raster graphics and geometric content.

Any number of constituent constraints of the type CommonContent may be contained in a document. CommonContent is a composite logical object class whose immediate subordinates consist of an arbitrary ordered sequence of one or more of the following constituent constraints:

o CommonText;

o PageNumber

o CommonRaster;

o CommonGeometric.

When the generic layout structure is present, constituents of the type CommonContent and their associated subordinate object classes are constrained to be laid out in frames representing header or footer areas using the 'logical source' mechanism (see 6.3.6).

6.2.3.2 CommonText

CommonText is a constituent constraint that represents the common character content that is to be laid out in the header or footer area of a document. For example, running header or footer ~~text can be represented by this constituent~~ content that appears on each page in a sequence of pages can be represented by this constituent.

CommonText is a basic logical object class that references one or more content portions each containing processable, formatted and formatted processable character content.

6.2.3.3 PageNumber

PageNumber is a constituent constraint that represents common character content that is to be laid out in a the header or footer area of a document. This constituent is specifically used when it is required to present a running header or footer which contains an automatically generated page number.

PageNumber is a basic logical object class that contains a content generator. This content generator contains a reference to a page number which is automatically evaluated when the document is laid out. This provides the means of representing the page numbers that are displayed on the consecutive pages of a document.

Each page number consists of a single number which may be represented in the form of Arabic or Roman numerals or in its alphabetic equivalent. Page numbering schemes may ~~be initialised to 0 or greater at the document root or page set level~~ can start at 0 or any value greater than 0.

The format of the content generators is defined in clause 6.6.6.

6.2.3.4 CommonRaster

CommonRaster is a constituent constraint that represents the common raster graphics content that is to be laid out in the header or footer area of a document. For example, this constraint may be used to represent a logo which is to be laid out on each page of a document.

Common raster is a basic logical object class which references a single content portion containing formatted processable raster graphics content.

6.2.3.5 CommonGeometric

CommonGeometric is a constituent constraint that represents the common geometric graphics content that is to be laid out in the header or footer area of a document. For example, this constraint may be used to represent a graphical icon which is to be laid out on each page of a document.

CommonGeometric is a basic logical object class which references a single content portion containing formatted processable geometric graphics content.

6.3 Layout characteristics

This clause defines the layout constituent constraints provided by this profile to represent the characteristics of documents.

Different constituent constraints may be used to represent and distinguish parts of a document that have different layout characteristics. This clause describes the general characteristics and typical uses of the constituent constraints that are provided.

The descriptions of the layout characteristics represented by each of the constituent constraints is provided for guidance only. It is the responsibility of the user to determine how a document is to be represented using the constituents provided. Adherence to these guidelines may enhance the mutual understanding of a document by an originator and a recipient.

6.3.1 Overview of layout characteristics

The document structure allows the document content to be laid out and presentation of in one or more page sets. Each page set may be used for different parts of the document, for example, the title page, foreword, table of contents, document body and appendices.

Each page set consists of a series of pages. In general, each page may be sub-divided into three areas; the body area, which is used to layout the document body, and the header and footer areas, which may be used to layout the common content.

Four page layout types are supported by this profile. Each page layout type specifies how the body, header and footer areas are positioned within each page and how the content may be presented within each of those areas. These four types are referred to as page layouts A, B, C and D and are illustrated in Figures 1, 2, 3 and 4 respectively.

It is intended that all applications which use this profile should support page layout A, whereas support for the other three page layouts may be specified as optional.

Page layout A is used when the character content is to be laid out horizontally (from left to right or from right to left) and from top to bottom within the body area. This layout is typically used for documents written in Latin based, Hebrew and Arabic languages.

Page layout B is used when the character content is to be laid out vertically (bottom to top or top to bottom) and from left to right within the body area. This layout is typically used for documents written in Latin based, Hebrew and Arabic languages in which it is required to layout the content in landscape orientation within the body area of the page.

Page layouts C and D are used when the character content is to be laid out vertically and from right to left within the body area. These layouts are typically used in documents written in languages which use ideograms, such as Japanese and Chinese characters.

The body area may be further sub-divided into areas composed of single and multiple columns and an area may be reserved for footnotes. In addition, the header and footer areas may be sub-divided to allow the representation of different content types.

6.3.2 DocumentLayoutRoot

DocumentLayoutRoot is a constituent constraint that represents the top level in the document layout structure. Its immediate subordinates consist of a sequence of one or more constituents of the type PageSet. The numbering schemes for pages can be initialised on this constituent constraint.

6.3.3 PageSet

PageSet is a constituent constraint that represents a grouping of pages within a document. A PageSet is typically used to represent a part of a document that has different layout requirements from other parts of a document. Also, a PageSet may correspond to a part of a document that has a certain logical significance, for example, a PageSet might represent the front matter in a document or an individual chapter.

Only one level of PageSet is allowed in a document. However, a document may contain any number of class definitions of the type PageSet which may be used, for example, to provide a choice of alternative layouts for different parts of a document or to specify the exact layout requirements for each successive part of a document.

The immediate subordinates of a PageSet consist of a combination of constituent constraints of the types Page, RectoPage and VersoPage, as described in clause 6.3.4.1.

6.3.4 Page characteristics

6.3.4.1 Page constituents

Three constituent constraints are provided to represent the pages within a document, namely Page, RectoPage and VersoPage.

The only difference in the characteristics of these page types concerns the values can be specified for the parameter *side of sheet* in the attribute *Medium type.* In the case of Page, the value of this parameter may be specified as *recto*, *verso* or *unspecified*. In the case of RectoPage, the value of this parameter may be *recto* or *unspecified*; and in the case of VersoPage, the value of this parameter ~~must be specified as *recto* and *verso* respectively~~ may be *verso* or *unspecified*.

The pages that make up a page set consist of an optional initial page which is represented by the constituent constraint Page and which is optionally followed by either:

- o A sequence of pages represented by the constituent constraint Page. All pages in this sequence must have the same layout characteristics (see note~~1~~) but these characteristics may differ from those of the initial page.

- o A sequence of pages which are intended to be laid out alternatively on the *recto* and *verso* (or on the *verso* and *recto*) sides of the presentation medium and are represented by the constituent constraints RectoPage and VersoPage respectively. All pages in this sequence must have the same layout characteristics (see note~~1~~) but these characteristics may differ from those of the initial page.

A page set must contain at least one page.

An initial page is typically used at the beginning of a document or of a section within a document. It may be used, for example, for a title page whose layout requirements differ from the following pages.

The following restrictions also apply to the pages within a page set:

- o all the pages must have the same dimensions and orientation (see 6.3.4.2);

- o all pages are to be laid out on the same size of presentation medium (see 6.3.4.3).

Note: The layout characteristics of pages are specified in 6.3.4.5. Pages having the same layout characteristics are pages for which the body area, header area (if present) and footer area (if present) have the same dimensions and position within the page (see 6.3.4.3). However, pages having the same layout characteristics do not necessarily have the same position on the presentation medium (see 6.3.4.4).

6.3.4.2 Page dimensions

The dimensions of the pages may be specified as any value (in BMUs) that is equivalent to or less than ISO A3 or ANSI B paper sizes in portrait or landscape orientations. The dimensions may be specified in portrait or landscape orientation.

Dimensions equivalent to or less than the common assured reproduction area of ISO A4 and North American Letter (NAL) in portrait or landscape orientation are basic values. Larger page sizes are non-basic and their use must be indicated in the document profile.

Any default page dimensions may be specified in the document profile subject to the maximum dimensions defined above.

6.3.4.3 Nominal page sizes

The nominal page sizes that may be specified are listed in Table 1. These may be specified in portrait or landscape orientations. All values of nominal page size are non-basic and hence all values used in a document must be indicated in the document profile.

Any of nominal page size defined in Table 1, subject to the restrictions specified above, may be specified as the default value in the document profile.

Table 1 also includes the recommended assured reproduction area (ARA). Information loss may occur when a document is reproduced if the dimensions of constituents of the type page exceed the ARA for the specified nominal page size.

Table 1 - Nominal page sizes

Page type	Size (in or mm)	Size (BMU)	ARA (BMU)
ISO A5	148mm x 210mm	6922 x 9920	not defined
ISO A4	210mm x 297mm	9920 x 14030	9240 x 13200
ISO A3	297mm x 420mm	14031 x 19843	13200 x 18480
ANSI legal	8.5in x 14in	10200 x 16800	9240 x 18480
ANSI A	8.5in x 11in	10200 x 13200	9240 x 12400
ANSI B	11in x 17in	13200 x 20400	12744 x 19656
Japan-legal	257mm X 364mm	12141 x 17196	11200 x 15300
Japan-letter	182mm x 257mm	8598 x 12141	7600 x 10200

6.3.4.4 Page offset

The page offset is the distance of the position of the left and top edges of the page relative to the left and top edges respectively of the presentation medium on which each page is reproduced. Any value of page offset may be specified provided that no part of the page area lies outside the area of the nominal page. Also, page offsets specified for the initial, recto and verso pages within a given page set may differ. The default page offset may be specified in the document profile.

6.3.4.5 Page layout characteristics

Each page in a document may be subdivided into three rectangular areas, as follows:

o a body area which is reserved for content that belongs to the body part of the document (as defined in 6.3.5);

o a header area which is reserved for common header content (see 6.3.6);

o a footer area which is reserved for common footer content (see 6.3.6).

The body area is mandatory and must occur on every page in a document. The header and footer areas are both optional.

Also these three areas must be entirely contained within the page area and must not overlap.

Four types of page layout are supported as defined below.

6.3.4.5.1 Page layout A

For page layout A the header footer area are placed above and below the body area. The layout paths in the header, body and footer areas are specified as 270 degrees. This type of layout is illustrated in Figure 1.

6.3.4.5.2 Page layout B

For page layout B the header footer area are placed above and below the body area. The layout path in the body area is specified as 0 degrees; in the header and footer areas the layout paths are specified as 270 degrees. This type of layout is illustrated in Figure 2.

6.3.4.5.3 Page layout C

For page layout C the header footer area are placed above and below the body area. The layout path in the body area is specified as 180 degrees; in the header and footer areas, the layout paths are specified as 270 degrees. This type of layout is illustrated in Figure 3.

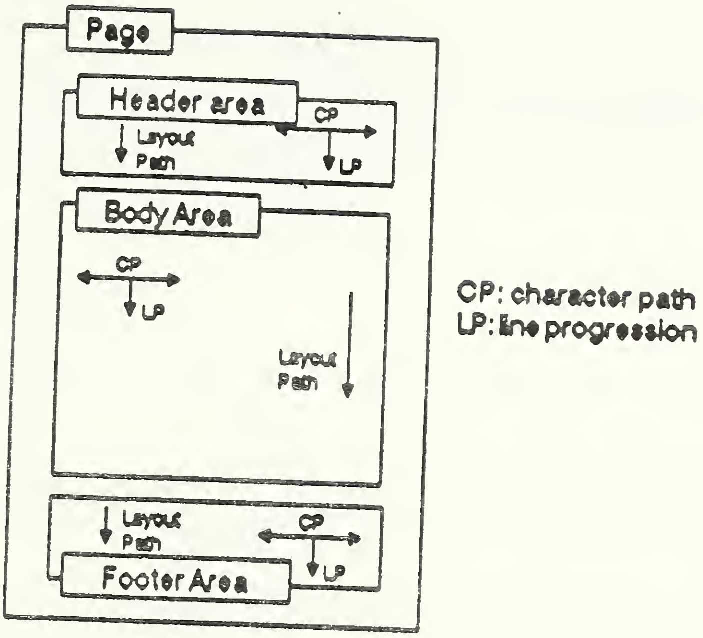


Figure 1 - Page layout type A

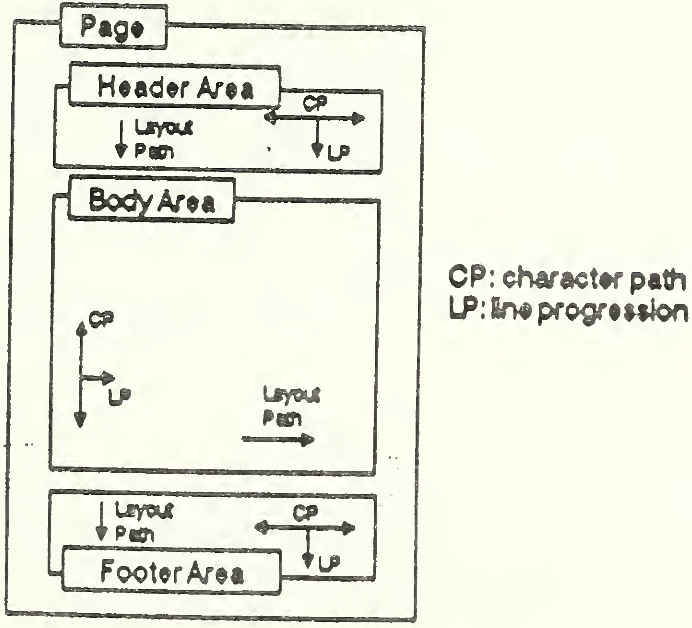


Figure 2 - Page layout type B

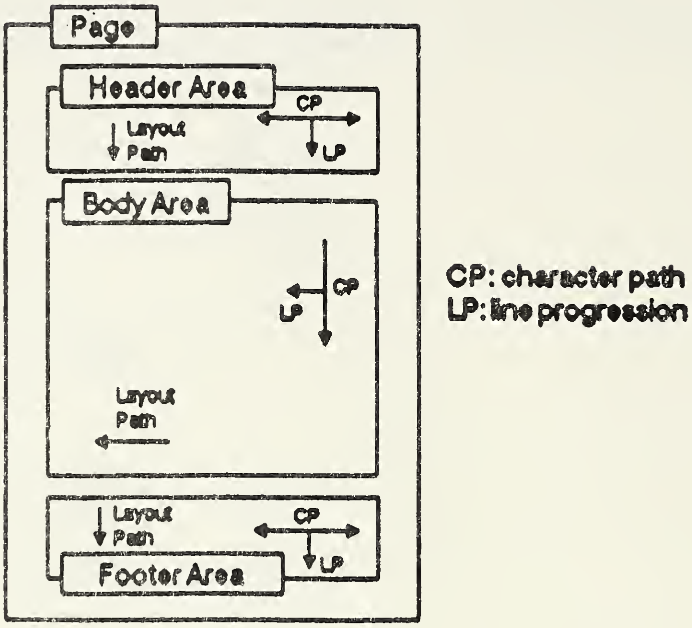


Figure 3 - Page layout type C

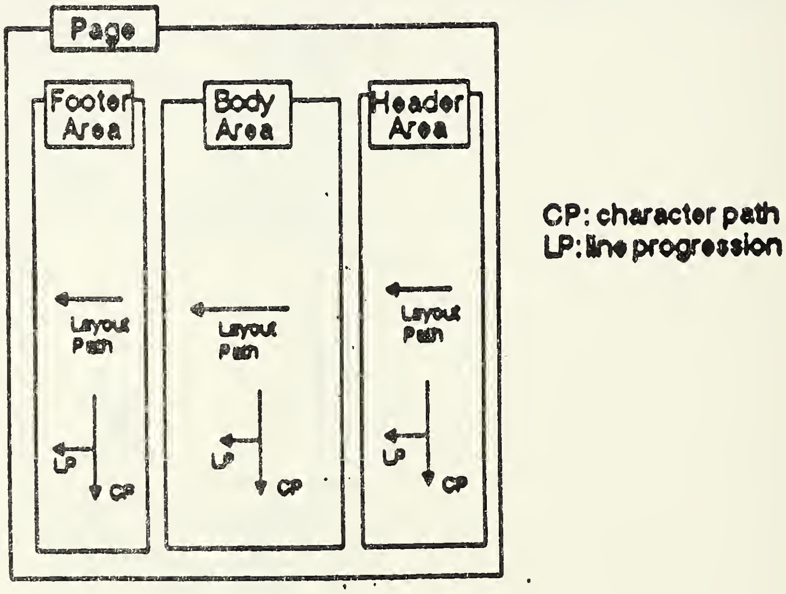


Figure 4 - Page layout type D

6.3.4.5.4 Page layout D

For page layout D the header and footer areas are placed to the right and left of the body area respectively. The layout paths in the header, body and footer areas are all specified as 180 degrees. This type of layout is illustrated in Figure 4.

6.3.5 Body area characteristics

The body area is the area within a page where the main matter of the document, that is the *body* part of the document, is laid out.

The body area may consist of a single frame into which the content is directly laid out. In this case, the body area is represented by a BasicBody frame.

Alternatively, the body area may be subdivided into different rectangular regions to provide for combinations of single or multiple column layout and the layout of footnotes. In this case, the body frame is represented by a VariableCompositeBody frame.

6.3.5.1 BasicBody

BasicBody is a constituent constraint which defines a lowest level frame into which content is directly laid out.

The position and dimensions of this frame are fixed. The layout path specified depends upon the page layout type being used (see 6.3.4.5).

6.3.5.2 VariableCompositeBody

VariableCompositeBody is a constituent constraint that defines a composite frame which contains one or more subordinate variably positioned frames. A VariableCompositeBody frame has a fixed position and fixed dimensions. The layout path specified for this frame depends upon the page layout type being used (see 6.3.4.5).

The immediate subordinates of frames of this type consist of an arbitrary ordered sequence of one or more frames of the following types:

o BasicFloat;

o SnakingColumns;

o SynchronizedColumns.

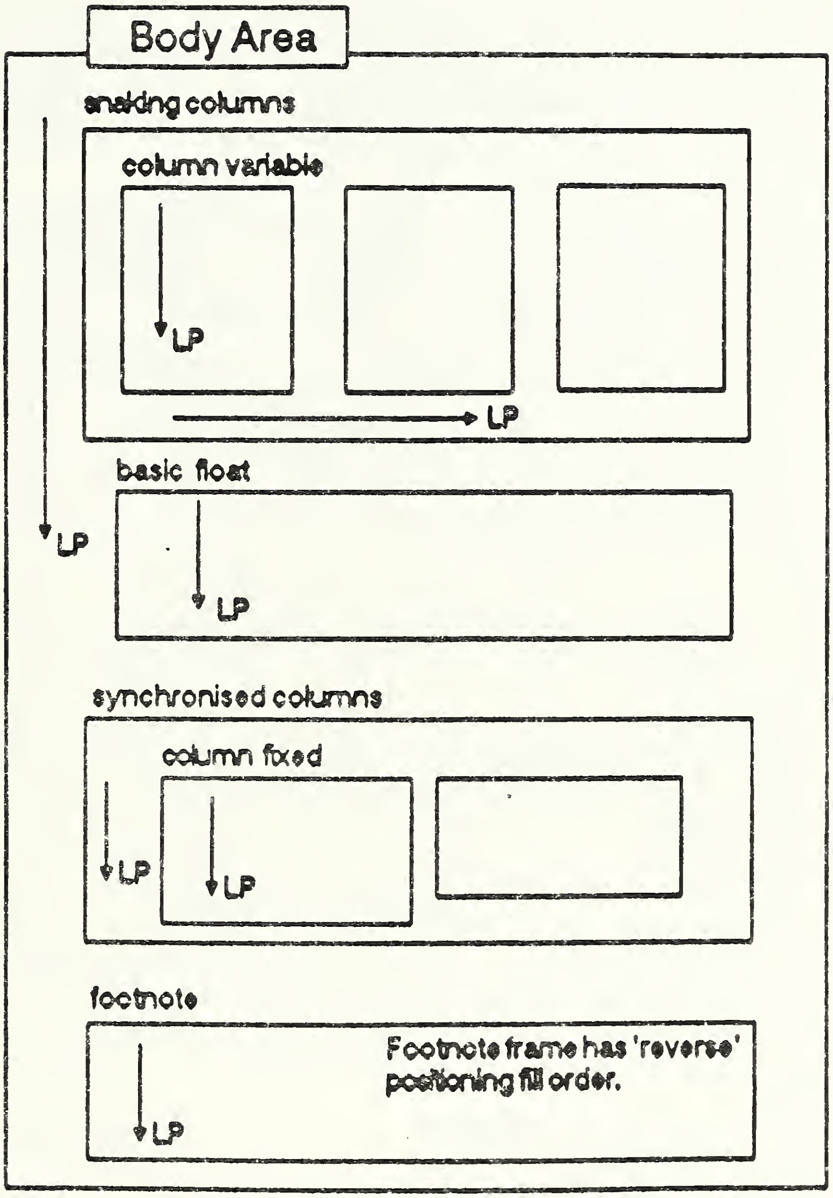
It may also contain a single frame of the type FootnoteArea.

The subordinate frames are all variably positioned and have variable dimensions. Frames of the type BasicFloat, SnakingColumns and SynchronizedColumns are laid out in the direction of the layout path of the body area (i.e., their positioning fill order is *normal*), whereas FootnoteArea frames are laid out in the direction opposite to that of the body area layout path (i.e., in *reverse* positioning fill order).

Thus the relative positions of these frames in the body area may vary and depend upon the positions of other frames (if any) that are placed in the same body area.

Figures 5, 6 and 7 provide illustrations of the layout of frames within a VariableCompositeBody frame for the various page layout types.

A choice of subordinate frames of the types listed above may be specified for a VariableCompositeBody frame. Different frame types can be selected using various layout directives (see 6.4) and hence the layout characteristics of the body areas within a page set may change from page to page within a page set.



Note: LP = layout path

Figure 5 - Example of body area layout for page layout A

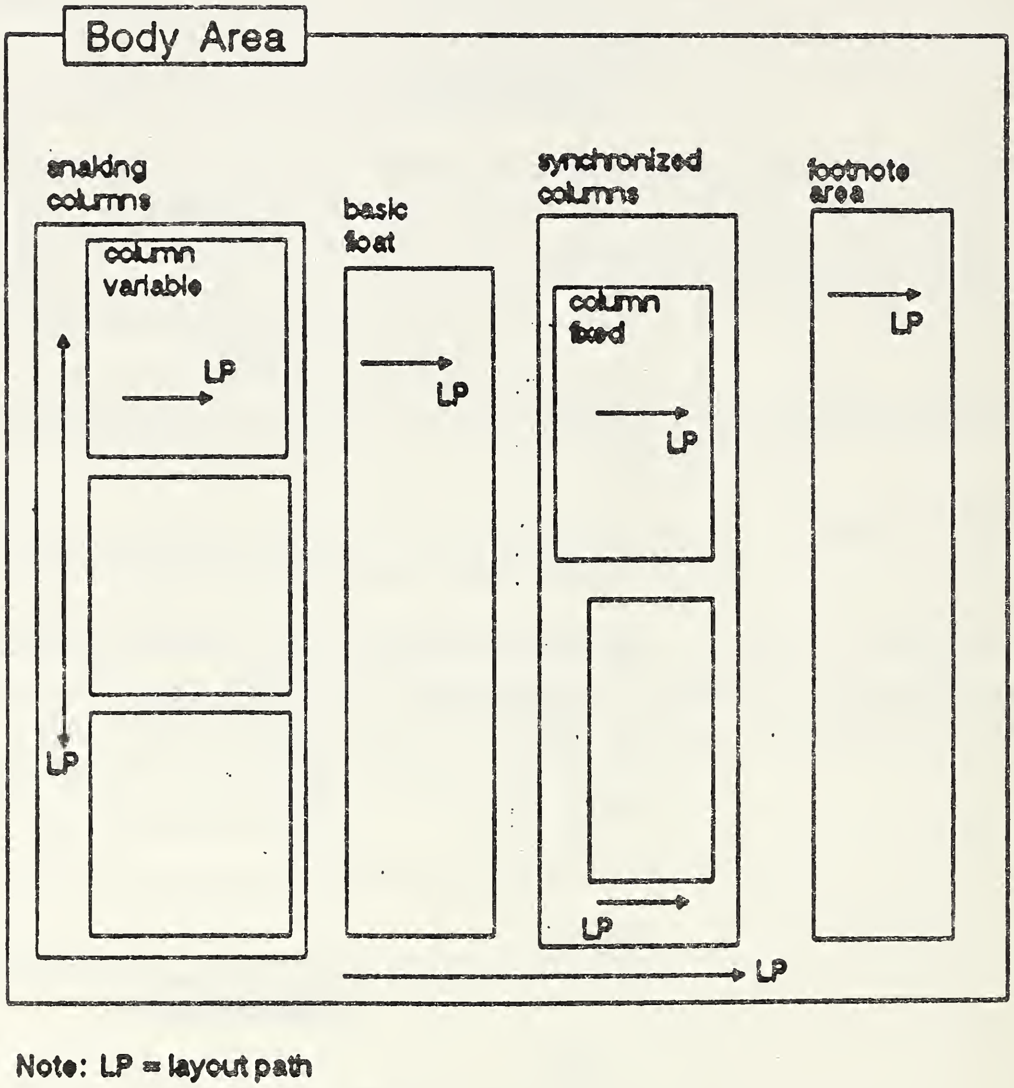
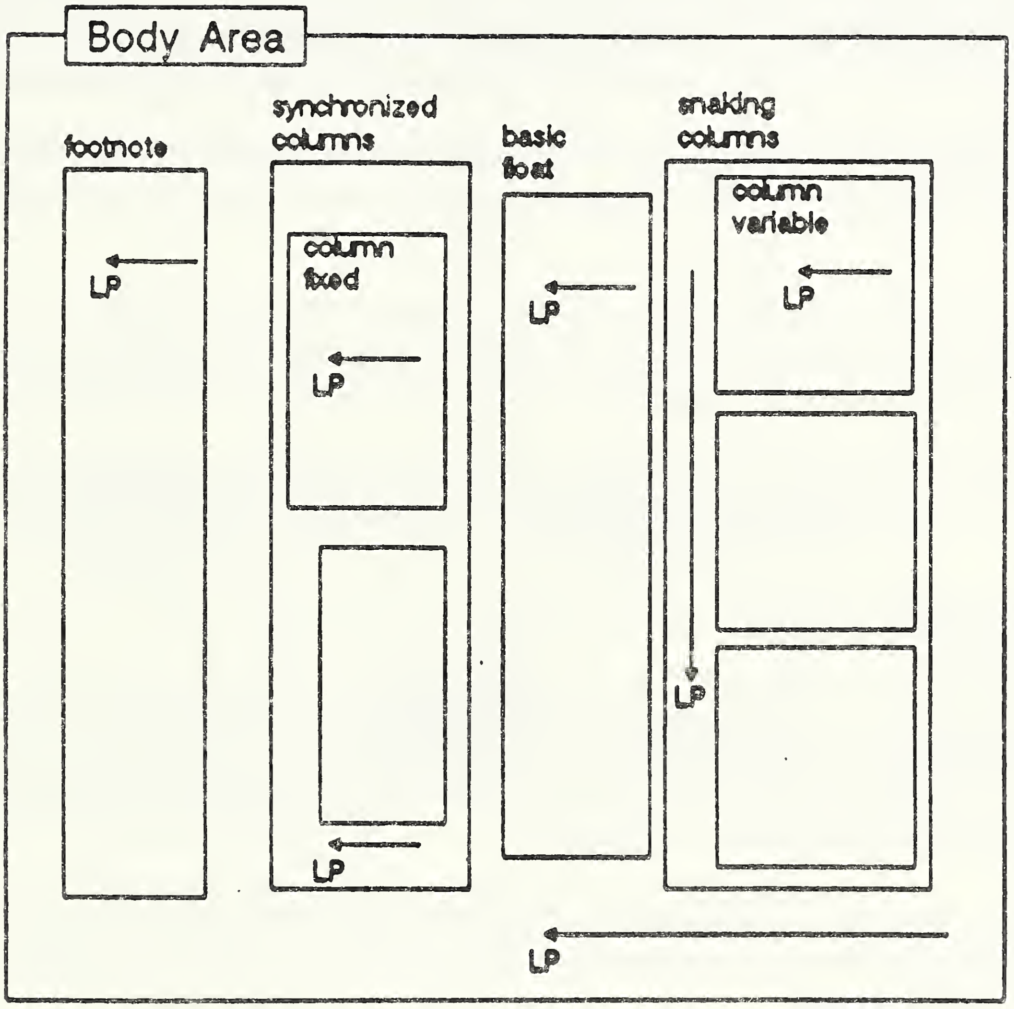


Figure 6 - Example of body area layout for page layout type B



Note: LP = layout path

Figure 7 - Example of body area layout for page layout types C and D

6.3.5.3 BasicFloat

BasicFloat is a constituent constraint that defines a lowest level frame that is used to represent a single column area within a body area. A single column area is typically used to layout content in the form of a single column. This is a variably positioned frame.

The dimension of the edge of this frame which is orthogonal to the direction of the layout path of the body area is fixed or defaults to the maximum value allowed within the body area.

The dimensions of the edge in the direction parallel to the layout path of the body area is specified as *rule-b*. This dimension is therefore automatically adjusted during the layout process to be the minimum required to contain all the content allocated to the frame.

The layout path specified for this frame is the same as that specified for the body area. Content may only be laid out in this frame in the direction of the layout path specified.

6.3.5.4 SnakingColumns

SnakingColumns is a constituent constraint that defines a composite frame that represents a synchronized column area within a body area. A synchronized columns area is typically used for the layout of one or more columns of content in which the content is allowed to flow freely from one column to the next.

This is frame which is variably positioned. Its immediate subordinates consist one or more frames of the type ColumnVariable. Examples of the layout of SnakingColumns frames are given in Figure 8 and 9.

The dimension of the edge of a SnakingColumns frame which is orthogonal to the direction of the layout path of the body area is fixed or defaults to the maximum value allowed within the body area.

The dimensions of the edge of this frame in the direction parallel to the layout path of the body area is specified as *rule-b*. This dimension is therefore automatically adjusted to accommodate the subordinate frames which are laid out in it.

The layout path for a SnakingColumns frame may be specified as 0 or 180 degrees in the case of page layout A, 90 or 270 degrees in the case of page layout B, and 270 degrees in the cases of page layouts C and D.

The attribute "Balance" may be specified for a SnakingColumns frame to indicate that two of more of the subordinate ColumnVariable frames are to be equal in length in the vertical dimension in the case of page layout A and equal in length in the horizontal dimension in the cases of page layouts B, C and D (see note~~1~~).

Note: It is intended that the attribute "Balance" may be ignored when the subordinate ColumnVariable frames have unequal widths.

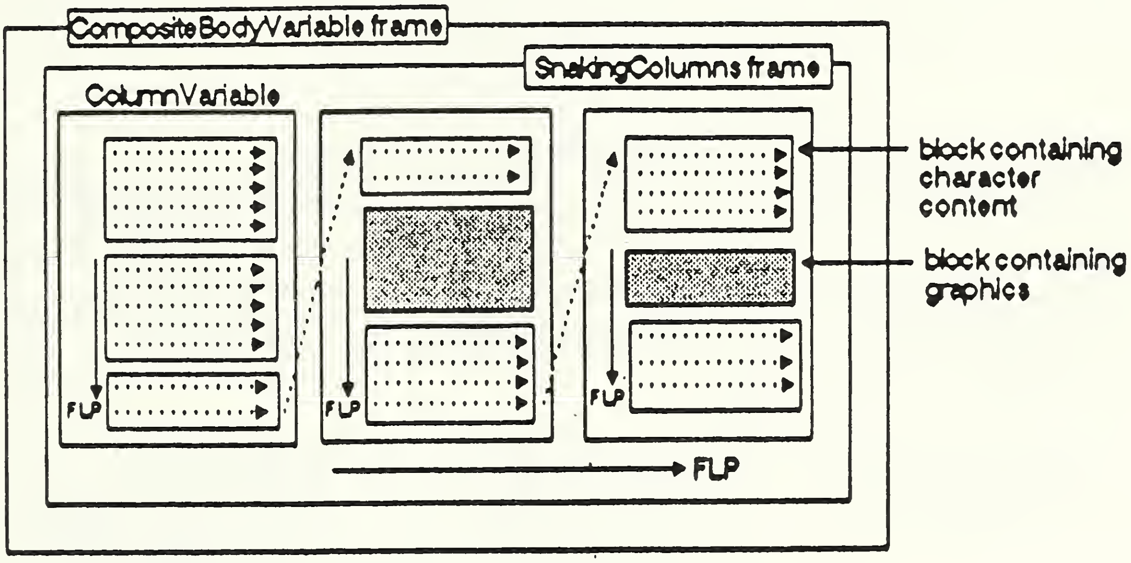
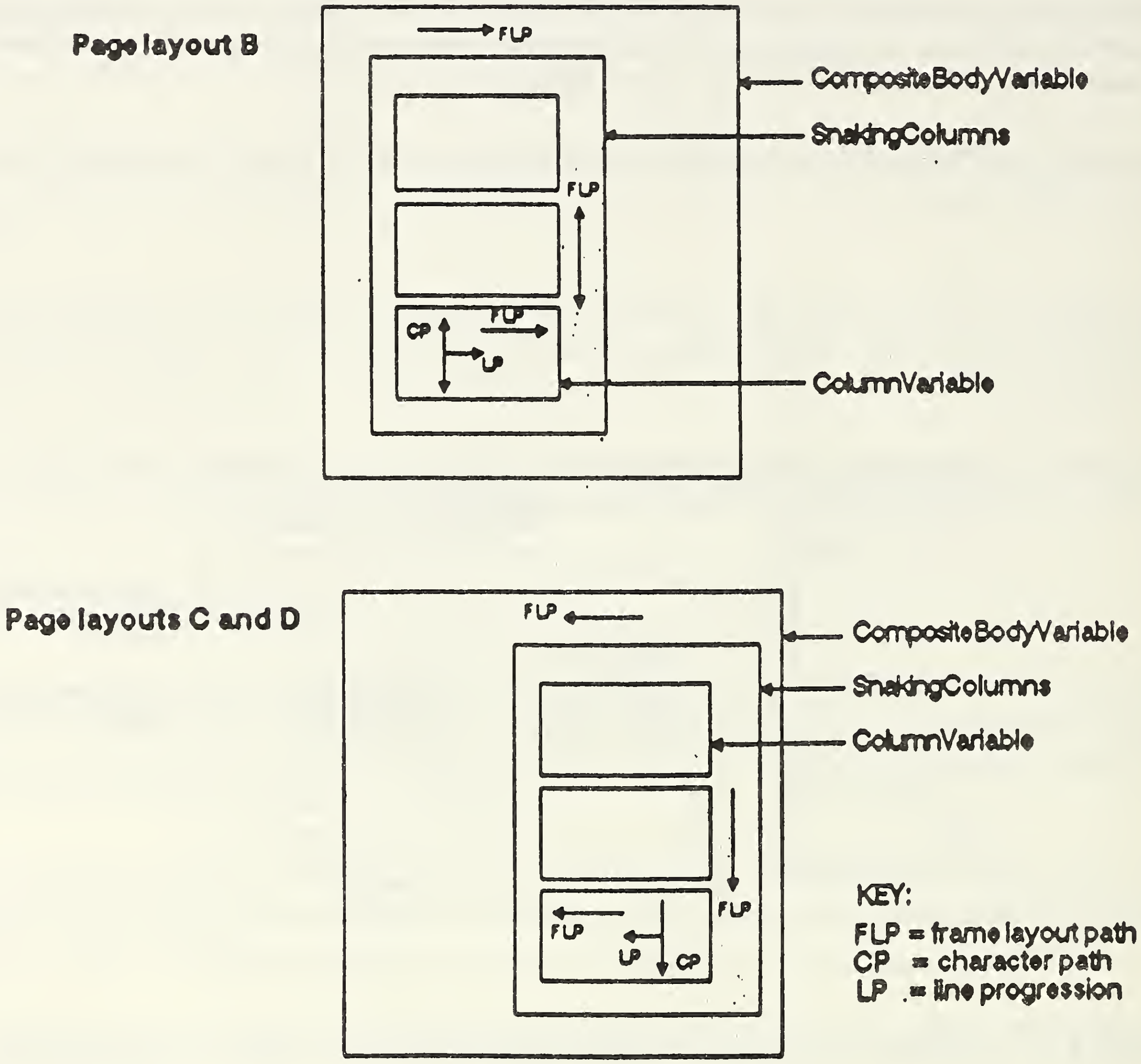

CompositeBodyVariable frame
SnakingColumns frame
ColumnVariable
FLP
FLP
FLP
FLP
block containing character content
block containing graphics

Figure 8 - Example of layout of SnakingColumns for page layout type A


Page layout B

Page layouts C and D

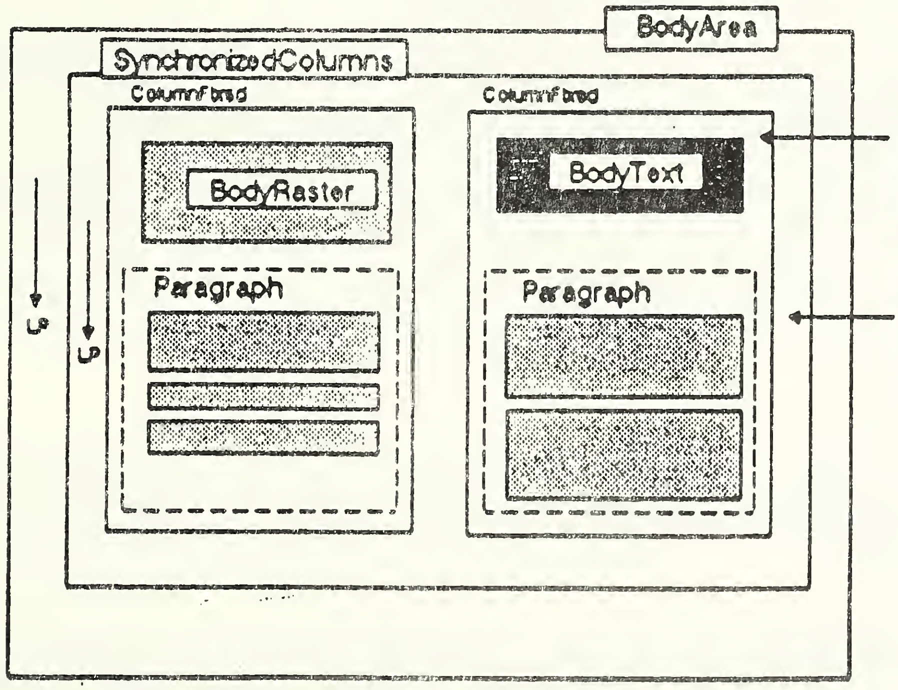
Figure 9 - Example of the layout of SnakingColumns for page layout types B, C and D

6.3.5.5 SynchronisedColumns

SynchronizedColumns is a constituent constraint that defines a composite frame that represents a synchronized columns area within a body area. A synchronized columns area is typically used to represent one or more columns of content such that the content laid out in each column belongs to different layout streams. Thus content laid out in one column is not allowed to flow into the next column.

This type of column layout is typically used when it is required to layout separate amounts of content in parallel with one another such that they are aligned. Examples are the synchronized layout of content belonging to different languages and the layout of a figure in parallel with some text. An example is shown in Figure 10.

With regard to positioning and dimensioning, SynchronizedColumns frames have the same characteristics as SnakingColumns frames.



blocks aligned along this line as specifed by the attribute "synchronization"

alignment of two constituents of the type Paragraph

Figure 10 - Example of SynchronisedColumns layout for page layout type A

The immediate subordinates of a SynchronizedColumns frame consist of any number of frames of the type ColumnFixed. An example is shown in Figure 10.

The layout path for a SynchronizedColumns frame is 270 degrees for page layout A, 0 degrees for page layout B and 180 degrees for page layouts C and D.

6.3.5.6 ColumnVariable

ColumnVariable is a constituent constraint that defines a lowest level frame that is used to represents a column of content within a SnakingColumns frame. This is a frame which is variably positioned.

The dimension of this frame in the direction parallel to the layout path of the superior SnakingColumns frame (i.e., the column width) is fixed. The dimensions of different instances of ColumnVariable frames within a given SnakingColumns frame may differ to allow columns of different widths to be specified.

The dimension in the direction orthogonal to the layout path of the superior frame (i.e., the column length) may be specified as *rule-b* or *maximum-size.*

The layout path for ColumnVariable frames is 270 degrees in the case of page layout A, 0 degrees in page layout B and 180 degrees in page layouts C and D.

All ColumnVariable frames subordinate to the same SnakingColumns frame must have the same category name; different names may be used for ColumnVariable frames laid out in different SnakingColumns frames.

6.3.5.7 ColumnFixed

ColumnFixed is a constituent constraint that defines a lowest level frame that used to represent a column of content within a SynchronizedColumns frame. This is a frame which is has a fixed position.

The dimension of this frame in the direction orthogonal to the layout path of the superior SynchronizedColumns frame (i.e., the column width) may be fixed or specified as 'maximum size' (see note~~1~~) in all page layout types. This dimension may differ for different instances of ColumnFixed frames within a given SynchronizedColumns frame to allow columns of different widths to be specified. However, the widths must be specified such that the columns do not overlap.

The dimension of this frame in the direction parallel to the layout path of the superior frame (i.e., the column length) may specified as *rule-b* or *maximum-size* in the cases of page layouts A and B. In the cases of page layouts C and D, this dimension may only be specified as *maximum-size*.

The ColumnFixed frames subordinate to a given SynchronizedColumns frame must have different category names.

The layout path for ColumnFixed frames must be equal to the layout path of the superior SynchronizedColumns frame.

The content laid out in different ColumnFixed frames within the same SynchronizedColumns frame may be specified as *synchronised* by using the attribute "Synchronization".

Note: The value *maximum-size* may only be specified for the last ColumnFixed frames laid out in a SynchronizedColumns frame to prevent overlapping of the frames.

6.3.5.8 FootnoteArea

FootnoteArea is a constituent constraint that defines a lowest level frame that is used to represent a footnote area within a body area. A footnote area is typically used for the layout of footnotes.

Frames of this type are variably positioned with a positioning fill order specified as *reverse*. Hence this frame is positioned adjacent to the leading edge of the VariableCompositeBody frame.

The dimension of FootnoteAreas frames in the direction orthogonal to the layout path of its superior frame is fixed or specified as *maximum-size*. In the direction of the layout path, the dimension is specified by *rule-b* which means that this dimension is automatically adjusted to contain all the content that is allocated to it.

The layout path for FootnoteArea frames is the same as that specified for the body area.

The only content that may be laid out in this frame is content which is associated with basic logical objects that are subordinate the composite logical object *FootnoteBody*. To achieve this, the category name *Footnote* is specified for this frame.

6.3.6 Header and footer area characteristics

The header and footer areas may consist of basic areas or composite areas.

A basic header or footer area is an area into which the content is directly laid out. This type of area is represented by a constituent constraint of the type BasicHeader or BasicFooter respectively.

A composite header or footer area is an area which is subdivided into separate sourced content and arranged content areas to provide greater versatility with regard to the layout of the content. This type of area is represented by a constituent constraint of the type CompositeHeader or CompositeFooter respectively.

In the case of basic header or footer areas, the content allocated to these areas is derived from the common part of the logical structure of a document. In the case of composite header or footer areas, the content may again be derived from the common part of the logical structure of a document but the content may also be derived from common content specified in the generic layout structure.

6.3.6.1 BasicHeader and BasicFooter

BasicHeader and Basic Footer are constituent constraints that define lowest level frames that represent areas within a page that are reserved for common content.

These types of frame have fixed positions and dimensions. The positioning of these frames within a page and layout paths that may be specified for them depends upon the page layout type use (see 6.3.4.5)

The content that is laid out in these frames is derived, using the logical source mechanism, from the content associated with the composite logical object classes of the type CommonContent.

6.3.6.2 CompositeHeader and CompositeFooter

CompositeHeader and CompositeFooter are constituent constraints that define composite frames that represent areas within a page that are reserved for common content.

These types of frame have fixed positions and dimensions. The positioning of these frames within a page and layout paths that may be specified for them depends upon the page layout type use (see 6.3.4.5)

The subordinates of these frames may consist of either:

- o any number and combination of variably positioned frames of the types SourcedContentVariable and ArrangedContentVariable, or
- o any number and combination of fixed positioned frames of the types SourcedContentFixed and ArrangedContentFixed.

The subordinate frames within a CompositeHeader or CompositeFooter frame may overlap without restriction.

6.3.6.3 SourcedContentVariable

SourcedContentVariable is a constituent constraint that defines a lowest level frame that represents a region within a header footer area that contains common content derived from the generic logical structure. This frame is variably positioned and its layout path is the same as that of the containing header or footer area.

The dimension of the edge of this frame which is orthogonal to the direction of the layout path of the superior frame is fixed or specified as *maximum-size*. The dimensions of the edge in the direction parallel to the layout path of the superior frame is specified as either fixed or *rule-b*.

This frame is required to specify the attribute "Logical source" which indicates the particular instance of the constituent constraint CommonContent which contains the content to be laid out in this frame.

Typically, this frame is used for the positioning of content which is generated during the layout process, such a character sequence containing a page number.

6.3.6.4 ArrangedContentVariable

ArrangedContentVariable is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains pre-defined common content contained in the generic layout structure. The positioning, dimensioning and layout path characteristics of this frame are the same as that for SourcedContentVariable frames.

This frame references one or more blocks of type GenericBlock (see 6.3.7) which contain the content to be laid out in this frame. Thus, this frame is typically used when it is required to layout pre-determined common content.

6.3.6.5 SourcedContentFixed

SourcedContentFixed is a constituent constraint that defines a lowest level frame that represents a region within a header footer area that contains common content derived from the generic logical structure. This frame has a fixed position and dimensions and its layout path is equal to that of the containing header or footer area.

This frame is required to specify the attribute "Logical source" which indicates the particular instance of the constituent constraint CommonContent which contains the content to be laid out in this frame.

Thus, as in the case of SourcedContentVariable frames, this frame is used for the positioning of content which is generated during the layout process, such as a character sequence containing a page number.

6.3.6.6 ArrangedContentFixed

ArrangedContentFixed is a constituent constraint that defines a lowest level frame that represents a region within a header or footer area that contains pre-defined common content derived from the generic layout structure. This frame references one or blocks of type GenericBlock (see 6.3.7) which contain the content to be laid out in this frame. Thus this frame is typically used when it is required to layout common content at pre-determined positions in the header or footer areas.

The positioning, dimensioning and layout path characteristics of this frame are the same as that for SourcedContentFixed frames.

6.3.7 GenericBlock and SpecificBlock

Two types of constituent constraints of the type *block* are defined, namely GenericBlock and SpecificBlock.

Objects of the type GenericBlock may occur in the generic layout structure as subordinates to object classes of the types ArrangedContentVariable and ArrangedContentFixed. When the layout process is performed to produce a document in formatted processable form, equivalent blocks may occur in the specific layout structure. Objects of this type are therefore restricted to occur within the header and footer areas of the page.

Objects of type SpecificBlock may only occur in the specific layout structure. They are created during the document layout process and result from the layout of basic logical objects into lowest level frames that constitute the body, header and footer areas.

6.4 Document layout characteristics

6.4.1 Flow controls

Various mechanisms are provided to control the allocation of constituent constraints representing the *body* parts of the logical structure of a document to pages sets, pages and body areas. These are described in clauses 6.4.1.1, 6.4.1.2 and 6.4.1.3. The mechanisms for controlling the layout of the *common* parts of a document are described in 6.4.1.5.

6.4.1.1 Allocation of content to page sets

Two methods of allocating the constituent constraints associated with the *body* part of the document to page sets are provided. These are a) Layout in a nominated page set and b) Starting a new page set.

The first method provides the ability to specify that part of a document is to be laid out entirely within a specified page set. This may be specified for constituent constraints of the types Passage and NumberedSegment using the attribute *Layout object class* which specifies the object identifier of the required page set.

The second method provides the ability to specify that a particular logical constituent constraint in a document and all subsequent parts of a document are to be laid out starting at the beginning of a new page set. This may be specified for constituent constraints of the types Passage, NumberedSegment, Number, FootnoteReference, BodyText, BodyRaster and BodyGeometric using the attribute *New layout object* which specifies the object identifier of the required page set.

6.4.1.2 Page breaks

This provides the ability to specify that a particular logical constituent constraint in a document and all subsequent parts of a document are to be laid out starting at the beginning of a new page. The page specified must belong to the page set in which the immediate preceding logical constituent constraint is laid out (see note~~1~~).

This may be specified for logical constituent constraints of the types Passage, NumberedSegment, Number, FootnoteReference, Bodytext, BodyRaster and BodyGeometric.

This is achieved using the attribute "New layout object". This attribute may specify the value *page* indicating that the constituent constraint is to be laid out starting on the next available page which may be of any class. Alternatively, the attribute may specify that the constituent constraint is to be laid out starting on a page of a particular class. This is achieved by specifying the object identifier of the required page class.

Note: The specification of a page breaks must not be used to layout part of a document in a new page set. If a new page set is required, then this should be explicitly specified as described in 6.4.1.2.

6.4.1.3 Allocation of content to body areas

If the page to which the content is allocated contains a basic body area, then the content is laid out in sequential order in that body area in the form of single column.

If the page contains a composite body area, then the content is allocated to single, snaking and synchronized columns areas and footnote areas as described below.

6.4.1.4 Layout of contents into column areas

When laying content into a composite body area, it is necessary to indicate the type of column area that is to be used.

Logical constituent constraints of the types NumberedSegment, Number, FootnoteReference, BodyText, BodyRaster and Bodygeometric can be specified to be laid out starting at the beginning of single columns area, snaking columns area or, synchronized columns area. When a particular type of area has been specified the document content continues to be laid out in this area until a different area is selected. This can occur at any point in the document for the above logical ~~entity~~ constraint types.

If there is insufficient area on one page to layout all the content allocated to a particular type of area, then the layout of the content will automatically continue in the same type of area in the next page. Thus content is allowed to flow freely from one page to the next when the type of layout used at the end of one page is the same as that at the beginning of the next page.

However, if a new page set or new page is explicitly specified as in clause 6.4.1.1 or 6.4.1.2 then it is necessary to explicitly specify the type of column area required.

The selection of the required area is achieved using the attribute *New layout object* which specifies the object identifier of the frame class that represents~~ing~~ the required single, snaking or synchronized columns area. In the case of single column areas, the attribute *New layout object* may indicate the category name corresponding to the frame class of the single column area that is required.

When the layout ~~is to~~ occurs in a snaking columns area, column breaks can be indicated by also using the attribute *New layout object*. ~~can be used to indicate the particular column, that is, the particular ColumnVariable frame, that is to be used for laying out the content. This control allows the content to be laid out starting in the first, second, third column etc. If no control is specified then the layout starts in the first column to be laid out in the snaking columns area. The attribute *New layout object* also provides a means of specifying a *column break*, that is this attribute can be used to indicate that subsequent content is to be laid out starting in a new column of a specified type within the snaking columns area (see note 1).~~

When the layout is to occur in a synchronized columns area, category names are used to control the particular columns that are to be used to layout the logical entities. Each column within a synchronized columns area must have a different category name and each basic logical entities to be laid out in this particular area must have a category name corresponding to a name allocated to one of the columns. The logical entities allocated to different columns may be aligned using the attribute "Synchronization".

~~Note: To achieve this type of control it necessary to specify a SnakingColumns frame which contains a sequence of different ColumnVariable frames.~~

6.4.1.4.1 Layout of footnotes

Basic logical constituent constraints that represent the content belonging to a footnote (i.e., FootnoteNumber and FootnoteText) are constrained to be laid out in a footnote area which is represented by a FootnoteArea frame (see 6.3.5.8).

This constraint is specified by means of category names. That is, the logical constituents of the types FootnoteNumber and FootnoteText and layout constituents of the type FootnoteArea are all required to have the category name *Footnote*.

More than one footnote may be placed in a footnote area within a given body area. In this case the content belonging to the footnotes are laid out sequentially in the footnote area in accordance with their reading order.

If the content belonging to a footnote cannot all be accommodated in the footnote area on one page, then the content may freely flow into the footnote area on the next page. Alternatively, it is possible to specify that a footnote is to laid out entirely within a particular footnote area. This is achieved using the attribute *Indivisibility*.

6.4.1.5 Allocation of content to header-footer areas

A header or footer area may be basic or composite (see 6.3.6). In the case of a basic area, the frame representing that area specifies the attribute *Logical source* which indicates the particular instance of the constituent constraint of the type CommonContent that is to be laid out in that area. The basic logical constituents subordinate to CommonContent are then laid out in accordance with their sequential order.

In the case of a composite header or footer area (see 6.3.6.2), the area is divided into one or more separate areas, each of which is represented by a lowest level frame. The content allocated to the separate areas may be derived from one of two sources. That is, the content may be pre-defined and represented by one or more blocks which are directly associated with the lowest level frame. Alternatively, the lowest level frame may specify the attribute *Logical source* which, as above, indicates the particular logical entity of the type CommonContent that is to be laid out in that frame.

6.4.2 Layout of the document contents

Various constraints may be specified to control the layout of the content into the body, header and footer areas. These constraints are described below.

6.4.2.1 Margins

The margins are the minimum distances, or offsets, between a part of the document content and the edge of the particular area in which that content is laid out. The margins define the maximum extents of the available area into which the content can be positioned.

Margins may be specified for any constituent constraint representing a basic logical object; different margin values may be specified for different constituent constraints without restriction.

Four margins may be independently specified for each constituent constraint, namely:

o trailing edge margin;

o leading edge margin;

o right hand edge margin;

o left hand edge margin.

These margins are defined in relationship to the layout path specified for the frame in which the content is to be laid out in (see Figure 11).

Any combination of the above margins may be specified for a particular constituent constraint. These margins are specified by the attribute *Offset.* Any value may be specified in units of BMUs. If a particular margin is not specified then it is assumed to be 0 BMUs.

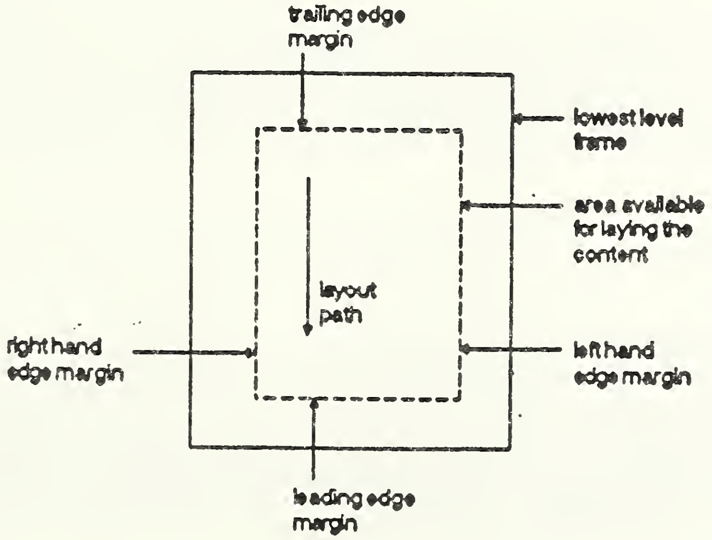


Figure 11 - Specification of margins

6.4.2.2 Separation

Separation is the minimum distance between one basic logical entity and the next when they are laid out. It may be specified for any constituent constraint representing a basic logical object. This distance is specified in BMUs by the attribute *Separation*. If no value is specified, then the minimum distance is assumed to be 0 BMUs.

6.4.2.3 Indivisibility

Indivisibility provides the means to specify whether or not a basic or composite logical entity is allowed to be split over more than one page or over more than one area within a page. It may be specified for constituent constraints of the types Passage, NumberedSegment, Number, FootnoteReference and BodyText. The attribute *Indivisibility* is used to specify this feature.

6.4.2.4 Same layout object

Same layout object provides the means to specify that the content associated with a basic logical constituent constraint and the content associated with a previous basic logical constituent constraint are to be regarded as an unbroken stream of content within a page. This may be specified for constituent constraints of the types NumberedSegment, Paragraph, Number, Footnote, FootnoteReference, BodyText, BodyRaster and BodyGeometric.

The attribute *Same layout object* is used to specify this feature. This attribute contains an expression which indicates that the previous logical constituent constraint and the constituent constraint to which the attribute applies are to be laid out starting on the same page.

6.4.2.5 Concatenation

Concatenation provides the means to specify that the content associated with a basic logical entity and the content associated with the previous basic logical entity are to be regarded as an unbroken stream of content. This may be specified for BodyText, Number, FootnoteReference, Footnotenumber, FootnoteText, CommonText and PageNumber. The attribute *Concatenation* is used to specify this feature.

6.4.2.6 Block alignment

Block alignment allows the content associated with a basic logical entity to be specified as left *aligned*, *right aligned* or *centred* within the area in which that content is laid out. Left aligned means that the content is laid out adjacent to the left hand edge margin. Right aligned means that the content is laid out adjacent to the right hand edge margin and centred means that the content is laid out midway between the left and right margins.

This feature may only be specified using the attribute *Block alignment* for constituent constraints of the types BodyText and CommonText, when they contain formatted character content, BodyRaster, and BodyGeometric, CommonRaster and CommonGeometric.

6.4.3 Layout controls applicable in the absence of a generic logical structure

In processable form documents, the generic layout structure is optional. If the generic layout structure is omitted, then it is the responsibility of the receiver to define an appropriate layout structure. No limitations are placed on the layout structure that is used.

When a generic layout structure is not specified within a processable form document, then restrictions are placed on the layout control functions described in 6.4.1 and 6.4.2 that can be specified within the document. These restrictions are indicated below.

o It is not possible to specify that certain logical parts of a document are to be allocated to a given page set or that a part of a document is to be laid out starting in a new page set, as defined in 6.4.1.1.

o It is possible to specify page breaks as defined in 6.4.1.2, but it is only possible to indicate that the layout should begin on a new page. It is not possible to specify a particular page class.

o The logical parts of the document that are intended to be laid out in the body area and in the header/footer areas of each page can be distinguished by means of application comments (see 6.6.4). An exception is that it is not possible to distinguish whether the common content is to be placed in a header or footer area or split between the two.

o It is not possible to indicate the type of layout area to be used to layout each logical constituent in the body part of a document. That is, it is not possible to indicate whether single column or multiple column areas are to be used (6.4.1.3.1). THis must be decided by the receiver.

o Footnotes within the body part of a document can be distinguished by use of the attribute *Application comments.* Footnotes are intended to be read and laid out separately from the other logical constituents of the body part (see 6.4.1.3.2). However, it is the responsibility of the receiver to decide how footnotes are laid out.

o Margins, separation, indivisibility, same layout object, concatenation and block alignment, as defined in 6.4.2, can all be specified. Only one restriction applies. Indivisibility (see 6.4.2.3) may be used to specify that a logical constituent constraint is not to be split over more than one page, but indivisibility can not be specified for other types of layout areas such as single or multiple column areas.

6.5 Content layout and imaging characteristics

A document may contain character, raster graphics and geometric graphics content.

The content architectures that may be specified using the attribute *Content architecture class* are formatted character, processable character, formatted processable character, formatted processable raster graphics and formatted processable geometric graphics. Any of these may be specified as the default in the document profile.

6.5.1 Character content

This clause defines the features that are applicable to the character content contained in a document and the presentation attributes and control functions that may be used to specify these features. These features may apply to basic logical and layout components unless otherwise indicated.

The default values for the following features may be specified in the document profile:

o graphic character sets

o graphic character subrepertoire

o code extension announcers;

o line spacing;

o character spacing;

o character path;

o line progression;

o character orientation;

o graphic rendition, including the parameters values: default rendition, bold, italicised, underlined, crossed out, primary font, 1st alternative font, 2nd alternative font, 3rd alternative font, 4th alternative font, 5th alternative font, 6th alternative font, 7th alternative font, 8th alternative font, 9th alternative font, doubly underlined, normal intensity, not underlined, not crossed out;

o tabulation;

o indentation;

o alignment;

o first line format;

o itemisation;

o widow size;

o orphan size;

o character fonts;

o kerning offset;

o proportional line spacing

o initial offset.

The specification in a document of a non-basic feature by a presentation attribute or control function must be indicated in the document profile.

6.5.1.1 Character content architecture class

Processable and formatted processable form documents may contain processable, formatted or formatted processable character content. Formatted form documents may only contain formatted character content.

When using character content, any number of content portions may be associated a basic component.

The content information in a content portion may be absent. This is to allow the representation and interchange of documents in which parts of the contents can be supplied, for example, during subsequent editing.

6.5.1.2 Character repertoires

The basic character set supported by this profile is the primary character set of ISO 8859-1. This must be designated to the G0 set and invoked to the GL.

Any other graphic character set which is registered in accordance with ISO 2375 may be designated and invoked at any point in the document provided its use is announced in the document profile as a non-basic value using the character presentation feature *Graphic character sets*. No locking shift functions are specified in this presentation feature.

The code extension techniques allowed for the designation and invocation of character sets to the left hand side and the right hand side of the 8-bit code table (GL and GR respectively) are defined in clause 6.5.1.4.

Using these code extension techniques, the graphic character sets designated and/or invoked at the beginning of a content portion containing character content are specified using the presentation attribute *graphics character sets*. The character sets may be also be changed at any point within a content portion.

The default graphic character sets which apply to the content portions within a document can be specified in the document profile using the presentation attribute *graphic character sets*.

If the character set defined in ISO 6937-2 is designated and invoked, then the use of any sub-repertoire registered according to ISO 7350 may be specified. All sub-repertoires are non-basic and their use must be indicated in the document profile.

~~Character sets may be designated and invoked at the beginning of a content associated with a basic component using the presentation attribute "Graphic character sets". If ISO 6937 is invoked then one of its subrepertoires may be indicated by the presentation attribute "Graphic character subrepertoire".~~

~~The graphic character set used may be changed at any point within the content using the code extension techniques specified in 6.5.1.3. If a character sub-repertoire of ISO 6937 is specified, then this sub-repertoire may not be changed within the content.~~

Note: The basic character repertoire supported by this profile is not the standard default value specified in ISO 8613-6; hence it may be necessary to specify, in the document profile of a particular document, that this is the default value being used for that document.

6.5.1.3 Code extension techniques

The code extension techniques specified in ISO 2022 may be used subject to the following restrictions:

o G0 set: only the primary character sets of ISO 6937-2, ISO 8859-X (where ISO 8859-X corresponds to any finalized part of ISO 8859) and a version of ISO 646 may be designated for this set; these character sets may only be invoked in GL;

o G1, G2, G3 sets: no restrictions are placed on the character sets that may be designated for these sets; these sets may only be invoked in GR;

o The locking and single shift functions allowed should be restricted to the following:

LS0 for the G0 set
LS1R for the G1 set
LS2R for the G2 set
LS3R for the G3 set
SS2
SS3;

Tutorial Note:

Here Gl and GR refer to the left and right hand parts, respectively, of the 8-bit code table.

o When specifying the presentation attribute *Graphic character sets*, it is necessary to invoke character sets for both GL and GR. Thus an allowed character set must be designated into G0, as specified above, and invoked into GR. It is also necessary to invoke a character set into GR which has been designated into G1, G2 or G3 sets.

o The empty set should be designated and invoked in GR if no other specific set is invoked into GR;

The above restrictions are illustrated in Figure 12 and 13.

The announcement and encoding of these functions are to be as specified in ISO 2022.

The code extension announcers may be used anywhere in a content portion to change the character sets invoked in GL and GR or to select single characters from the G2 or G3 sets.

~~Code extension announcers which are used in the content associated with a basic component that are not specified as default values in the document profile shall be specified in the presentation attribute "code extension announcers" for the particular basic component concerned.~~

The code extension techniques that are used or may be used in a basic component must be specified by the presentation attribute *Code extension announcers*. The default code extension announcers used throughout a document may be specified in the document profile, also using the presentation attribute *Code extension announcers.*

Tutorial Note:

In accordance with ISO 8613-6, there is no restriction concerning the number of graphic character sets which may be designated and/or invoked in the presentation attribute *Graphic character sets* providing the restrictions defined in this clause are applied. Hence, designation to a particular G set overrides a previous designation to that set. Additionally, invocation to GL or GR overrides the previous invocation to the GL or GR, respectively. Thus, the sequential order of designation and/or invocation sequences in the attribute *Graphics character sets* is significant

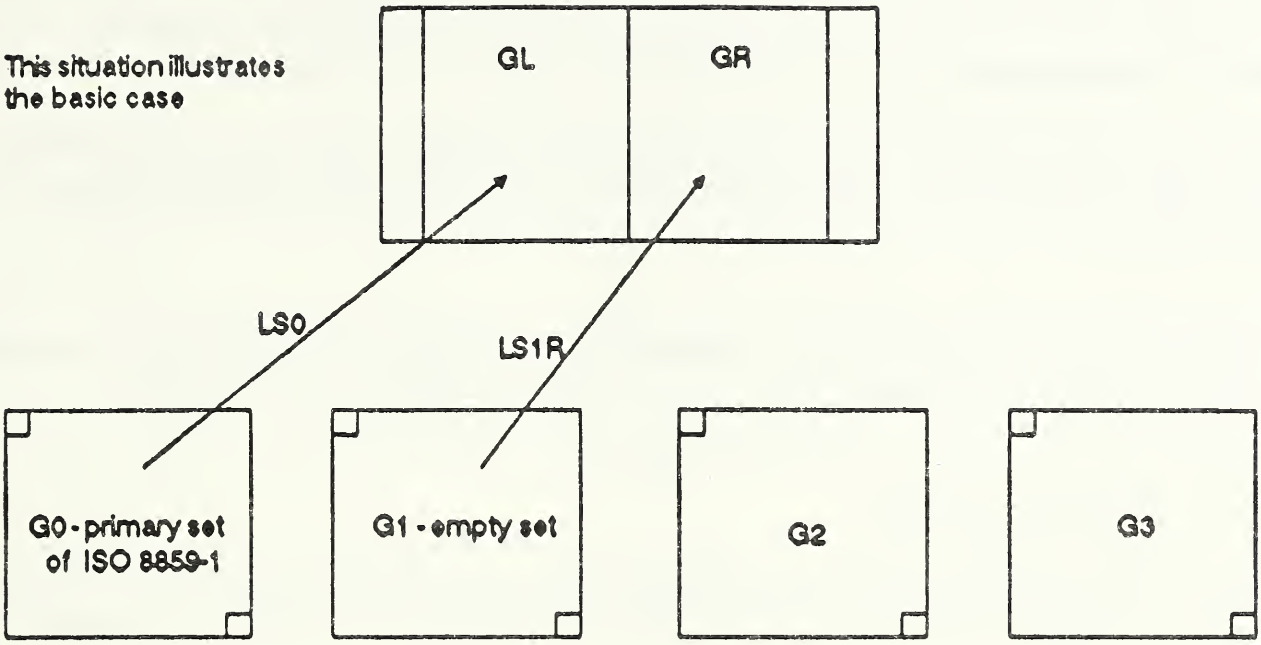


Figure 12 - Code extension features for the basic case

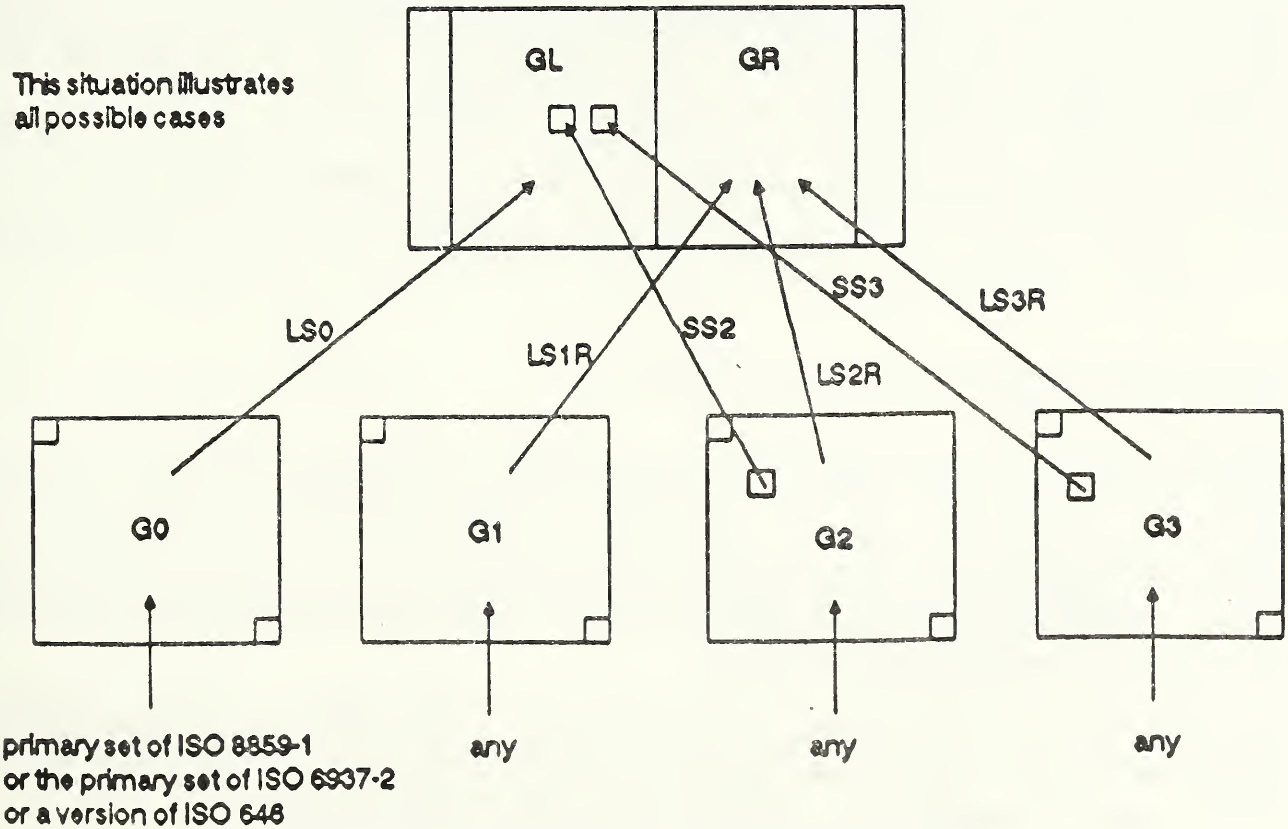

This situation illustrates
all possible cases
GL
GR
LS0
LS1R
SS2
SS3
LS2R
LS3R
G0
G1
G2
G3
primary set of ISO 8859-1
or the primary set of ISO 6937-2
or a version of ISO 646
any
any
any

Figure 13 - Code extension features for all possible cases

6.5.1.4 Line spacing

Any value of line spacing may be specified. Values of 150, 200, 300 and 400 BMUs are basic; the use of any other value in a document is non-basic and must be indicated in the document profile.

The line spacing may be specified at the beginning of the content associated with a basic component using the presentation attribute "Line spacing". The value may be changed anywhere within the content portion using the control functions SVS and SLS.

6.5.1.5 Character spacing

Any value of character spacing may be specified. Values greater than or equal to 100 are basic; the use of any other value in a document is non-basic and must be indicated in the document profile.

The character spacing may be specified at the beginning of the content associated with a basic component using the attribute "Character spacing". The value may be changed anywhere within a content portion using the control functions SHS or SCS.

Note: 1. A character spacing of 160 BMUs is provided for use with Korean Han-gul characters.

2. SHS parameters of 0, 1, 2, 3 and 4 are currently provided. The use of parameter 5 and 6 are currently being studied for use with Chinese characters.

6.5.1.6 Character path and line progression

Both horizontal and vertical writing directions may be used within a document. In the case of horizontal writing, the characters progress either from left to right or from right to left across the page and the line progression is from the top of the page to the bottom. In the case of vertical writing, the characters progress from the top of the page to the bottom and the line progression is from the right to the left. The use of these writing directions is restricted by the page layout type used.

For page layout A, only horizontal writing may be used in the body, header and footer areas. Thus, in this case the character path and line progression is specified either as 0 and 270 degrees respectively or 180 and 90 degrees respectively.

For page layout B, again only horizontal writing may be used in the body, header and footer areas. However, in this case the content in the body area is presented for viewing with the page in landscape orientation and the content in the header and footer areas is presented for viewing when the page is in the portrait orientation.

Thus for page layout B, in the body area the character path and line progression is specified either as 90 and 270 degrees respectively or 270 and 90 degrees respectively. In the header and footer areas, the character path and line progression is specified as in page layout A.

For page layout C, only vertical writing may be used in the body area and only horizontal writing may be used in the header and footer areas. Thus in the body area the character path and line progression are specified as 270 and 90 degrees respectively. In the header and footer areas, the character path and line progression is specified as in page layout A.

For page layout D, only vertical writing may be used in the body, header and footer areas. Thus in all these areas, the character path and line progression are specified as 270 and 90 degrees respectively.

A character path value of 0 degrees and a line progression value of 270 degrees are basic values. All other values are non-basic and their use in a document must be indicated in the document profile.

The values of character path and line spacing may be specified at the beginning of the content associated with a basic component using the presentation attributes *Character path* and *Line progression*, respectively. These values cannot be changed within a content portion.

6.5.1.7 Character orientation

The character orientation may be specified as 0 or 90 degrees depending on whether vertical or horizontal writing is used (see 6.5.1.6).

When vertical writing is used, characters may only be orientated at 0 degrees. When horizontal writing is used, characters may be orientated at 0 or 90 degrees.

A value of 0 degrees is basic; a value of 90 degrees is non-basic and its use in a document must be indicated in the document profile.

The value of the character orientation is specified at the beginning of the content associated with a basic component by the presentation attribute *Character orientation.* This value cannot be changed within the content.

Note: A character orientation of 90 degrees is typically used when it is required to mix ideogram characters with Latin characters when vertical writing is used. In order to achieve this, character strings orientated at 0 and 90 degrees must be defined in separate basic components.

6.5.1.8 Emphasis

The following modes of emphasising graphic characters may be distinguished:

o normal rendition;

o normal intensity;

o increased intensity (bold);

o italicised;

o not italicised;

o underlined;

o doubly underlined;

o not underlined;

o crossed-out;

o not crossed-out.

All the above modes of emphasis are basic. If no default mode is explicitly specified in the document profile, then the default mode is normal rendition (see below).

The mode of emphasis may be specified at the beginning of the content associated with a basic component using the presentation attribute *Graphic rendition.* The mode may be changed anywhere within the content using the control function SGR.

The mode of emphasis remains in effect within the content associated with a basic component until changed into a mutually exclusive mode or by the specification of *normal rendition* (see below). Mutually exclusive modes are normal/increased intensity, italicized/not italicized, underlined/not underlined and crossed out/not crossed-out. One mode from each mutually exclusive set may be in operation at any point in the document content.

Normal rendition cancels the effect of all methods of emphasis that are currently in operation and specifies that the text should be displayed in accordance with the default rendition parameters set for the presentation device. Thus, for example, if it is required to ensure that the content is not underlined, then it is necessary to explicitly specify that underlined is not to be used.

6.5.1.9 Tabulation

Tabulation stop positions may be specified at any character position along the character path. Each stop is specified by means of the following:

o The tabulation position relative to the margin position in the direction opposite to the character path;

o An optional alignment qualifier that specifies the type of alignment to be used at the designated tabulation position. The type may be specified as one of the following:

- start aligned;
- end aligned;
- centred;
- aligned around.

These alignment qualifiers are defined in ISO 8613-6. If the alignment qualifier is not explicitly specified, then it is assumed that start aligned is to be used.

Only one set of tabulation stops can be specified to be applicable to the content associated with a basic component. No limit is placed on the number of tabulation stops that can be specified within a given set.

The set of tabulation stop positions associated with the content of a basic component are specified using the presentation attribute *Line layout table.* Tabulation stop positions are invoked within the content using the control function STAB.

6.5.1.10 Indentation

Indentation is the distance between the first character on a line of content and the position of the margin position in the direction opposite to the direction of the character path. Thus the value of the indentation specified determines the line home position (as defined in ISO 8613-6).

Indentation acts as temporary alteration in the position of the offset in the direction opposite to the direction of the character path. When text is formatted, it is intended to be laid out between the indentation position and the margin position in the direction of the character path.

Any value of indentation may be specified for basic logical components using the presentation attribute *Indentation.* The indentation value may not be changed within a content portion.

6.5.1.11 Alignment

This feature is concerned with how the first and last characters on each line of character content is to be laid out during the formatting process.

The following values of alignment may be specified:

o start aligned;

o end aligned;

o centred;

o justified.

The semantics of these values are as defined in ISO 8613-6.

The presentation attribute *Alignment* is used to specify the alignment that is applicable to the content associated with a basic component. The alignment value cannot be changed within a content portion.

6.5.1.12 First line format

This feature specifies how the first line of the content associated with a basic component is to be laid out and provides for the itemisation of paragraphs.

It allows the first character in the content to be positioned at some point along the character path relative to the indentation position (as specified in 6.5.1.10). This point may be in the direction of the character path or in the direction opposite to the direction of the character path relative to the indentation position.

In addition, this feature provides for the specification of an item identifier on the first line. The item identifier is a string of characters that precedes and is separated from the remaining characters that form the first line. The control function CR (Carriage Return) is used as the separator.

The features provided correspond to examples 10.1 to 10.4 shown in Figure 10 of ISO 8613-6.

First line format is specified by the presentation attribute *First line offset* and *Itemisation*; there are no restrictions on the values that may be specified.

6.5.1.13 Widow and orphan sizes

The widow size specifies the minimum number of lines of content that must be allocated to a following frame or page when the content associated with a basic logical component is laid out such that it flows over two frames or pages. To accommodate this, it may be necessary to move a number of line of content from one frame or page to the next frame or page.

The orphan size specifies the minimum number of lines of content that must be placed in the current frame or page when the content associated with a basic logical component is split over two frames or pages. If this minimum cannot be accommodated, then the whole content must be placed in the next frame or page.

Any value of widow or orphan size may be specified using the presentation attributes *Widow size* and *Orphan size*, respectively.

6.5.1.14 Fonts

Any number of fonts may used within a document. The fonts used in a particular document are specified in the document profile using the attribute *Font list*.

Further information concerning the specification of font references in the document profile is given in Annex B.

The fonts that may be used within the content associated with each basic component are specified by the presentation attribute *Character fonts*. Up to 10 fonts taken from the list specified by the attribute *Font list* may be specified by the attribute *Character fonts*.

The font to be used at the start of the content associated with a basic component is specified using the attribute *Graphic rendition*. The fonts used within the content may be changed using the control function *SGR*.

The document profile may specify, using the attribute *Character fonts*, a default set of up to 10 fonts that are applicable to the whole document.

If the use of a particular font is explicitly specified, the character spacing used is determined from the attributes of the font. In this case constant or variable character spacing may be used, depending on the particular font specified. If no font is explicitly specified, then constant character spacing is used in accordance with clause 6.5.1.5.

6.5.1.15 Reverse character strings

Bi-directional writing is supported by this profile. Hence, a string of characters in a content portion associated with a basic component may be specified to be imaged in the reverse direction of the immediately preceding character string. Such strings can be specified by the control function *SRS* as defined in ISO 8613-6.

This control function is provided for cases in which the text belongs to different languages and the character content is written, for example, from left to right or from right to left within the same line of characters, dependent upon the language and/or character set being used.

Note: The use of this control function cannot be indicated in the document profile. Thus it is intended that implementations should ignore this control function when reverse character string layout and presentation is not supported.

~~6.5.1.16 Parallel text~~

~~A string of characters in a content portion associated with a basic component may be specificed to be imaged in parallel which another character string. Such strings can be specified by the control function PTX as defined in ISO 8613-6.~~

~~This control function is primarily provided to support certain language dependent features and should be regarded as a non-basic feature, although it is not possible to indicate the use of this control function in the document profile.~~

6.5.1.17 Kerning offset

A kerning offset value for the content associated with a basic component may be specified using the presentation attribute *Kerning offset*. It is necessary to specify such a value when certain fonts are invoked to ensure that no part of character images are positioned outside the boundary of the available area.

6.5.1.18 Proportional line spacing

The use of proportional line spacing may be invoked for the content associated with a basic logical component using the attribute *Proportional line spacing*. When this invocation occurs, the line spacing is determined from the attributes associated with the font used and may vary from one line to the next. This process is application dependent.

Proportional line spacing may only take effect when a font is explicitly specified; otherwise the value of the attribute *Proportional spacing* is ignored and the line spacing is determined according to clause 6.5.1.4.

6.5.1.19 Superscripts and subscripts

Superscripts and subscripts may be specified anywhere with in the content associated with a basic component by using the control functions *PLU* and *PLD*. The use of these control function shall be in accordance with ISO 8613-6.

6.5.1.20 Line breaks

The control functions *BPH* and *NBH* may be inserted in processable form character content to indicate where line breaks may occur or may not occur respectively, when the content is laid out.

6.5.1.21 Substitution of characters

The control function *SUB* is provided to represent characters produced by a local system that cannot be represented by a character within a character set supported by this profile.

6.5.1.22 Initial point

The initial point which is applicable to basic layout components may be specified by the attribute *Initial offset*. Any value may be specified.

6.5.1.23 Use of control functions

The following is a list of all the control functions and parameter values (where applicable) may be specified in character content:

SHS -	select character spacing (allowed parameter values: 0, 1, 2, 3, 4)
SCS -	set character spacing (allowed parameter values: any)
SVS -	select line spacing (allowed parameter values: any)
SLS -	set line spacing (allowed parameter values: any)
SGR -	set graphic rendition (allowed parameter values: 0, 1, 2, 3, 4, 9-19, 21-24, 29)
STAB -	selective tabulation (allowed parameter values: any)
SRS -	start reverse string (allowed parameters: any)
~~PTX -~~	~~parallel texts (allowed parameters: any)~~
PLD -	partial line down
PLU -	partial line up
BPH -	break permitted here
NBH -	no break here
JFY -	no justify
SUB -	substitute character
SP -	space
CR -	carriage return

LF -	line feed
SOS -	start of string
ST -	string terminator
..	code extension control functions (see 6.5.1.4)

The use of all these control functions, with the exception of SP, CR, LF, SOS and ST are described in various clauses in 6.5.

6.5.1.24 Formatting the content

All formatting of the content must be carried out by the imaging process and not by the content layout process (see ISO 8613-6). Thus the attribute *Formatting indicator* shall not be specified within documents that are conformant with this profile.

6.5.2 Raster graphics content

This clause defines the features that are applicable to the raster graphics content contained in a document. These features may apply to basic logical and layout components unless otherwise indicated.

The default values for the following features may be specified in the document profile:

o type of coding;

o compression;

o pel spacing;

o spacing ratio;

o clipping;

o image dimensions.

The specification in a document of a non-basic feature by a presentation or coding attribute or control function must be indicated in the document profile.

6.5.2.1 Raster graphics content architectures

Only the formatted processable raster graphics content architecture class may be used in documents that conform to this document application profile. This type of content may be used in processable, formatted and formatted processable form documents.

When using raster graphics content, only one content portion may be associated with an object or object class.

Also, the scalable or fixed dimension content layout process may be used when laying out and imaging the content depending upon the specification of the presentation attributes *Pel spacing* and *Image dimensions* as described in clauses 6.5.2.5 and 6.5.2.7. Both forms of content layout processes may be used in a single document.

6.5.2.2 Raster graphics encoding methods

The content may be encoded in accordance with the encoding schemes defined in CCITT Recommendations T.4 and T.6. In the case of T.4, either the one-dimensional or two dimensional encoding scheme may be used. Also the *bit-map encoding scheme* defined in ISO 8613-7 may be used. All these forms of encoding may be used in a single document and all are basic. *Uncompressed* mode of encoding may also be used but as a non-basic feature

When using the T.4 or T.6 encoding method, the relationship between the order of pels and the order of bits in the octets in the coded data stream shall be such that the first pel in the order of bits is allocated to the least significant bit of an octet. In the case of bit-map encoding, the order of encoding shall be that the first pel is allocated to the most significant bit of an octet.

In a content portion, it is required that both the coding attributes *Number of lines* and *Number of pels per line* are specified. The value of these attributes shall be a positive number; otherwise no restriction is placed on the values that may be specified. Thus this profile places no restriction of the size of the pel arrays that may be used.

The type of encoding method used is specified by the attribute *Type of coding*. The use of this attribute is non-mandatory. Thus, if this attribute is not specified for a particular content portion and if the content architecture class specified corresponds to the formatted raster graphics content architecture class, then the default encoding method is assumed to be T.6.

6.5.2.3 Pel path and line progression

The pel path and line progression supported by this profile are 0 degrees and 270 degrees respectively. This profile does not allow the specification of other values.

6.5.2.4 Clipping

A sub-region within a pel array represented by a content portion associated with a basic component may be defined using the presentation attribute *Clipping*. No restriction is placed on the use of this attribute.

6.5.2.5 Pel spacing

The pel spacing is the distance in BMUs between any two pels on a line when a pel array is imaged. Any value may be explicitly specified provided that the spacing between pels is not less than 1 BMU. The pel spacing need not be an integer value. Also, the value *null* may be specified, indicating that the scalable layout process is to be used.

The specification of the value *null* or spacings of 16, 12, 8, 6, 5, 4, 3, 2, and 1 BMU between adjacent pels are basic. The specification of any other spacing is non-basic and must be indicated in the document profile.

The pel spacing applicable to content associated with basic logical components is specified by the presentation attribute *Pel spacing*.

Note: 1. The basic pel spacing values listed above are equivalent to resolutions of 75, 100, 150, 200, 240, 300, 400, 600 and 1200 pels per 25.4mm respectively when the BMU is interpreted as 1/1200 inch.

2. The attribute *Pel spacing* specifies two integers, the ratio of which determines the pels spacing. No restriction is placed on the values of these integers except the restriction recommended in clause 178.

6.5.2.6 Spacing ratio

The spacing ratio is the ratio between the pel spacing and the line spacing when a pel array is imaged. This ratio is used to determine the line spacing from the pel spacing specified.

No restrictions are placed on the value of this ratio providing that the resultant line spacing is not less than 1 BMU. Also, the line spacing need not be an integral number of BMUs. All values are basic.

The default value may be specified in the document profile. If no default value is explicitly specified then the default value is the ratio 1:1, that is the line spacing is equal to the pel spacing.

The spacing ratio applicable to the content associated with a basic logical component is specified by the presentation attribute *Spacing ratio*.

6.5.2.7 Image dimensions

The image dimensions are the constraints to be applied to the size of the image produced when laying out a pel array represented by a content portion associated with a basic logical component.

These constraints are specified for basic logical components by the presentation attribute *Image dimensions*. The value of this attribute is only taken into account if the value of the attribute *Pel spacing* is *null*.

6.5.3 Geometric graphics content

A document may contain graphic images composed of geometric graphic controls encoded as CGM metafiles in accordance with ISO 8632. Each CGM figure must consist of a single picture only. Each GCM figure may specify its minimum dimensions.

Further information concerning the specification of geometric graphics content information is given in Annex B.

6.6 Miscellaneous features

6.6.1 Resource documents

Object classes of the types BodyText, BodyRaster and BodyGeometric, CommonText, CommonRaster, CommonGeometric and GenericBlock may refer to corresponding constituents in a resource document.

The constituents in the resource document may refer to content portions and to layout and presentation styles that are contained within the resource document. The constituents listed above are the only ones that are allowed in a resource document.

6.6.2 External documents

In the case of processable and formatted processable, either the generic logical structure or the generic layout structure or both of these structures may be contained in an external document.

6.6.3 Borders

Borders may be specified for all the frame types defined in clauses of 6.3.5 and 6.3.6 using the attribute *Borders*. All the features of borders specified in ISO 8613-2, clause 5.4.3, may be specified. The use of borders is a non-basic feature and must be indicated in the document profile. Borders cannot be specified for the constituents GenericBlock and SpecificBlock.

6.6.4 Application comments

Specification of the attribute *Application comments* is mandatory for all object classes contained in a document that conforms to this profile. Specification of this attribute is optional for objects.

This attribute is structured so that it contains two fields. The first field is mandatory when the attribute is specified and contains a numeric string which uniquely identifies the constituent for which the attribute is specified. This facilitates the processing of documents. A list of these identifiers is given in Table 2~~/T.505~~.

The second field is optional and may contain any information that is relevant to the application or user. The format of the second field is not defined in this profile and the interpretation of this field depends upon a private agreement between the originator and recipient of the document. However, this field must not contain any information that relates to the layout and presentation of the document.

The encoding of the attribute *Application comments* is defined in clause 8~~.3~~.

Table 2: Constituent constraint number string identifiers

Logical Constraint	Numeric String Identifier
DocumentLogicalRoot	0
Passage	1
NumberedSegment	2
Number	3
Paragraph	6
Footnote	8
FootnoteNumber	9
FootnoteReference	10
FootnoteBody	11
FootnoteText	12
BodyText	14
BodyRaster	17
BodyGeometric	18
CommonContent	19
CommonText	20
CommonRaster	21
CommonGeometric	22
PageNumber	~~37~~40

Table 2 - (*continued*)

Layout	Numeric String
Constraint	Identifier
DocumentLayoutRoot	0
PageSet	1
Page	2
RectoPage	3
VersoPage	4
CompositeHeader	5
VariableCompositeBody	7
ColumnFixed	8
ColumnVariable	9
SnakingColumns	10
SynchronizedColumns	11
BasicFloat	12
FootnoteArea	15
ArrangedContentFixed	16
ArrangedContentVariable	17
SourcedContentFixed	18
SourcedContentVariable	19
BasicHeader	27
BasicBody	28
GenericBlock	29
SpecificBlock	30
CompositeFooter	32
BasicFooter	~~31~~33

Note: The value of each numeric string identifier is unique for constituents within either the logical or layout structures. Also the number string identifiers are unique within the series of hierarchically related profiles to which this profile belongs.

6.6.5 Alternative representation

The content information in a content portion may be replaced by a string of characters specified in the attribute *Alternative representation*. This attribute may be specified in content portions that contain character, raster graphics or geometric graphics content.

The specification and use of this attribute is optional. The string of characters specified must belong to one of the character repertoires indicated in the document profile attribute *Alternative representation character sets*. If the latter attribute is not explicitly specified in the document profile, then the default character set is the minimum subrepertoire of ISO 6937-2. The control functions CR and LF may also be used within the character string but no other control function is allowed.

6.6.6 Page numbering

As described in section 6.2.3.3, the constituent constraint PageNumber may contain a content generator which may refer to a page number. This content generator is evaluated when the document is laid out and this mechanism provides a means of reproducing the appropriate number of each page of a document.

The content generator has the following format:

<string-literal><num-expr><string-literal>

The format of this content generator is defined in the macro HEADERFOOTERSTRING (see note ~~1~~).

The <string-literal> fields are optional and are pre-defined character strings. The basic character repertoire used to specify these strings is the primary character repertoire of ISO 8859-1. Any other character repertoire, and subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer and indicated in the document profile as a non-basic value. No other control functions may be used in these strings.

The field <num-expr> is a reference to a binding PGnum which specifies the number of the page concerned. This binding is initialised at the document layout root or page set level (see the macro INITIALISEPGNUM in 7.2.1) and automatically incremented on each successive page (see macro PAGENUMBER in 7.2.1).

The content associated with logical object classes of the type PageNumber is laid out in a frame of one of the following types: BasicHeader, BasicFooter, SourcedContentVariable, SourcedContentFixed (see 6.3.6) using the logical source mechanism. Thus when the appropriate frame is being laid out, the field <num-expr> in the content generator contained in a logical object class of the type PageNumber is evaluated and this determines the value of the binding PGnum that is associated with the current page being laid out.

The number associated with the binding PGnum is applied to a string function during its evaluation in order to convert the number into a character string. This enables the number to be represented in the form of an Arabic numeric string, an upper or lower case Roman numeric string or an upper or lower case alphabetic string.

Each page class can refer to a different instance of logical object classes of the type PageNumber and this allows different page numbering formats to be used for different parts of the document.

An example of page numbering is *Page X* which consists of two concatenated character strings. The first is the literal character string *Page* and this is concatenated to a string function denoted by *X*. When *X* is evaluated in a particular instance it may, for example, return the character string *iv*, the Roman numeral (lower case) for the number *4*.

Note: Unless otherwise stated, the macros referred to in this clause are defined in clause 7.3.1.

6.6.7 Segment numbering

As described in section 6.2.2.4, the constituent Number contains a content generator which when evaluated during the layout process produces an identifier which serves to identify the Numbered Segment to which the constituent Number belongs.

The format of this identifier is as follows:

<pre-str><num-str><suf-str>

This format is defined in the macro SEGMENTNUMBER (see note~~1~~).

The fields <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length. The basic character repertoire used to specify these strings is the primary character repertoire of ISO 8859-1. Any other character repertoire, and subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer and indicated in the document profile as a non-basic value. No other control functions may be used in these strings.

The field <num-str> is the segment identifier which consists of a single number or a sequence of two or more numbers, each of which is separated by a *separator.* The separator is a character string and may, for example, consist of a full stop or space. An example of a segment identifier is *6.3.4.2.1.* Thus segment identifiers have the general form:

<number>[<separator><number>]...

where [..]... indicates optional repetition.

In a document, the prefix and suffix and separator character strings are string literals or carried by the bindings *prefix-<n>* and *suffix-<n>* respectively. The separator character strings are carried by bindings of the form *<separator-<n>* and the segment identifier <num-str> is carried by the binding *numberstring-<n>.* In all these bindings *<n>* is a sequence of one or more digits and the document may contain any number of different bindings of these types. For example, *prefix-1* and *suffix-2* may be used to carry the prefix and suffix strings used in the first and second numbered segments.

These bindings can be initialised at the document logical root, passage or at any numbered segment level to start the numbering scheme sequence at a subordinate level of numbered segment. They can also be re-specified at any level within the numbering scheme. The initialisation of bindings is specified by the macro INITIALISEANY.

In order to evaluate the value of *numberstring-<n>* for each numbered segment, a number is assigned to each numbered segment at a given level. If the numbered segments are all of the same class then this number can be determined by the ORDINAL numeric function. If they are of different classes, then the number is carried by a binding of the form *number-<n>.*

A different binding of the type *number-<n>* is used for each numbered segment level and is initialised at a higher level constituent than the one in which it is used. The number associated with each numbered segment level is automatically incremented for each successive numbered segment (see the macro USENUMBERS).

The binding *numberstring-<n>* that is applicable to a given level of numbered segment is now constructed as follows:

<numberstring-x><separator-y><number-z>

Hence, the segment identifier consists of a concatenation of up to three fields. The field *<numberstring-x>* is a reference to the segment identifier applicable to the immediately superior level of numbered segment (if any). This identifier is in the form of a character string. The field *<separator-y>* is optional and is a reference to a separator defined at some higher level in the document structure.

The field *<number-z>* is the number applicable to the given numbered segment whose identifier is being constructed. As indicated above, this number can be determined from an ORDINAL expression or by reference a binding of the form *number-<n>* which is specified for the same numbered segment whose identifier is being constructed. In either case, a string function is applied to the number to convert it into a character string. This string function allows the number to be represented in one of the following forms: Arabic number string, upper or lower case Roman numeral string, or upper or lower case alphabetic characters. This construction is defined in the macro USENUMBERSTRINGS.

The constructed binding of the form *numberstring-<n>* is then available for constructing the identifiers at lower levels of numbered segment. This binding is also referred to in a content generator carried by the constituent Number, which causes the identifier (with optional prefix and suffix strings) to be generator and reproduced when the document is laid out.

~~All numbers at a particular level within a passage must have the same format, that is, character string representation and structure.~~

Numbers at a particular level within a Passage can vary in format and character string representation and structure. This feature can be used to facilitate a varied numbering scheme for different NumberedSegment constituent constraints that are utilized to represent numbered sections, tables, figures and other numbered document structures.

Note: The macros referred to in this clause are defined in clause 7.3.1.

6.6.8 Footnote numbering

A footnote number is a character string that identifies a given footnote. The format of this string is as follows:

<string-literal><num-str><string-literal>

This format is defined in the macro *FNOTENUMBER*.

The *<string-literal>* fields are optional and are predefined prefix and suffix character strings. The basic character repertoire used to specify these strings is the primary character repertoire of ISO 8859-1. Any other character repertoire, and subrepertoire if appropriate, may be used provided that it is designated and invoked by the appropriate code extension announcer and indicated in the document profile as a non-basic value. No other control functions may be used in these strings.

The field *<num-str>* is an automatically generated numeral or a user supplied character string that generally serves to identify a particular footnote. Numerals may be represented in the form of Arabic numerals, upper or lower case Roman numerals or upper or lower case alphabetic characters. Automatically generated footnote numbers are incremented sequentially from an initial value which may be set to any positive value at the beginning of the document and reset at any Passage.

~~A footnote number is a label that identifies a given footnote. The label may be automatically generated or supplied by the user. This label may be represented in the form of Arabic numerals, upper or lower case Roman numerals or upper or lower case Alphabetic characters. Automatically generated footnote numbers are incremented sequentially from an initial value which may be set to any positive value at the beginning of the document and reset at any passage. Also the label may have a pre-defined prefix or suffix string.~~

A single binding *fnotenumber* is provide to represent footnote numbers. This may be initialised to any non-negative number at the logical root or on any Passage (see specification of the macro INITIALISEFNOTE).

The footnote number is incremented using a binding expression at each footnote object (see the macro INCFNOTENUMBER). This is then made into a character string using a string function. This value is assigned to the binding *fnotestring* (see the macro FNOTENUMBERSTRING).

Alternatively, a character string literal may be assigned to the binding *fnotestring*; this provides the user with the ability to supply particular footnote labels for individual footnotes (see the macro FNOTESTRINGLITERAL).

The constituents FootnoteReference and FootnoteNumber contain content generators which reference the binding *fnotestring*. The content generator may include pre-defined character strings which may be specified as prefixes and suffixes to the content of the binding *fnotestring*. For example, prefixes and suffixes may be used to specify the control functions required to indicate that the footnote label is to form a superscript. The format of the content generator is defined using the macro FNOTENUMBER.

6.6.9 User readable comments

Information which is to be interpreted as comments relevant to constituents and associated content portions may be specified using the attribute *User readable comments.* This information is intended for presentation to humans.

The information consists of a string of characters which must belong to one of the character repertoires indicated in the document profile attribute *Comments character sets* (see 6.7.4.1). If the latter attribute is not explicitly specified, then the default character set is the minimum repertoire of ISO 6937-2. The control functions *CR* and *LF* and code extension control functions may also be used within the character string. However, no other control functions are allowed.

6.6.10 User visible name

Information which may be used to idenlify constituents within a document may be specified using the attribute *User visible name.* This information is intended for presentation to humans, for example, to assist in the editing of documents.

The information consists of a string of characters which must belong to one of the character repertoires indicated in the document profile attribute *Comments character sets* (see 6.7.4.1). If the latter attribute is not explicitly specified, then the default character set is the minimum repertoire of ISO 6937-2. The control functions *CR* and *LF* and code extension control functions may also be used within the character string. However, no other control functions are allowed.

6.7 Document management features

Information relating to the document as a whole is specified in the document profile which is represented by the constituent DocumentProfile. This constituent must be specified in every document.

The information in the document profile is classified into the following categories:

o document constituent information;

o document identification information;

o document default information;

o non-basic characteristics information;

o document management information.

The information in the document profile may be of interest to the user or may be used for machine processing of the document.

6.7.1 Document constituent information

This information specifies which constituents are used to represent the document, including constituents that are external to the interchanged document. This information is divided into three categories.

6.7.1.1 Presence of document constituents

This information indicates which constituents are included in the document. That is, this information indicates whether or not the document contains a generic logical structure, a specific logical structure, a generic layout structure, a specific layout structure, layout styles and presentation styles (see note ~~1~~). It is mandatory to specify this information in the document profile.

Note: If the generic logical or layout structure is external to the document (see 6.7.1.3), then it is still necessary to indicate that these structures are present and form part of the document.

6.7.1.2 Resource document information

This information consists of a reference to a resource document (see 6.6.1). This is specified by the attribute *Resource document*. If constituents in the document contain references to object classes in a resource document, then it is mandatory to specify this information in the document profile.

6.7.1.3 External document information

This information consists of a reference to an external document which may consist of a generic logical structure or generic layout structure or both of these structures (see 6.6.2). If such a reference is required, then it is specified by the attribute *External document class* in the document profile.

6.7.2 Document identification information

This information relates to the identification of the document. This information is divided into six categories.

6.7.2.1 Document application profile information

This information indicates the document application profile to which the document belongs. It is mandatory to specify this information using the attribute *Document application profile.* Two document application profiles are supported by this specification. Each profile has its own unique identifier. One is provided for the profile defining an ASN.1 based encoding of ODIF and a second is provided for the profile defining a SGML/ODL/SDIF based encoding of ODIF (see 8).

6.7.2.2 Document architecture class information

This information indicates the document architecture class to which the document belongs (see 6.1). It is mandatory to specify this information using the attribute *Document architecture class.*

6.7.2.3 Content architecture class information

This information indicates the content architecture classes used in the document (see 6.5.1.1, 6.5.2.2 and 6.5.3.3). It is mandatory to specify this information using the attribute *Content architecture class.*

6.7.2.4 Interchange format class information

This information indicates the interchange format class used to represent the document (see 8). It is mandatory to specify this information using the attribute *Interchange format class* for data streams conforming to Interchange Format Class A. This attribute has no meaning for data streams conforming to Interchange Format SDIF.

6.7.2.5 ODA version information

This information indicates the ISO standard or CCITT Recommendation to which the document conforms. It also specifies a calender date, which indicates that the document conforms to the version of the ISO standard or CCITT Recommendation and any addenda that are current on that date. It is mandatory to specify this information using the attribute *ODA version*.

6.7.2.6 Document reference

This information serves to identify the document. Typically this information is allocated to the document by the creator of the document. The identifier may consist of an ASN.1 object identifier or a string of characters. It is mandatory to specify this information using the attribute *Document reference*.

6.7.3 Document default information

This information specifies various default values for attributes used in the document. The default values that are allowed are specified in the various clauses of clause clause 6 of this profile. The specification of this information is only required when it is required to specify a default value which is other than the standard default value specified in ISO 8613.

Default values for the following groups of attributes can be specified:

o document architecture attributes;

o character content attributes;

o raster graphics attributes;

o geometric graphics attributes.

6.7.4 Non-basic characteristics information

This information specifies the non-basic attribute values specified in the document. It is mandatory to specify a non-basic attribute in the document profile when such a value is used in the document.

The following types of non-basic attributes can be specified:

o document profile character sets;

o comment character sets;

o alternative representation character sets;

o page dimensions;

o medium-type;

o layout path;

o borders;

o character presentation features;

o raster graphics presentation features;

o raster graphics coding attributes.

6.7.4.1 Profile character sets

Some document profile attributes have values consisting of character strings, for example, the document management attributes. The graphic character sets assumed to be designed and invoked at the beginning of these character strings is specified by the document profile attribute *Profile character sets.*

The graphic character sets that are designated and invoked by the attribute *Profile character sets* are subject to the following restrictions:

o GO set: only the primary character sets of ISO 6937-2 and ISO 8859-X (where ISO 8859-X corresponds to any finalized part of ISO 8859) and a version of ISO 646 may be designated for this set. These graphic character sets may only be invoked in GL;

o G1, G2 and G3 sets: no restrictions are placed on the graphics character sets that may be designated for these sets. These graphic character sets may only be invoked in GR;

o The empty set must be designated into G1 and invoked into GR if no other specific set is invoked in GR.

If the attribute *Profile character sets* is not specified, then the character set designated and invoked is assumed to be the minimum subrepertoire of ISO 6937-2.

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of T.61 is designated and invoked in this attribute.

6.7.4.2 Comments character sets

The character sets assumed to have been designated and invoked at the beginning of the character strings specified by the attributes *User readable comments* (see 6.6.8) and *User visible name* (see 6.6.9) are specified using the document profile attribute *Comment character sets.*

It also specifies code extension techniques and the graphics character sets which may be used in the attributes *User readable comments* and *User visible name.*

If this attribute is specified, the code extension techniques which may be used in the attributes *User readable comments* and *User visible name* should be announced by appropriate code extension announcers. The use of *G0* and *LS0* should always be announced by appropriate code extension announcers. The use of *G0* set and *LS0* should always be announced. Other code extension announcers are to be specified according to the requirements of a particular document.

The restrictions on the use of code extension techniques as defined in 6.5.1.4 also apply.

All the graphic character sets which may be used in the attribute *User readable comments* and *User visible name* should be designated in the attribute *Comments character sets.*

There are no restrictions concerning the numbers of graphic character sets which are designated and/or invoked in the attribute *Comments character sets.* Hence, designation to the same G set overrides the previous G set and invocation to the same GL or GR overrides the previous GL or GR.

If the attribute *Comments character sets* is not specified, then the character set designated and invoked is assumed to be the minimum subrepertoire of ISO 6937-2.

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of T.61 is designated and invoked in this attribute.

6.7.4.3 Alternative representation character sets

This attribute specifies the graphic character sets designated and invoked at the beginning of the attribute *Alternative representation*, other than the standard default graphic character sets.

The restriction on graphic character sets described in 6.7.4.1 is also applied to this attribute. If this attribute is not explicitly specified in the document profile, the minimum subrepertoire of ISO 6937-2 is used in the attribute *Alternative representation*.

When the Teletex subrepertoire of ISO 6937-2 is needed, the primary set and the supplementary set of T.61 is designated and invoked in this attribute.

6.7.5 Fonts list

This information specifies all the fonts (if any) used in the document. It is specified using the attribute *Font list*.

6.7.6 Document management attributes

This information contains information about the content of the document and its purpose. Information relating to the following may be specified:

o document description (see note +);

o dates and times;

o originators;

o other user information;

o external references;

o local file references;

o content attributes;

o security information.

The attributes that may be used to specify this information are defined in ISO 8613-4.

The string of characters used in the document management attributes must belong to the character set indicated in the document profile attribute *Profile character sets* (see 6.7.3.1). If this attribute is not explicitly specified in the document profile, then the default character set is the minimum subrepertoire of ISO 6937-2.

The control functions SP, CR and LF may also be used within the character strings, but no other control functions are allowed. Hence, the graphic character set cannot be changed in the document management attributes.

Note: The document description includes the specification of the document reference (see 6.7.2.6).

7 Specification of constituent constraints

The structure diagrams illustrating the relationships between the constituents in the logical structures are shown in Figures 14, 15 and 16. The macros indicated on these diagrams are defined in 7.3.1. These macros define the permissible values for the attribute *Generator for subordinates* that are applicable to the constituents and, in effect, define the allowed structures that are supported by this profile.

The structure diagrams illustrating the layout structures are shown in Figures 17, 18 and 19. The macros indicated in these diagrams are defined in 7.4.1.

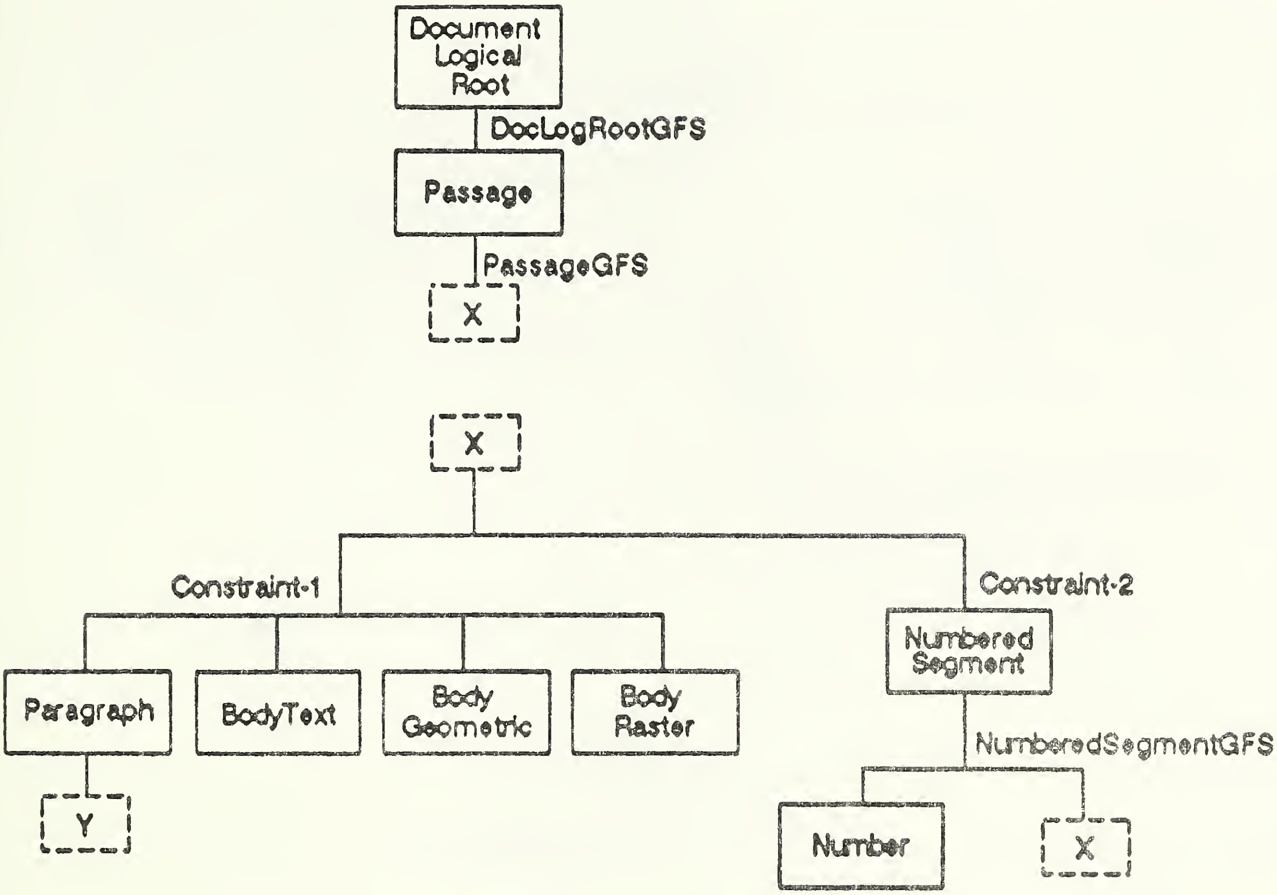


Figure 14 - Logical structure for the *body* part - Passage and NumberedSegments

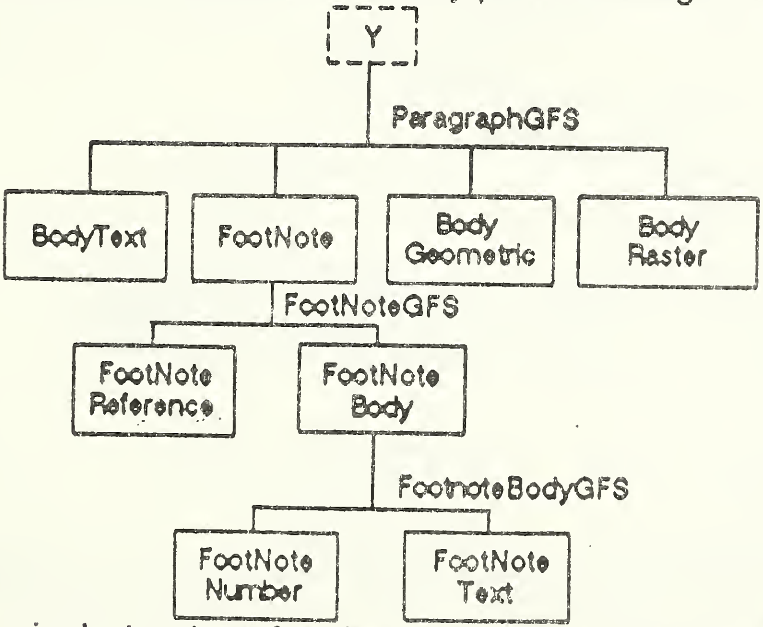


Figure 15 - Logical structure for the *body* part - Paragraph

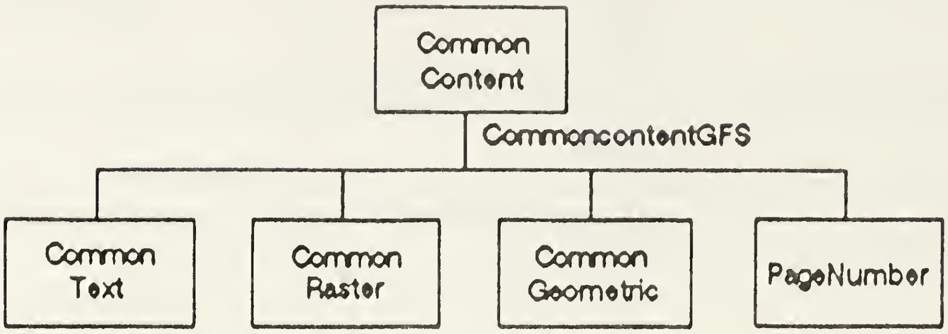


Figure 16 - Logical structure for the *common* part

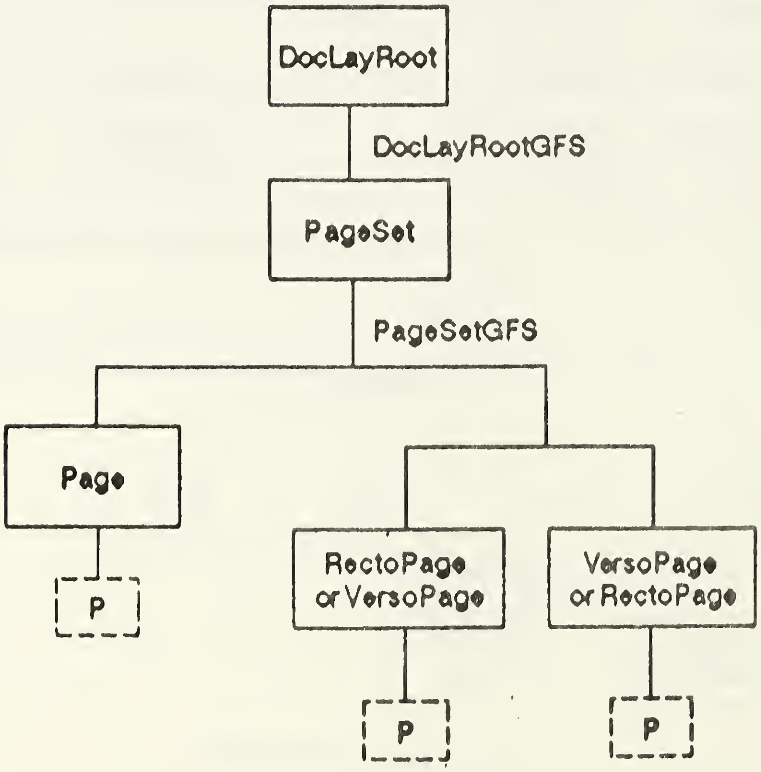


Figure 17 - Layout structure for DocumentLayoutRoot and PageSet

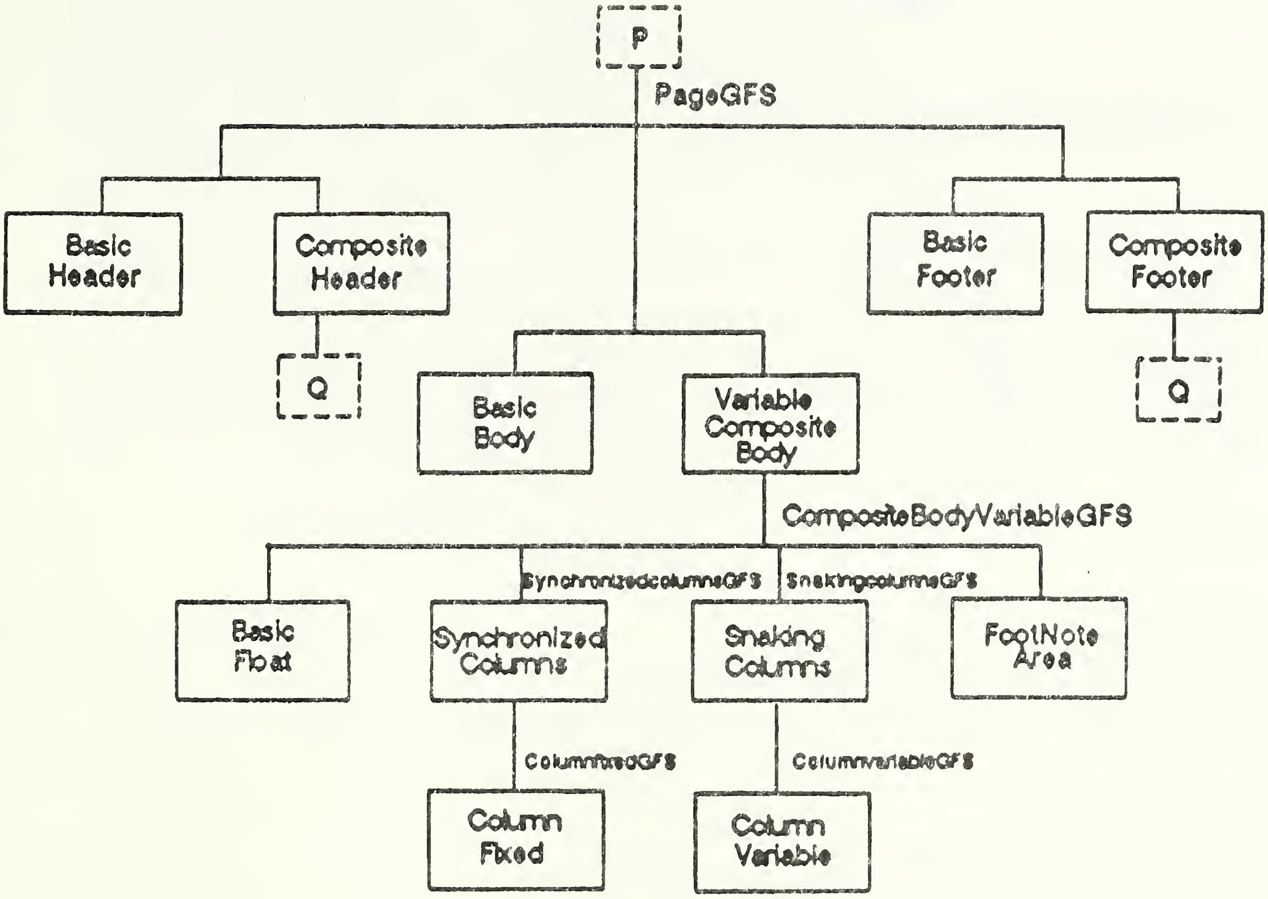


Figure 18 - Layout structure for Page, RectoPage and VersoPage

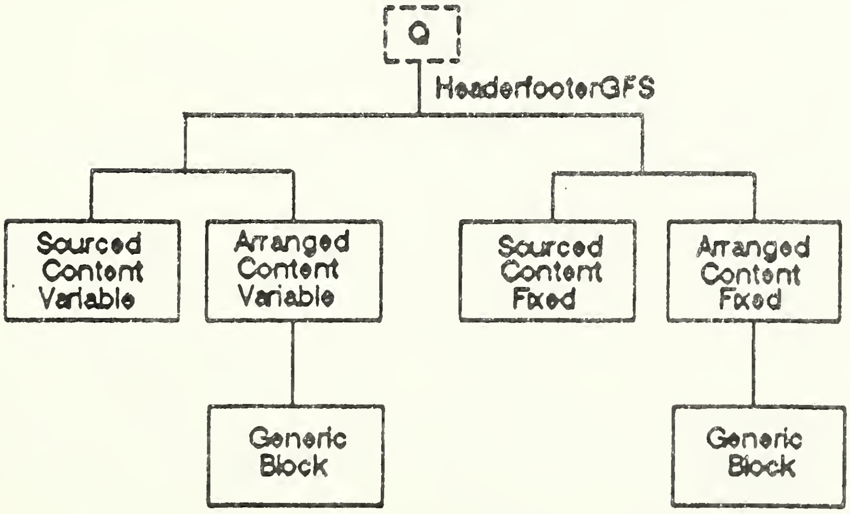


Figure 19 - Layout structure for header and footer frames

7.1 Document profile constraints

7.1.1 Macro definitions

```
DEFINE(FC,  "ASN.1{2 8 2 6 0}" -- formatted character content --)
DEFINE(PC,  "ASN.1{2 8 2 6 1}" -- processable character content --)
DEFINE(FPC, "ASN.1{2 8 2 6 2}" -- formatted processable character content)
DEFINE(FPR, "ASN.1{2 8 2 7 2}" -- formatted processable raster
                                  graphics content --)
DEFINE(FPG, "ASN.1{2 8 2 8 0}" -- formatted processable geometric
                                  graphics content --)
DEFINE(FDA, "{'formatted'}")
DEFINE(PDA, "{'processable'}")
DEFINE(FPDA, "{'formatted-processable'}")
DEFINE(DAC, "DocumentProfile (Document-architecture-class)")


DEFINE(NominalPageSizes, "
        {REQ #horizontal-dimension {6922},
         REQ #vertical-dimension {9920}     -- ISO A5 portrait}
        |{REQ #horizontal-dimension {9920},
          REQ #vertical-dimension {6922}    -- ISO A5 landscape}
        |{REQ #horizontal-dimension {9920},
          REQ #vertical-dimension {14030}   -- ISO A4 portrait}
        |{REQ #horizontal-dimension {14030},
          REQ #vertical-dimension {9920}    -- ISO A4 landscape}
        |{REQ #horizontal-dimension {14031},
          REQ #vertical-dimension {19843}   -- ISO A3 portrait}
        |{REQ #horizontal-dimension {19843},
          REQ #vertical-dimension {14031}   -- ISO A3 landscape}
        |{REQ #horizontal-dimension {10200},
          REQ #vertical-dimension {16800}   -- ANSI legal portrait}
        |{REQ #horizontal-dimension {16800},
          REQ #vertical-dimension {10200}   -- ANSI legal landscape}
        |{REQ #horizontal-dimension {10200},
          REQ #vertical-dimension {13200}    -- ANSI A portrait}
        |{REQ #horizontal-dimension {13200},
      REQ #vertical-dimension {10200}   -- ANSI A landscape}
     |{REQ #horizontal-dimension {13200},
      REQ #vertical-dimension {20400}   -- ANSI B portrait}
     |{REQ #horizontal-dimension {20400},
      REQ #vertical-dimension {13200}   -- ANSI B landscape}
     |{REQ #horizontal-dimension {12141},
      REQ #vertical-dimension {17196}   -- Japanese legal portrait}
     |{REQ #horizontal-dimension {17196},
      REQ #vertical-dimension {12141}   -- Japanese legal landscape}
     |{REQ #horizontal-dimension {8598},
      REQ #vertical-dimension {12141}   -- Japanese letter portrait}
     |{REQ #horizontal-dimension {12141},
      REQ #vertical-dimension {8598}    -- Japanese letter landscape}
           ")


DEFINE(GRAPHICRENDITIONS "
          {'cancel'|'increased-intensity'
           |'italicised'|'underlined'|'crossed-out'
           |'primary-font'|'first-alternative-font'
           |'second-alternative-font'|'third-alternative-font'
           |'fourth-alternative-font'|'fifth-alternative-font'
           |'sixth-alternative-font'|'seventh-alternative-font'
           |'eighth-alternative-font'|'ninth-alternative-font'
           |'doubly-underlined'|'normal-intensity'
           |'not-italicised'|'not-underlined'|'not-crossed-out'}...
               ")
```

-- Macro defing permissible code extension announcers. This macro may be used in each constituent constraint or presentation style constraint. Note that all the values are basic. --

```
DEFINE(CDEXTEN, "  ESC 02/00 05/00,      -- LS0 --
          [ESC 02/00 05/03],     -- LSR1 --
          [ESC 02/00 05/05],     -- LSR2 --
          [ESC 02/00 05/07],     -- LSR3 --
          [ESC 02/00 05/10],     -- SS2 --
          [ESC 02/00 05/11]      -- SS3 --
                ")
```

```
-- Macro defining code extension announcers for profile default values --

DEFINE(DAP-DEFAULT-CDEXTAN, "$CDEXTAN")


-- Macros defining final character for designation --

DEFINE(FCORE,  "04/02 -- the 94 characters of the IRV of ISO 646
                (revised 1990) (i.e ASCII) --")

DEFINE(F646,   "-- a final character designating any version of ISO 646
            except 04/02 --")

DEFINE(F94S,   "-- a final character designating any registered 94 single
            byte graphic character set --")

DEFINE(F94M,   "-- a final character designating any registered 94 multi
            byte graphic character set --")

DEFINE(F96S,   "-- a final character designating any registered 96 single
            byte graphic character set --")

DEFINE(F96M,   "-- a final character designating any registered 96 multi
            byte graphic character set --")

DEFINE(FEMPTY, "07/14  -- the empty set --")


-- Macros defining designation sequences --

DEFINE(DEG-CORE-GO,  "ESC 02/08 $FCORE")
             -- Designate the 94 characters of the IRV of
               ISO 646 to G0 --

DEFINE(DEG-646-GO,   "ESC 02/08 $F646")
             -- Designate any version of ISO 646, except 04/02,
               to GO --

DEFINE(DEG-ANY-G1,   "{ESC 02/09 $F94S
              |ESC 02/04 02/09 $F94M
              |ESC 02/13 $F96S
```

```
                |ESC 02/04 02/13 $F96M}")
              -- Designate any character set to G1 --

DEFINE(DEG-ANY-G2,   "{ESC 02/10 $F94S
                |ESC 02/04 02/10 $F94M
                |ESC 02/14 $F96S
                |ESC 02/04 02/14 $F96M}")
              -- Designate any character set to G2 --

DEFINE(DEG-ANY-G3,   "{ESC 02/11 $F94S
                |ESC 02/04 02/11 $F94M
                |ESC 02/15 $F96S
                |ESC 02/04 02/15 $F96M}")
              -- Designate any character set to G3 --

DEFINE(DEG-EMPTY-G1, "ESC 02/09 $FEMPTY")
              -- Designate the empty set to G1 --


-- Macros defining shift functions --

DEFINE(LSO,    "00/15")       -- locking shift invoking G0 to GL --

DEFINE(LS1R,   "ESC 07/14")   -- locking shift invoking G1 to GR --

DEFINE(LS2R,   "ESC 07/13")   -- locking shift invoking G2 to GR --

DEFINE(LS3R,   "ESC 07/14")   -- locking shift invoking G3 to GR --

DEFINE(SS2,    "08/14")       -- single shift invoking G2 to GL --

DEFINE(SS3,    "08/15")       -- single shift invoking G3 to GL --


  -- Macro defining permissible graphic character sets. --

DEFINE(PERMIT-GRCHAR, " {$DEG-CORE-GO $LS0
                  |$DEG-646-G0 $LS0},
                  {$DEG-ANY-G1 $LS1R
                   |$DEG-ANY-G2 $LS2R
                   |$DEG-ANY-G3 $LS3R}...
```

```
                    |{$DEG-EMPTY-G1 $LS1R}  ")


 -- Macro defining default graphic character sets --

DEFINE(DAP-DEFAULT-GRCHAR, "$PERMIT-GRCHAR")


 -- Macro defining basic character sets. Note that this macro is defined
    for clarification of the specification and is not to be used in any
    other part of this DAP specification. --

DEFINE(BASIC-GRCHAR, "  $DEG-CORE-G0 $LSO,
                  $DEG-EMPTY-G1 $LS1R  ")


 -- Macro defining non-basic character sets --

DEFINE(NON-BASIC-GRCHAR,  "  {$DEG-646-G0
                      |$DEG-ANY-G1
                      |$DEG-ANY-G2
                      |$DEG-ANY-G3}... ")

 -- Macro defining character sets used in document profile attributes --

DEFINE(PROFCHAR, "  {$DEG-CORE-G0 $LS0,
                |$DEG-646-G0 $LS0},
               {$DEG-ANY-G1 $LS1R
                |$DEG-ANY-G2 $LS2R
                |$DEG-ANY-G3 $LS3R
                |$DEG-EMPTY-G1 $LS1R}  ")


 -- Macro defining comments character sets --

DEFINE(COMCHAR, " {ESC 02/00 05/00,         -- LS0 --
               [ESC 02/00 05/03],       -- LSR1 --
               [ESC 02/00 05/05],       -- LSR2 --
               [ESC 02/00 05/07],       -- LSR3 --
               [ESC 02/00 05/10],       -- SS2 --
               [ESC 02/00 05/11]},      -- SS3 --
```

```
              {$DEG-CORE-G0 [LS0]
               |$DEG-646-G0 [LS0]},
              {{$DEG-ANY-G1 [$LS1R]
               |$DEG-ANY-G2 [$LS2R]
               |$DEG-ANY-G3 [$LS3R]}...
               |$DEG-EMPTY-G1 $LS1R}}  ")
```

```
 -- Macro defining character sets used for alternative representation --

DEFINE(ALTCHAR, "$PROFCHAR")
```

7.1.2 Constituent constraints

7.1.2.1 DocumentProfile

```
{

   CASE $DAC OF {

      $FDA:  PERM  Generic-layout-structure    {'factor-generator-set'},
             REQ   Specific-layout-structure   {'present'},
             PERM  Presentation-styles         {'present'}

      $PDA:  PERM  Generic-layout-structure    {'complete-generator-set'},
             REQ   Generic-logical-structure   {'complete-generator-set'},
             REQ   Specific-logical-structure  {'present'},
             PERM  Presentation-styles         {'present'},
             PERM  Layout-styles               {'present'}

      $FPDA: REQ   Generic-layout-structure    {'complete-generator-set'},
             REQ   Specific-layout-structure   {'present'},
             REQ   Generic-logical-structure   {'complete-generator-set'},
             REQ   Specific-logical-structure  {'present'},
             PERM  Presentation-styles         {'present'},
             PERM  Layout-styles               {'present'}

             }
   PERM  External-document-class   {ANY_VALUE},
PERM  Resource-document          {ANY_VALUE},

PERM  Resources          {MUL{REQ #resource-identifier {ANY_VALUE},
                     REQ #resource-object-class
                             -identifier {ANY_VALUE}},

 --  document characteristics --

REQ   Document-application-profile  {~~--- to be supplied ---~~ -- See clause 8 for a
                                    definition of the permitted values for this attribute
                                    ---},
```

Editor's Note: The value for this attribute is dependent on the encoding being supported.

```
REQ   Document-application-profile-defaults {

   -- document architecture defaults --

  CASE $DAC OF {
     $FDA   PERM  #content-architecture-class  {$FC|$FPC},
     $PDA   REQ   #content-architecture-class  {$FC|$PC|$FPC},
     $FPDA  REQ   #content-architecture-class  {$FC|$FPC}
          }

  PERM  #dimensions     {REQ #horizontal-dimension
                  {REQ #fixed-dimension {<=14030}},
                 REQ #vertical-dimension
                  {REQ #fixed-dimension {<=19840)}}}
                         -- up to ISO A3 portrait --
                |{REQ #horizontal-dimension
                  {REQ #fixed-dimension {<=19840}},
                 REQ #vertical-dimension
                  {REQ #fixed-dimension {<=14030}}}
                         -- up to ISO A3 landscape --
                |{REQ #horizontal-dimension
                  {REQ #fixed-dimension {<=13200}},
                 REQ #vertical-dimension
                  {REQ #fixed-dimension {<=20400)}}}
                         -- up to ANSI B portrait --
                |{REQ #horizontal-dimension
                  {REQ #fixed-dimension {<=20400}},
                REQ #vertical-dimension
                 {REQ #fixed-dimension {<=13200}}}
                      -- up to ANSI B landscape --},

PERM  #medium-type   {PERM #nominal-page-size{$NominalPageSizes},
               PERM #side-of-sheet {ANY_VALUE}}


PERM  #layout-path    {'0-degrees'|'180-degrees'|'270-degrees'},

PERM  #type-of-coding  {ASN.1{2 8 3 6 0} -- character encoding --
               |ASN.1{2 8 3 7 0} -- T.6 encoding --
               |ASN.1{2 8 3 7 1} -- T.4 one dimensional
                                   encoding --
               |ASN.1{2 8 3 7 2} -- T.4 two dimensional
                                   encoding --
               |ASN.1{2 8 3 7 4} -- bitmap encoding --},

PERM #character-content-defaults  {
  PERM #alignment                  {ANY_VALUE),
  PERM #character-fonts             {ANY_VALUE},
  PERM #character-path              {'0-degrees'
                     |'90-degrees'
                     |'180 degrees'
                     |'270-degrees'},
  PERM #character-spacing            {ANY_VALUE},
  PERM #character-orientation         {'0-degrees'
                      |'90-degrees'},
  PERM #code-extension-announcers      {$CDEXTAN},
  PERM #first-line-format           {ANY_VALUE},
  PERM #graphic-character-sets        {$BASIC-GRCHAR,
                     $DAP-DEFAULT-GRCHAR},
  PERM #graphic-character-subrepertoire {ANY_VALUE},
  PERM #graphic-rendition            {$GRAPHICRENDITIONS},
  PERM #indentation                {ANY_VALUE},
  PERM #initial-offset             {ANY_VALUE},
  PERM #itemisation                {ANY_VALUE},
  PERM #kerning-offset              {ANY_VALUE},
  PERM #line-layout-table            {ANY_VALUE},
  PERM #line-progression             {'90-degrees'
                     |'270-degrees'},
        PERM #line-spacing              {ANY_VALUE},
        PERM #orphan-size               {ANY_VALUE},
        PERM #proportional-line-spacing     {ANY_VALUE},
        PERM #widow-size                {ANY_VALUE}}},

      PERM #raster-graphic-content-defaults  {
        PERM #clipping                {ANY_VALUE},
        PERM #image-dimensions              {ANY_VALUE},
        PERM #pel-spacing               {ANY_VALUE},
        PERM #spacing-ratio             {ANY_VALUE},
        PERM #compression                {ANY_VALUE}}},

    REQ   Document-architecture-class   {$FDA|$PDA|$FPDA},

    REQ   Content-architecture-classes  {[$FC],[$PC],[$FPC],[$FPR],[$FPG]},

    REQ   Interchange-format            {~~'if a'~~ -- See clause 8 for the definition of the
                                        permitted values for this attribute.--},
```

Editor's Note: The applicability of this attribute is dependent on the encoding being supported.

```
    REQ   Oda-version        {REQ #standard-or-recommendation("ISO 8613"),
                         REQ #publication-date(-- to be supplied)}},


     -- non basic document characteristics --

    PERM  Profile-character-sets     {$PROFCHAR},

    PERM  Comments-character-sets    {$COMCHAR},

    PERM  Alternative-representation-character-sets  {$ALTCHAR},

    PERM  Page-dimensions  {PMUL
                     {REQ #horizontal-dimension
                       {REQ #fixed-dimension {9241..14030}},
                      REQ #vertical-dimension
                       {REQ #fixed-dimension {12401..19840}}}
                              -- up to ISO A3 portrait --
                    |{REQ #horizontal-dimension
                       {REQ #fixed-dimension {12401..19840}},
                REQ #vertical-dimension
                  {REQ #fixed-dimension {9241..14030}}}
                              -- up to ISO A3 landscape --
              |{REQ #horizontal-dimension
                  {REQ #fixed-dimension {9241..13200}},
                REQ #vertical-dimension
                  {REQ #fixed-dimension {12401..20400}}}
                              -- up to ANSI B portrait --
              |{REQ #horizontal-dimension
                  {REQ #fixed-dimension {12401..20400}},
                REQ #vertical-dimension
                  {REQ #fixed-dimension {9241..13200}}}
                              -- up to ANSI B landscape --},

 -- any value of dimensions which is greater than the common assured
    reproduction area of ISO A4 and NAL is non-basic --

PERM  Medium-type     {PMUL
               {PERM #nominal-page-size{$NominalPageSizes},
                PERM #side-of-sheet{'recto'|'verso'}}},

   -- all values of "medium type" are non-basic --

PERM  Layout-path         {'0-degrees','90-degrees','180-degrees'},

PERM  Border            {ANY_VALUE},

PERM  Ra-gr-coding-attributes  {
    PERM #compression                {ANY_VALUE}},

PERM  Presentation-features {
  PERM #character-presentation-features  {
     PERM #character-orientation       {'90-degrees'},
     PMUL #character-path            {'90-degrees'
                              |'180-degrees'
                              |'270-degrees'},
     PMUL #character-spacing           {<100},
     PMUL #graphic-character-sets       {ANY_EXCEPT $BASIC-GRCHAR},
     PMUL #graphic-character-
                subrepertoire     {ANY_VALUE},
     PMUL #line-spacing              {ANY_EXCEPT 150,200,300,400},
      PERM #line-progression           {'90-degrees'}},

    PERM #raster-graphics-presentation-features {
      PMUL #pel-spacing    {REQ #length{ANY_EXCEPT 16,12,8,6,5,4,3,2,1}
                            REQ #pel-spaces{ANY_EXCEPT 1}}}},


    -- additional document characteristics

  PERM  Fonts-list            {PMUL{REQ #font-identifier {ANY_VALUE},
                            REQ #font-reference {ANY_VALUE}}},

    -- the format of the parameter "font-reference" is defined in
       clause 8.4 --

    -- document management attributes   {

    -- document -description --
  PERM  Title                     {ANY_STRING},
  PERM  Subject                     {ANY_STRING},
  PERM  Document-type                  {ANY_STRING},
  PERM  Abstract                    {ANY_STRING},
  PERM  Keywords                     {ANY_STRING},
  REQ   Document-reference              {ANY_VALUE},

    -- dates and times --
  PERM  Document-date-and-time          {ANY_STRING},
  PERM  Creation-date-and-time         {ANY_STRING},
  PERM  Local-filing-date-and-time     {ANY_STRING},
  PERM  Expiry-date-and-time          {ANY_STRING},
  PERM  Start-date-and-time          {ANY_STRING},
  PERM  Purge-date-and-time           {ANY_STRING},
  PERM  Release-date-and-time          {ANY_STRING},
  PERM  Revision-history            {ANY_VALUE},

    -- originators --
  PERM  Organizations               {ANY_STRING},
  PERM  Preparers                  {ANY_VALUE},
  PERM  Owners                     {ANY_VALUE},
  PERM  Authors                   {ANY_VALUE},
   -- other user information --
 PERM  Copyright                  {ANY_VALUE},
 PERM  Status                    {ANY_STRING},
 PERM  User-specific-codes          {ANY_STRING},
 PERM  Distribution-list            {ANY_VALUE},
 PERM  Additional-information        {ANY_VALUE},

  -- external references --
 PERM  References-to-other-documents  {ANY_VALUE},
 PERM  Superseded-documents         {ANY_VALUE},

   -- local file references --
 PERM  Local-file-references        {ANY_VALUE},

   -- content attributes --
 PERM  Document-size               {ANY_INTEGER},
 PERM  Number-of-pages              {ANY_INTEGER},
 PERM  Languages                  {ANY_STRING},

   -- security information --
 PERM  Authorization              {ANY_VALUE},
 PERM  Security-classification       {ANY_STRING},
 PERM  Access-rights               {ANY_STRING}}
```

7.2 Logical constituent constraints

7.2.1 Macro definitions

```
DEFINE(DocLogRootGFS, "
<construction-expr>      ::= <construction-term>
                   |<construction-type>;

<construction-term>      ::= <construction-factor>
                   |OPT <construction-factor>
                   |REP <construction-factor>
                   |OPT REP <construction-factor>;

<construction-type>      ::= SEQ({<construction-term>}...)
                   |CHO({<construction-term>}...);
<construction-factor>    ::= OBJECT_CLASS_ID_OF(Passage)
                         |<construction-type>;
                     ")


DEFINE(CONSTRAINT-1, "
<constraint-1>           ::= <construction-term>
                         |<construction-type>;

<construction-term>      ::= <construction-factor>
                         |OPT <construction-factor>
                         |REP <construction-factor>
                         |OPT REP <construction-factor>;

<construction-type>      ::= SEQ({<construction-term>}...)
                         |CHO({<construction-term>}...);

<construction-factor>    ::= OBJECT_CLASS_ID_OF(Paragraph)
                         |OBJECT_CLASS_ID_OF(BodyText)
                         |OBJECT_CLASS_ID_OF(BodyRaster)
                         |OBJECT_CLASS_ID_OF(BodyGeometric)
                         |<construction-type>;
                   ")


DEFINE(CONSTRAINT-2 "
<constraint-2>           ::= OBJECT_CLASS_ID_OF(NumberedSegment)
                         |OPT REP OBJECT_CLASS_ID_OF(NumberedSegment)
                         |REP OBJECT_CLASS_ID_OF(NumberedSegment)
                         |OPT OBJECT_CLASS_ID_OF(NumberedSegment)
                         |CHO({OBJECT_CLASS_ID_OF(NumberedSegment)}...);
                  ")


DEFINE(PassageGFS, "
<construction-expr>      ::= <constraint-1>
                         |<constraint-2>
                         |SEQ(<constraint-1><constraint-2>);
$CONSTRAINT-1
$CONSTRAINT-2     ")
```

```
DEFINE(NumberedSegmentGFS, "
<construction-expr>      ::= SEQ(<constraint-3>[<constraint-1>]
                          [<constraint-2>]);

<constraint-3>          ::= OBJECT_CLASS_ID_OF(Number);

$CONSTRAINT-1
$CONSTRAINT-2
                   ")


DEFINE(ParagraphGFS, "
<construction-expr>      ::= <construction-term>
                       |<construction-type>;

<construction-term>      ::= <construction-factor>
                       |OPT <construction-factor>
                       |REP <construction-factor>
                       |OPT REP <construction-factor>;

<construction-type>      ::= SEQ({<construction-term>}...)
                       |CHO({<construction-term>}...);

<construction-factor>    ::= OBJECT_CLASS_ID_OF(BodyText)
                       |OBJECT_CLASS_ID_OF(BodyRaster)
                       |OBJECT_CLASS_ID_OF(BodyGeometric)
                       |OBJECT_CLASS_ID_OF(Footnote)
                       |<construction-type>;
               ")


DEFINE(FootnoteGFS,  "
<construction-expr>      ::= SEQ(OBJECT_CLASS_ID_OF(FootnoteReference)
                          OBJECT_CLASS_ID_OF(FootnoteBody));
               ")


DEFINE(FootnoteBodyGFS, "
<construction-expr>      ::= SEQ(OBJECT_CLASS_ID_OF(FootnoteNumber
                                   <constraint-4>);
<constraint-4>          ::= OBJECT_CLASS_ID_OF(FootnoteText)
                       |REP(OBJECT_CLASS_ID_OF(FootnoteText))
                       |CHO({OBJECT_CLASS_ID_OF(FootnoteText)}...)
                       |REP CHO({OBJECT_CLASS_ID_OF(FootnoteText)}...);
                 ")


DEFINE(CommonContentGFS, "
<construction-expr>      ::= <construction-factor>
                       |SEQ(<construction-factor>...)

<construction-factor>    ::= OBJECT_CLASS_ID_OF(CommonText)
                       |OBJECT_CLASS_ID_OF(PageNumber)
                       |OBJECT_CLASS_ID_OF(CommonRaster)
                       |OBJECT_CLASS_ID_OF(CommonGeometric);
                 ")


DEFINE(N,          "
<n>     ::=  -- any string of characters from the set of
              characters: "0"..."9"
                ")


DEFINE(PREFIXES,    "
<prefixes>   ::=  'prefixes-'<n>;
$N
                ")


DEFINE(SUFFIXES,    "
<suffixes>   ::=  'suffixes-'<n>;
$N
                ")


DEFINE(SEPARATORS,   "
<separators> ::=  'separators-'<n>;
$N
                ")
```

```
DEFINE(NUMBERS,      "
<numbers>    ::=  'numbers-'<n>;
$N
                 ")


DEFINE(NUMBERSTRING, "
<numberstring> ::= 'numberstring-'<n>;
$N               ")


DEFINE(STRINGFUNCTION, "
<string-function>        ::=  MK_STR|U_ALPHA|L_ALPHA|U_ROM|L_ROM;
                 ")


DEFINE(INITIALISEANY, "
                  {REQ #binding-identifer{<prefixes>},
                   REQ #binding-value{ANY_STRING}
                  |{REQ #binding-identifer{<suffixes>},
                   REQ #binding-value{ANY_STRING}
                  |{REQ #binding-identifer{<separators>},
                   REQ #binding-value{ANY_STRING}
                  |{REQ #binding-identifer{<numbers>},
                   REQ #binding-value{ANY_INTEGER}
                  |{REQ #binding-identifer{<numberstring>},
                   REQ #binding-value{ANY_STRING}
$PREFIXES
$SUFFIXES
$SEPARATORS
$NUMBERS
$NUMBERSTRING
                 ")


DEFINE(USENUMBERSTRINGS, "
                  {REQ #binding-identifer{<numberstring>},
                   REQ #binding-value{<hierarchic-expr>|<simple-expr>}

<hierarchic-expr>        ::=  B_REF(SUP(CURR_OBJ))(<numberstring>)
                      +B_REF(SUP(CURR-OBJ))(<separator>))
```

```
                       +<simple-expr>;

<simple-expr>            ::= <string-function>(B_REF(CURR-OBJ)(<numbers>))
                       |<string-function>(ORD(CURR_OBJ));

$NUMBERSTRING
$SEPARATORS
$NUMBERS
$STRINGFUNCTION
                     ")


DEFINE(USENUMBERS, "
                    {REQ #binding-identifer{numbers>},
                     REQ #binding-value
                         {INC(B_REF(PREC(CURR_OBJ))(<numbers>)}
$NUMBERS
                  ")


DEFINE(SEGMENTNUMBER,  ")
<string-expr>           ::= [<pre-st>]<num-st>[suf-st>];

<num-str>               ::= B_REF(SUP(CURR_OBJ))(<numberstring>);

<pre-st>               ::= B_REF(SUP(CURR_OBJ))(<prefixes>)
                       |ANY_STRING;

<suf-st>               ::= B_REF(SUP(CURR_OBJ))(<suffixes>)
                       |ANY_STRING;

$NUMBERSTRING
$PREFIXES
$SUFFIXES
                    ")


DEFINE(INITIALISEFNOTE   "
                    {REQ #binding-identifer{"fnotenumber"},
                     REQ #binding-value{>=0}
                    ")
```

```
DEFINE(INCFNOTENUMBER  "
                  {REQ #binding-identifer{"fnotenumber"},
                   REQ #binding-value{INC(B_REF(PREC
                                (CURR-OBJ))(fnotenumber)}
                 ")


DEFINE(FNOTENUMBERSTRING  "
                  {REQ #binding-identifer{"fnotestring"},
                   REQ #binding-value{<string-function>
                               B_REF(CURR_OBJ)(fnotenumber)}

<string-function>       ::=  MK_STR|U_ALPHA|L_ALPHA|U_ROM|L_ROM;
               ")


DEFINE(FNOTESTRINGLITERAL  "
                  {REQ #binding-identifer{"fnotestring"},
                   REQ #binding-value{ANY_STRING}
                    ")


DEFINE(FNOTENUMBER  "
<string-expr>          ::=  [ANY_STRING],<num-str>[ANY_STRING];

<num-str>              ::=  B_REF(SUP(CURR_OBJ))(fnotestring);
               ")



DEFINE(HEADERFOOTERSTRING  "
<string-expr>    ::=   [ANY_STRING]{<string-function><num-exp>}[ANY_STRING];

<num-exp>        ::=   B_REF(SUP(CURR_INST(<class-or-type1>,
                  CURR_OBJ)))(PGnum)
                 |B_REF(CURR_INST(<class-or-type2>,
                  CURR_OBJ)))(PGnum);

<class-or-type1> ::=  FRAME;
<class-or-type2>  ::= PAGE
                 |OBJECT_CLASS_ID_OF(Page)
                 |OBJECT_CLASS_ID_OF(RectoPage)
                 |OBJECT_CLASS_ID_OF(VersoPage);

$STRINGFUNCTION
              ")
```

7.2.2 Factor constraints

7.2.2.1 FACTOR: ANY-LOGICAL

```
                                   {

GENERIC:
  REQ   Object-type              {VIRTUAL},
  REQ   Object-class-identifier    {ANY_VALUE},
SPECIFIC:
  PERM  Object-type              {VIRTUAL},
  REQ   Object-identifier        {ANY_VALUE},
  REQ   Object-class             {VIRTUAL},
SPECIFIC_AND_GENERIC:
  PERM  User-readable-comments     {ANY_STRING},
  PERM  User-visible-name        {ANY_STRING}}
```

7.2.3 Constituent constraints

7.2.3.1 DocumentLogicalRoot

```
                                   :ANY-LOGICAL {

GENERIC:
  REQ   Object-type              {'document-logical-root'},
  REQ   Generator-for-subordinates  {$DocLogRootGFS},
  REQ   Application-comments       {REQ #constraint-name {"0"},
                          PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM  Object-type              {'document-logical-root'},
  REQ   Object-class            {OBJECT_CLASS_ID_OF
                               (DocumentLogicalRoot)},
  REQ   Subordinates            {SUB_ID_OF(Passage)+},
  PERM  Application-comments       {REQ #constraint-name {"0"},
                     PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
  PERM  Bindings               {PMUL{$INITIALISEANY},
                     PERM{$INITIALISEFNOTE}}
```

7.2.3.2 Passage

```
                              :ANY-LOGICAL {

GENERIC:
  REQ   Object-type            {'composite-logical-object'},
  REQ   Generator-for-subordinates  {$PassageGFS},
  REQ   Application-comments       {REQ #constraint-name {"1"},
                     PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM  Object-type             {'composite-logical-object'},
  REQ   Object-class           {OBJECT_CLASS_ID_OF(Passage)},
  REQ   Subordinates            {SUB_ID_OF(NumberedSegment)+,
                     SUB_ID_OF(BodyText)+,
                     SUB_ID_OF(BodyRaster)+,
                     SUB_ID_OF(BodyGeometric)+,
                     SUB_ID_OF(Paragraph)+},
  PERM  Application-comments       {REQ #constraint-name {"1"},
                     PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
  PERM  Layout-style            {STYLE_ID_OF(L-Style1)},
  PERM  Bindings               {PMUL{$INITIALISEANY},
                     PERM{$INITIALISEFNOTE}}
```

7.2.3.3 NumberedSegment

```
                              :ANY-LOGICAL {

GENERIC:
  REQ   Object-type            {'composite-logical-object'},
  REQ   Generator-for-subordinates  {$NumberedSegmentGFS},
   REQ    Application-comments       {REQ #constraint-name {"2"},
                              PERM #external-data {ANY_VALUE}},
   PERM   Bindings                 {PMUL{$INITIALISEANY},
                              PERM{$USENUMBERS},
                              PERM{$USENUMBERSTRING}},
SPECIFIC:
   PERM   Object-type               {'composite-logical-object'},
   REQ    Object-class             {OBJECT_CLASS_ID_OF(NumberedSegment)},
   REQ    Subordinates              {SUB_ID_OF(Number),
                              SUB_ID_OF(NumberedSegment)+,
                              SUB_ID_OF(BodyText)+,
                              SUB_ID_OF(BodyRaster)+,
                              SUB_ID_OF(BodyGeometric)+,
                              SUB_ID_OF(Paragraph)+},
   PERM   Bindings                 {PMUL{$INITIALISEANY}},
   PERM   Application-comments       {REQ #constraint-name {"2"},
                              PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM   Layout-style              {STYLE_ID_OF(L-Style2)}}
```

7.2.3.4 Number

```
                                     :ANY-LOGICAL {

GENERIC:
   REQ    Object-type              {'basic-logical-object'},
   REQ    Content-generator          {$SEGMENTNUMBER},
   REQ    Application-comments        {REQ #constraint-name {"3"},
                              PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM   Object-type               {'basic-logical-object'},
   REQ    Object-class             {OBJECT_CLASS_ID_OF(Number)},
   PERM   Content-portions            {CONTENT_ID_OF(Character-
                                      content-portion)+},
   PERM   Application-comments        {REQ #constraint-name {"3"},
                              PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM   Layout-style              {STYLE_ID_OF(L-Style3)},
   PERM   Presentation-style          {STYLE_ID_OF(P-Style1)},
   PERM   Content-Architecture-Class  {$FC|$PC|$FPC}}
```

7.2.3.5 Paragraph

```
                                          :ANY-LOGICAL {

GENERIC:
   REQ    Object-type                {'composite-logical-object'},
   REQ    Generator-for-subordinates  {$ParagraphGFS},
   REQ    Application-comments        {REQ #constraint-name {"6"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM   Object-type                {'composite-logical-object'},
   REQ    Object-class               {OBJECT_CLASS_ID_OF
                                      (Paragraph)},
   REQ    Subordinates               {SUB_ID_OF(BodyText)+,
                             SUB_ID_OF(Footnote)+,
                             SUB_ID_OF(BodyRaster)+,
                             SUB_ID_OF(BodyGeometric)+},
   PERM   Application-comments        {REQ #constraint-name {"6"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM   Layout-style               {STYLE_ID_OF(L-Style2)}}
```

7.2.3.6 BodyText

```
                                          :ANY-LOGICAL {

GENERIC:
   REQ    Object-type                {'basic-logical-object'},
   PERM   Resource                   {ANY_VALUE},
   REQ    Application-comments        {REQ #constraint-name {"14"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM   Object-type                {'basic-logical-object'},
   REQ    Object-class               {OBJECT_CLASS_ID_OF(BodyText)},
   PERM   Application-comments        {REQ #constraint-name {"14"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM   Layout-style               {STYLE_ID_OF(L-Style3)},
   PERM   Presentation-style           {STYLE_ID_OF(P-Style1)},
   PERM   Content-architecture-class  {$FC|$PC|$FPC},
   PERM   Content-portions           {CONTENT_ID_OF(
                                 Character-content-portion)+}}

  -- the attribute "content portion" must be specified either in the
     specific or generic part, otherwise the attribute "resource"
     must be specified --
```

7.2.3.7 BodyGeometric

```
                                         :ANY-LOGICAL {

GENERIC:
   REQ    Object-type                 {'basic-logical-object'},
   REQ    Content-architecture-class  {$FPG},
   PERM   Resource                    {ANY_VALUE},
   REQ    Application-comments        {REQ #constraint-name {"18"},
                                 PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM   Object-type                 {'basic-logical-object'},
   REQ    Object-class                {OBJECT_CLASS_ID_OF(BodyGeometric)},
   PERM   Content-architecture-class  {$FPG},
   PERM   Application-comments        {REQ #constraint-name {"18"},
                                 PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM   Layout-style                {STYLE_ID_OF(L-Style5)},
   PERM   Presentation-style          {STYLE_ID_OF(P-Style2)},
   PERM   Content-portions            {CONTENT_ID_OF(
                                 Geometric-content-portion)}}

  -- the attribute "content portion" must be specified either in the
     specific or generic part, otherwise the attribute "resource"
     must be specified --
```

7.2.3.8 BodyRaster

```
                                         :ANY-LOGICAL {

GENERIC:
  REQ   Object-type              {'basic-logical-object'},
  REQ   Content-architecture-class  {$FPR},
  PERM  Resource                 {ANY_VALUE},
  REQ   Application-comments       {REQ #constraint-name {"17"},
                            PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM  Object-type              {'basic-logical-object'},
  REQ   Object-class             {OBJECT_CLASS_ID_OF(BodyRaster)},
  PERM  Content-architecture-class  {$FPR},
  PERM  Application-comments       {REQ #constraint-name {"17"},
                            PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
  PERM  Layout-style             {STYLE_ID_OF(L-Style5)},
  PERM  Presentation-style         {STYLE_ID_OF(P-Style3)},
  PERM  Content-portions           {CONTENT_ID_OF(
                              Raster-content-portion)}}

 -- the attribute "content portion" must be specified either in the
   specific or generic part, otherwise the attribute "resource"
   must be specified --
```

7.2.3.9 Footnote

```
                                :ANY-LOGICAL {

GENERIC:
  REQ   Object-type              {'composite-logical-object'},
  REQ   Generator-for-subordinates  {$FootnoteGFS},
  PERM  Bindings                 {{$INCFNOTENUMBER},
                            {$FNOTENUMBERSTRING
                            |$FNOTESTRINGLITERAL},
  REQ   Application-comments       {REQ #constraint-name {"8"},
                            PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM  Object-type              {'composite-logical-object'},
  REQ   Object-class             {OBJECT_CLASS_ID_OF(Footnote)},
  REQ   Subordinates              {SUB_ID_OF(FootnoteReference),
                            SUB_ID_OF(FootnoteBody)},
  PERM  Bindings                 {$FNOTESTRINGLITERAL},
  PERM  Application-comments       {REQ #constraint-name {"8"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM  Layout-style            {STYLE_ID_OF(L-Style4)}}
```

7.2.3.10 FootnoteReference

```
                              :ANY-LOGICAL {

GENERIC:
   REQ   Object-type            {'basic-logical-object'},
   REQ   Content-generator         {$FNOTENUMBER},
   REQ   Application-comments        {REQ #constraint-name {"10"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM  Object-type             {'basic-logical-object'},
   REQ   Object-class           {OBJECT_CLASS_ID_OF
                                (FootnoteReference)},
   PERM  Content-portions          {CONTENT_ID_OF(Character-
                                   content-portion)+},
   PERM  Application-comments        {REQ #constraint-name {"10"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
   PERM  Layout-style            {STYLE_ID_OF(L-Style3)},
   PERM  Presentation-style        {STYLE_ID_OF(P-Style1)},
   PERM  Content-architecture-class  {$PC|$FPC}}
```

7.2.3.11 FootnoteBody

```
                              :ANY-LOGICAL {

GENERIC:
   REQ   Object-type            {'composite-logical-object'},
   REQ   Generator-for-subordinates  {$FootnoteBodyGFS},
   REQ   Application-comments        {REQ #constraint-name {"11"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC:
   PERM  Object-type             {'composite-logical-object'},
   REQ   Object-class           {OBJECT_CLASS_ID_OF(FootnoteBody)},
   REQ   Subordinates            {SUB_ID_OF(FootnoteNumber,
                        SUB_ID_OF(FootnoteText)+},
  PERM  Application-comments      {REQ #constraint-name {"11"},
                        PERM #external-data {ANY_VALUE}}}
```

7.2.3.12 FootnoteNumber

```
                              :ANY-LOGICAL {

GENERIC:
  REQ   Object-type              {'basic-logical-object'},
  REQ   Content-generator          {$FNOTENUMBER},
  REQ   Application-comments        {REQ #constraint-name {"9"},
                        PERM #external-data {ANY_VALUE}},
  REQ   Layout-style             {STYLE_ID_OF(L-Style9)},
SPECIFIC:
  PERM  Object-type              {'basic-logical-object'},
  REQ   Object-class             {OBJECT_CLASS_ID_OF(FootnoteNumber)},
  PERM  Content-portions           {CONTENT_ID_OF(Character-
                                   content-portion)+},
  PERM  Application-comments       {REQ #constraint-name {"9"},
                        PERM #external-data {ANY_VALUE}},
  PERM  Layout-style             {STYLE_ID_OF(L-Style9)},
SPECIFIC_AND_GENERIC:
  PERM  Presentation-style         {STYLE_ID_OF(P-Style1)},
  PERM  Content-architecture-class  {$FC|$PC|$FPC}}
```

7.2.3.13 FootnoteText

```
                              :ANY-LOGICAL {

GENERIC:
  REQ   Object-type              {'basic-logical-object'},
  REQ   Application-comments        {REQ #constraint-name {"12"},
                        PERM #external-data {ANY_VALUE}},
  REQ   Layout-style             {STYLE_ID_OF(L-Style6)},
SPECIFIC:
  PERM  Object-type              {'basic-logical-object'},
  REQ   Object-class             {OBJECT_CLASS_ID_OF(FootnoteText)},
  REQ   Content-portions          {CONTENT_ID_OF(Character-
                                  content-portion)+)},
  PERM   Application-comments       {REQ #constraint-name {"12"},
                                PERM #external-data {ANY_VALUE}},
  PERM   Layout-style             {STYLE_ID_OF(L-Style6)},
SPECIFIC_AND_GENERIC:
  PERM   Presentation-style        {STYLE_ID_OF(P-Style1)},
  PERM   Content-architecture-class  {$FC|$PC|$FPC}}
```

7.2.3.14 CommonContent

```
                                {

GENERIC:
  REQ    Object-type              {'composite-logical-object'},
  REQ    Object-class-identifier    {ANY_VALUE},
  REQ    Generator-for-subordinates  {$CommonContentGFS},
  REQ    Application-comments       {REQ #constraint-name {"19"},
                                PERM #external-data {ANY_VALUE}},
  PERM   User-readable-comments     {ANY_STRING},
  PERM   User-visible-name         {ANY_STRING}}
```

7.2.3.15 CommonText

```
                                {

GENERIC:
  REQ    Object-type              {'basic-logical-object'},
  REQ    Object-class-identifier    {ANY_VALUE},
  PERM   Content-portion           {CONTENT_ID_OF(Character-
                               content-portion)+},
  PERM   Resource                {ANY_VALUE},
  PERM   Layout-style             {STYLE_ID_OF(L-Style7)},
  PERM   Presentation-style        {STYLE_ID_OF(P-Style4)},
  PERM   Content-architecture-class  {$FC|$PC|$FPC},
  REQ    Application-comments       {REQ #constraint-name {"20"},
                                PERM #external-data {ANY_VALUE}},

  PERM   User-readable-comments     {ANY_STRING},
  PERM   User-visible-name         {ANY_STRING}}
```

```
-- either the attribute "content portion" or "resource" must be specified
   in the above constituent --
```

7.2.3.16 PageNumber

```
                                        {

GENERIC:
   REQ    Object-type                {'basic-logical-object'},
   REQ    Object-class-identifier    {ANY_VALUE},
   REQ    Content-generator          {$HEADERFOOTERSTRING},
   PERM   Layout-style               {STYLE_ID_OF(L-Style7)},
   PERM   Presentation-style         {STYLE_ID_OF(P-Style4)},
   PERM   Content-architecture-class {$FC|$PC|$FPC},
   REQ    Application-comments       {REQ #constraint-name {"40"},
                               PERM #external-data {ANY_VALUE}},
   PERM   User-readable-comments     {ANY_STRING},
   PERM   User-visible-name          {ANY_STRING}}
```

7.2.3.17 CommonGeometric

```
                                        {

GENERIC:
   REQ    Object-type                {'basic-logical-object'},
   REQ    Object-class-identifier    {ANY_VALUE},
   PERM   Content-portions           {CONTENT_ID_OF(
                                  Geometric-content-portion)},
   PERM   Resource                   {ANY_VALUE},
   PERM   Layout-style               {STYLE_ID_OF(L-Style8)},
   PERM   Presentation-style         {STYLE_ID_OF(P-Style2)},
   REQ    Content-architecture-class {$FPG},
   REQ    Application-comments       {REQ #constraint-name {"22"},
                               PERM #external-data {ANY_VALUE}},
   PERM   User-readable-comments     {ANY_STRING},
   PERM   User-visible-name          {ANY_STRING}}

-- either the attribute "content portion" or "resource" must be specified
   in the above constituent --
```

7.2.3.18 CommonRaster

```
                                   {

GENERIC
   REQ    Object-type           .  {'basic-logical-object'},
   REQ    Object-class-identifier   {ANY_VALUE},
   PERM   Content-portions          {CONTENT_ID_OF(
                             Raster-content-portion)},
   PERM   Resource                 {ANY_VALUE},
   PERM   Layout-style             {STYLE_ID_OF(L-Style8)},
   PERM   Presentation-style        {STYLE_ID_OF(P-Style3)},
   REQ    Content-architecture-class  {$FPR},
   REQ    Application-comments        {REQ #constraint-name {"21"},
                              PERM #external-data {ANY_VALUE}},
   PERM   User-readable-comments     {ANY_STRING},
   PERM   User-visible-name          {ANY_STRING}}

  -- either the attribute "content portion" or "resource" must be specified
     in the above constituent --
```

7.3 Layout constituent constraints

7.3.1 Macro definitions

```
DEFINE(DocLayRootGFS,  "
<construction-expr>    ::= <construction-term>
                   |<construction-type>;

<construction-term>    ::= <construction-factor>
                   |OPT <construction-factor>
                   |REP <construction-factor>
                   |OPT REP <construction-factor>;

<construction-type>    ::= SEQ({<construction-term>}...)
                   |CHO({<construction-term>}...);

<construction-factor>  ::= OBJECT_CLASS_ID_OF(PageSet)
```

```
                |<construction-type>;
               ")


DEFINE(PageSetGFS,    "
<construction-expr>    ::= <PageSet-1>
                |<PageSet-2>
                |<PageSet-3>
                |<SEQ(<PageSet-1><PageSet-2>)
                |<SEQ(<PageSet-1><PageSet-3>);

<PageSet-1>          ::= OBJECT_CLASS_ID_OF(Page)
                |OPT(OBJECT_CLASS_ID_OF(Page));

<PageSet-2>          ::= REP(OBJECT_CLASS_ID_OF(Page))
                |OPT REP(OBJECT_CLASS_ID_OF(Page));

<PageSet-3>          ::= OPT REP(SEQ(OBJECT_CLASS_ID_OF(RectoPage)
                    OPT(OBJECT_CLASS_ID_OF(VersoPage))))
                |OPT REP(SEQ(OBJECT_CLASS_ID_OF(VersoPage)
                     OPT(OBJECT_CLASS_ID_OF(RectoPage))))
                |REP(SEQ(OBJECT_CLASS_ID_OF(RectoPage)
                   OPT(OBJECT_CLASS_ID_OF(VersoPage))))
                |REP(SEQ(OBJECT_CLASS_ID_OF(VersoPage)
                   OPT(OBJECT_CLASS_ID_OF(RectoPage))));
            ")


DEFINE(PageGFS,         "
<construction-expr>  ::= SEQ([<headerarea>]<bodyarea>[<footerarea>])
               |<bodyarea>;

<headerarea>        ::= OBJECT_CLASS_ID_OF(BasicHeader)
               |OBJECT_CLASS_ID_OF(CompositeHeader);

<bodyarea>          ::= OBJECT_CLASS_ID_OF(VariableCompositeBody)
               |OBJECT_CLASS_ID_OF(BodyFrameVariable);

<footerarea>        ::= OBJECT_CLASS_ID_OF(BasicFooter)
               |OBJECT_CLASS_ID_OF(CompositeFooter);
              ")
```

```
DEFINE(VariableCompositeBodyGFS,   "
<construction-expr>  ::= <construction-term>
                     |<construction-type>
                     |SEQ(<construction-term>, <construction-footnote)
                     |SEQ(<construction-type>, <construction-footnote);

<construction-term>    ::= <construction-factor>
                      |OPT <construction-factor>
                      |REP <construction-factor>
                      |OPT REP <construction-factor>;

<construction-type>    ::= SEQ({<construction-term>}...)
                      |CHO({<construction-term>}...);

<construction-factor>  ::= OBJECT_CLASS_ID_OF(BasicFloat)
                      |OBJECT_CLASS_ID_OF(SnakingColumns)
                      |OBJECT_CLASS_ID_OF(SynchronizedColumns)
                      |<construction-type>;

<construction-footnote>   ::= OBJECT_CLASS_ID_OF(FootnoteArea)
                         |OPT OBJECT_CLASS_ID_OF(FootnoteArea);
                    ")


DEFINE(SnakingColumnsGFS,    "
<construction-expr>    ::=  SEQ({OBJECT_CLASS_ID_OF(ColumnVariable)}...)
                       |REP OBJECT_CLASS_ID_OF(ColumnVariable);
                            ")


DEFINE(SynchronizedColumnsGFS,    "
<construction-expr>    ::=  SEQ({OBJECT_CLASS_ID_OF(ColumnFixed)}...);


DEFINE(HeaderFooterGFS,    "
<construction-expr>    ::= <fixed-common-content-frames>
                      |<variable-common-content-frames>;

<fixed-common-content-frames>
                    ::= SEQ({OBJECT_CLASS_ID_OF(SourcedContentFixed)
                    |OBJECT_CLASS_ID_OF(ArrangedContentFixed)}...);

<variable-common-content-frames>
            ::= SEQ({OBJECT_CLASS_ID_OF(SourcedContentVariable)
                 |OBJECT_CLASS_ID_OF(ArrangedContentVariable)}...);
                 ")


DEFINE(PAGENUMBER, "
             {REQ #binding-identifier{"PGnum"},
              REQ #binding-value{INC(B_REF(PREC(CURR-OBJ))("PGnum"))}
             |{REQ #binding-identifier{"PGnum"},
              REQ #binding-value{ORD(CURR-OBJ)}
             ")


DEFINE(INITIALISEPGNUM, "
              REQ #binding-identifier{"PGnum"},
              REQ #binding-value{>=-1}
                ")


DEFINE(PDA-FPDA, "{'processable'|'formatted-processable'}")
```

7.3.2 Factor constraints

7.3.2.1 FACTOR: ANY-LAYOUT

```
                              {

GENERIC:
   REQ    Object-type              {VIRTUAL},
   REQ    Object-class-identifier    {ANY_VALUE},
   REQ    Application-comments       {VIRTUAL},
SPECIFIC:
   PERM   Object-type              {VIRTUAL},
   REQ    Object-identifier         {ANY_VALUE},
   CASE  $DAC OF {
   $FDA:  PERM   Object-class        {VIRTUAL},
  $FPDA: REQ   Object-class        {VIRTUAL},
           }
  REQ   Subordinates            {VIRTUAL},
  PERM  Application-comments      {VIRTUAL},
SPECIFIC_AND_GENERIC:
  PERM  User-readable-comments     {ANY_VALUE},
  PERM  User-visible-name         {ANY_VALUE}}
```

7.3.2.2 FACTOR: ANY-PAGE

```
                                   :ANY-LAYOUT {

GENERIC:
   REQ   Object-type             {'page'},
   REQ   Generator-for-subordinates {$PageGFS},
   CASE  $DAC OF {
       $PDA:  PERM  Bindings     {$PAGENUMBER},
       $FPDA: PERM  Bindings     {$PAGENUMBER},
            }
SPECIFIC:
   PERM  Object-type             {'page'},
   REQ   Subordinates            {SUB_ID_OF(BasicHeader),
                        SUB_ID_OF(CompositeHeader),
                        SUB_ID_OF(VariableCompositeBody),
                        SUB_ID_OF(BodyFrameVariable),
                        SUB_ID_OF(BasicFooter),
                        SUB_ID_OF(CompositeFooter)}
   CASE  $DAC OF {
       $FPDA:  PERM  Bindings  {REQ #binding-identifier{"PGnum"},
                       REQ #binding-identifier{>=0}}
            }
SPECIFIC_AND_GENERIC:
   PERM  Dimensions              {{REQ #horizontal-dimension
                         {REQ #fixed-dimension {<=14030}},
                        REQ #vertical-dimension
                         {REQ #fixed-dimension {<=19840)}}}
                           -- up to ISO A3 portrait --
                       |{REQ #horizontal-dimension
                          {REQ #fixed-dimension {<=19840}},
                        REQ #vertical-dimension
```

```
                                    {REQ #fixed-dimension {<=14030}}},
                                     -- up to ISO A3 landscape --
                                 |{REQ #horizontal-dimension
                                    {REQ #fixed-dimension {<=13200}},
                                  REQ #vertical-dimension
                                    {REQ #fixed-dimension {<=20400)}}}
                                    -- up to ANSI B portrait --
                                 |{REQ #horizontal-dimension
                                    {REQ #fixed-dimension {<=20400}},
                                  REQ #vertical-dimension
                                    {REQ #fixed-dimension {<=13200}}}
                                    -- up to ANSI B landscape --},
   PERM  Page-Position              {ANY_VALUE}}
```

7.3.2.3 FACTOR: ANY-FRAME-FIXED

```
                                            :ANY-LAYOUT {

GENERIC:
   REQ    Object-type            {'frame'},
SPECIFIC:
   PERM   Object-type            {'frame'},
   REQ    Subordinates           {VIRTUAL},
SPECIFIC_AND_GENERIC:
   PERM   Position               {REQ #fixed-position
                                 {REQ #horizontal-position {ANY_VALUE},
                                  REQ #vertical-position {ANY_VALUE}}},
   PERM   Dimension                {REQ #horizontal-dimension
                                 {REQ #fixed-dimension {ANY_VALUE}},
                                 REQ #vertical-dimension
                                 {REQ #fixed-dimension {ANY_VALUE}}},
   PERM   Border                 {ANY_VALUE}}
```

7.3.2.4 FACTOR: ANY-FRAME-VARIABLE

```
                                            :ANY-LAYOUT {

GENERIC:
   REQ    Object-type            {'frame'},
SPECIFIC:
   PERM   Object-type            {'frame'},
   REQ   Subordinates            {VIRTUAL},
   CASE  $DAC OF {
        $FPDA: REQ  Position    {REQ #fixed-position
                               {REQ #horizontal-position {ANY_VALUE},
                                REQ #vertical-dimension {ANY_VALUE}}},
              REQ  Dimension    {REQ #horizontal-dimension
                               {REQ #fixed-dimension {ANY_VALUE}},
                               REQ #vertical-dimension
                               {REQ #fixed-dimension {ANY_VALUE}}}
              }
SPECIFIC_AND_GENERIC:
   CASE  $DAC OF {
        $FDA:  PERM  Position   {REQ #fixed-position
                                {REQ #horizontal-position {ANY_VALUE},
                                 REQ #vertical-position {ANY_VALUE}}},
               PERM  Dimension  {REQ #horizontal-dimension
                                {REQ #fixed-dimension {ANY_VALUE}},
                               REQ #vertical-dimension
                                {REQ #fixed-dimension {ANY_VALUE}}}
              }
   PERM  Border                {ANY-VALUE}
```

7.3.2.5 FACTOR: BLOCK

```
                               {

SPECIFIC:
    REQ  Object-type            {'block'},
    REQ  Object-identifier      {ANY_VALUE},
    REQ  Content-portions         {CONTENT_ID_OF
                               (character-content-portion)+,
                               CONTENT_ID_OF
                               (raster-graphics-content-portion),
                               CONTENT_ID_OF
                               (geometric-graphics-content-portion)},
    PERM Presentation-style       {STYLE_ID_OF(P-style1)
                               |STYLE_ID_OF(P-style2)
                               |STYLE_ID_OF(P-style3)},
    PERM Content-architecture-class {$FC|$FPC|$FPR|$FPG},
    PERM Presentation-attributes {
      PERM #character-attributes {
        PERM #alignment                {ANY_VALUE},
        PERM #character-fonts           {ANY_VALUE},
        PERM #character-spacing          {ANY_VALUE},
          PERM #character-orientation      {'90-degrees'|'180-degrees'
                                          |'270-degrees'},
        PERM #code-extension-announcers {$CDEXTAN},
        PERM #first-line-offset         {ANY_VALUE},
        PERM #graphic-character-sets    {$BASIC-GRCHAR},
        PERM #graphic-character-
                subrepertoire     {ANY-VALUE},
        PERM #graphic-rendition         {$GRAPHICRENDITIONS},
        PERM #itemisation               {ANY_VALUE},
        PERM #kerning-offset             {ANY_VALUE},
        PERM #line-layout-table          {ANY_VALUE},
        PERM #line-spacing              {150|200|300|400},
        PERM #initial-offset           {ANY_VALUE}}},
   PERM  User-readable-comments    {ANY_STRING},
   PERM  User-visible-name         {ANY_STRING},
   PERM  Position                 {REQ #fixed-position
                              {REQ #horizontal-position {ANY_VALUE},
                               REQ #vertical-position {ANY_VALUE}}},
   PERM  Dimension                  {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}}}}
```

7.3.3 Constituent constraints

7.3.3.1 DocumentLayoutRoot

```
                                 :ANY-LAYOUT {

GENERIC:
   REQ    Object-type              {'document-layout-root'},
   REQ    Generator-for-subordinates  {$DocLayRootGFS},
   CASE  $DAC OF {
       $PDA:  PERM   Bindings      {$INITIALISEPGNUM},
       $FPDA: PERM   Bindings       {$INITIALISEPGNUM},
          }
   REQ    Application-comments       {REQ #constraint-name {"0"},
                            PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM   Object-type             {'document-layout-root'},
  CASE  $DAC OF {
     $FDA:   PERM  Object-class {OBJECT_CLASS_ID_OF
                                  (DocumentLayoutRoot)},
     $FPDA:  REQ   Object-class {OBJECT_CLASS_ID_OF
                                  (DocumentLayoutRoot)},
         }
  REQ    Subordinates            {SUB_ID_OF(PageSet)+},
  PERM   Application-comments       {REQ #constraint-name {"0"},
                            PERM #external-data {ANY_VALUE}}}
```

7.3.3.2 PageSet

```
                                   :ANY-LAYOUT {

GENERIC:
  REQ    Object-type             {'pageset'},
  REQ    Generator-for-subordinates  {$PageSetGFS},
  CASE  $DAC OF {
     $PDA:  PERM   Bindings      {$INITIALISEPGNUM},
     $FPDA: PERM   Bindings      {$INITIALISEPGNUM}
         }
  REQ    Application-comments       {REQ #constraint-name {"1"},
                            PERM #external-data {ANY_VALUE}},
SPECIFIC:
  PERM   Object-type             {'pageset'},
  CASE  $DAC OF {
      $FDA:   PERM  Object-class  {OBJECT_CLASS_ID_OF(PageSet)},
      $FPDA:  REQ   Object-class  {OBJECT_CLASS_ID_OF(PageSet)}
         }
  REQ    Subordinates            {SUB_ID_OF(Page)+,
                            SUB_ID_OF(RectoPage)+,
                            SUB_ID_OF(VersoPage)+},
  PERM   Application-comments       {REQ #constraint-name {"1"},
                            PERM #external-data {ANY_VALUE}}}
```

7.3.3.3 Page

```
                                   :ANY-PAGE {

GENERIC:
  REQ   Application-comments      {REQ #constraint-name {"2"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
       $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(Page)},
       $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(Page)}
          }
  PERM  Application-comments      {REQ #constraint-name {"2"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC_AND_GENERIC:
  PERM  Medium-Type            {PERM #nominal-page-size
                                 {$NominalPageSizes},
                         PERM #side-of-sheet {ANY_VALUE}}}
```

7.3.3.4 RectoPage

```
                                   :ANY-PAGE {

GENERIC:
  REQ   Application-comments      {REQ #constraint-name {"3"},
                         PERM #external-data {ANY_VALUE}},
  REQ   Medium-Type            {REQ  #nominal-page-size
                                 {$NominalPageSizes},
                         REQ  #side-of-sheet
                             {'recto'|'unspecified'}},
SPECIFIC:
  CASE  $DAC OF {
       $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(RectoPage)}
       $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(RectoPage)}
          }
  PERM  Application-comments      {REQ #constraint-name {"3"},
                         PERM #external-data {ANY_VALUE}},
  PERM  Medium-Type            {PERM #nominal-page-size
                                 {$NominalPageSizes},
                         PERM #side-of-sheet
                             {'recto'|'unspecified'}}
```

7.3.3.5 VersoPage

```
                                   :ANY-PAGE {

GENERIC:
  REQ   Application-comments      {REQ #constraint-name {"4"},
                        PERM #external-data {ANY_VALUE}},
  REQ   Medium-Type             {REQ  #nominal-page-size
                              {$NominalPageSizes},
                        REQ  #side-of-sheet
                              {'verso'|'unspecified'}},
SPECIFIC:
  CASE  $DAC OF {
       $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(VersoPage)}
       $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(VersoPage)}
          }
  PERM  Application-comments      {REQ #constraint-name {"4"},
                        PERM #external-data {ANY_VALUE}},
  PERM  Medium-Type             {PERM #nominal-page-size
                              {$NominalPageSizes},
                        PERM #side-of-sheet
                              {'verso'|'unspecified'}}}
```

7.3.3.6 BasicBody

```
                                   :ANY-FRAME-FIXED {

GENERIC:
  PERM  Layout-path             {'270-degrees' -- page layout A --
                        |'0-degrees'   -- page layout B
                        |'180-degrees' -- page layouts
                                        C and D --},
  REQ   Application-comments      {REQ #constraint-name {"28"},
                        PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
       $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(BasicBody)}
       $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(BasicBody)}
          }
  REQ   Subordinates            {SUB_ID_OF(SpecificBlock)+},
  PERM  Application-comments      {REQ #constraint-name {"28"},
                        PERM #external-data {ANY_VALUE}}}
```

7.3.3.7 VariableCompositeBody

```
                                    :ANY-FRAME-FIXED {

GENERIC:
  CASE  $DAC OF {
       $PDA|$FPDA:
        REQ   Generator-for-subordinates
                          {$VariableCompositeBodyGFS},
        PERM  Layout-path      {'270-degrees' -- page layout A --
                          |'0-degrees'   -- page layout B
                          |'180-degrees' -- page layouts
                                       C and D --}
          }
  REQ   Application-comments      {REQ #constraint-name {"7"},
                           PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
       $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                  (VariableCompositeBody)},
       $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                  (VariableCompositeBody)}
          }
  REQ   Subordinates            {SUB_ID_OF(BasicFloat)+,
                           SUB_ID_OF(SnakingColumns)+,
                           SUB_ID_OF(SynchronizedColumns)+,
                           SUB_ID_OF(FootnoteArea)},
  PERM   Application-comments      {REQ #constraint-name {"7"},
                           PERM #external-data {ANY_VALUE}}}
```

7.3.3.8 BasicFloat

```
                                    :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
       $PDA|$FPDA:
       REQ   Position         {REQ #variable-position {
                            PERM #offset {ANY_VALUE},
                            PERM #separation {ANY_VALUE},
                            PERM #alignment {ANY_VALUE},
                         PERM #fill-order {'normal'}}},

     PERM  Permitted-categories  {ANY_STRING}

     CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

     '270-degrees': -- page layout A --
         REQ  Dimension     {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE},
                              |REQ #maximum-size {'applies'}},
                             REQ #vertical-dimension
                              {REQ #rule-b {ANY_VALUE}}},
         PERM Layout-path    {'270-degrees'}

     '0-degrees': -- page layout B --
         REQ  Dimension     {REQ #horizontal-dimension
                              {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
         REQ  Layout-path   {'0-degrees'}

     '180-degrees': -- page layouts C and D --
         REQ  Dimension     {REQ #horizontal-dimension
                              {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
         REQ  Layout-path   {'180-degrees'}
        }
        }
  REQ    Application-comments      {REQ #constraint-name {"12"},
                         PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
     $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(BasicFloat)},
     $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(BasicFloat)}
        }
  REQ    Subordinates           {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments      {REQ #constraint-name {"12"},
                         PERM #external-data {ANY_VALUE}}}
```

7.3.3.9 SynchronisedColumns

```
                              :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
      $PDA|$FPDA:
      REQ   Generator-for-subordinates
                   {$SynchronizedColumnsGFS},
      REQ   Position        {REQ #variable-position {
                    PERM #offset {ANY_VALUE},
                    PERM #separation {ANY_VALUE},
                    PERM #alignment {ANY_VALUE},
                    PERM #fill-order {'normal'}}
      CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

      '270-degrees': -- page layout A --
          REQ  Dimension   {REQ #horizontal-dimension
                    {REQ #fixed-dimension {ANY_VALUE}
                    |REQ #maximum-size {'applies'}},
                   REQ #vertical-dimension
                    {REQ #rule-b {ANY_VALUE}}},
          PERM Layout-path  {'270-degrees'}

      '0-degrees': -- page layout B --
          REQ  Dimension   {REQ #horizontal-dimension
                    {REQ #rule-b {ANY_VALUE}},
                   REQ #vertical-dimension
                    {REQ #fixed-dimension {ANY_VALUE}
                    |REQ #maximum-size {'applies'}}},
          REQ  Layout-path  {'0-degrees'}

      '180-degrees': -- page layouts C and D --
          REQ  Dimension   {REQ #horizontal-dimension
                    {REQ #rule-b {ANY_VALUE}},
                   REQ #vertical-dimension
                    {REQ #fixed-dimension {ANY_VALUE}
                    |REQ #maximum-size {'applies'}}},
          REQ  Layout-path  {'180-degrees'}
        }
        }
  REQ    Application-comments       {REQ #constraint-name {"11"},
                                PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
         $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                     (SynchronizedColumns)}
         $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                     (SynchronizedColumns)}              }
  REQ    Subordinates              {SUB_ID_OF(ColumnFixed)+},
  PERM   Application-comments       {REQ #constraint-name {"11"},
                                PERM #external-data {ANY_VALUE}}}
```

7.3.3.10 SnakingColumns

```
                                        :ANY-FRAME-VARIABLE {

GENERIC:
  CASE   $DAC OF {
         $PDA|$FPDA:
         REQ  Generator-for-subordinates  {$SnakingColumnsGFS},
         REQ  Position            {REQ #variable-position {
                              PERM #offset {ANY_VALUE},
                              PERM #separation {ANY_VALUE},
                              PERM #alignment {ANY_VALUE},
                              PERM #fill-order {'normal'}},
         PERM Balance             {ANY_VALUE}
         CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

         '270-degrees': -- page layout A --
              REQ  Dimension    {REQ #horizontal-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                             |REQ #maximum-size {'applies'}},
                            REQ #vertical-dimension
                             {REQ #rule-b {ANY_VALUE}}},
              REQ  Layout-path  {'0-degrees'|'180-degrees'}

         '0-degrees': -- page layout B --
              REQ  Dimension    {REQ #horizontal-dimension
                              {REQ #rule-b {ANY_VALUE}},
                            REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
```

```
                        |REQ #maximum-size {'applies'}}},
          PER  Layout-path  {'90-degrees'|'270-degrees'}

      '180-degrees': -- page layouts C and D --
          REQ  Dimension    {REQ #horizontal-dimension
                        {REQ #rule-b {ANY_VALUE}},
                       REQ #vertical-dimension
                        {REQ #fixed-dimension {ANY_VALUE}
                        |REQ #maximum-size {'applies'}}},
          PERM Layout-path  {'270-degrees'}

         }
         }
  REQ    Application-comments       {REQ #constraint-name {"10"},
                       PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
      $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF
                             (Snakingcolumns)}
      $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF
                             (Snakingcolumns)}
         }
  REQ    Subordinates            {SUB_ID_OF(ColumnVariable)+},
  PERM   Application-comments       {REQ #constraint-name {"10"},
                       PERM #external-data {ANY_VALUE}}}
```

7.3.3.11 ColumnVariable

```
                              :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
      $PDA|$FPDA:

       PERM  Permitted-categories  {ANY_STRING},

       REQ   Position          {REQ #variable-position {
                         PERM #offset {ANY_VALUE},
                         PERM #separation {ANY_VALUE},
                         PERM #alignment {ANY_VALUE},
                         PERM #fill-order {'normal'}}
```

```
          CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

          '270-degrees':  -- page layout A --
               REQ  Dimension     {REQ #horizontal-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}},
                                  REQ #vertical-dimension
                                   {REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}}},
               PERM Layout-path   {'270-degrees'}

          '0-degrees: -- page layout B --
               REQ  Dimension    {REQ #horizontal-dimension
                                   {REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}},
                                  REQ #vertical-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}}},
               REQ  Layout-path   {'0-degrees'}

          '180-degrees: -- page layouts C and D --
               REQ  Dimension    {REQ #horizontal-dimension
                                   {REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}},
                                  REQ #vertical-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}}},
               REQ  Layout-path   {'180-degrees'}
              }
   REQ    Application-comments       {REQ #constraint-name {"9"},
                                  PERM #external-data {ANY_VALUE}},
SPECIFIC:
   CASE   $DAC OF {
        $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(ColumnVariable)}
        $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(ColumnVariable)}
              }
   REQ    Subordinates             {SUB_ID_OF(SpecificBlock)+)},
   PERM   Application-comments       {REQ #constraint-name {"9"},
                                  PERM #external-data {ANY_VALUE}}}
```

7.3.3.12 ColumnFixed

```
                          :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
      $PDA|$FPDA:

        PERM  Permitted-categories  {ANY_STRING},

        REQ   Position          {REQ #fixed-position
                            {REQ #horizontal-position {ANY_VALUE},
                             REQ #vertical-position {ANY_VALUE}}}

        CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

        '270-degrees': -- page layout a --
            REQ  Dimension    {REQ #horizontal-dimension
                             {REQ #fixed-dimension {ANY_VALUE},
                             |REQ #maximum-size {'applies}},
                            REQ #vertical-dimension
                             {REQ #rule-b {ANY_VALUE}
                             |REQ #maximum-size {'applies'}}},
            PERM Layout-path  {'270-degrees'}

        '0-degrees': -- page layout B --
            REQ  Dimension    {REQ #horizontal-dimension
                             {REQ #rule-b {ANY_VALUE}
                             |REQ #maximum-size {'applies'}},
                            REQ #vertical-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                             |REQ #maximum-size {ANY_VALUE}}},
            REQ  Layout-path  {'0-degrees'}

        '180-degrees': -- page layouts C and D --
            REQ  Dimension    {REQ #horizontal-dimension
                             {REQ #maximum-size {'applies'}},
                            REQ #vertical-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                             |REQ #maximum-size {'applies'}},
            REQ  Layout-path  {'180-degrees'}
          }
          }
  REQ   Application-comments      {REQ #constraint-name {8"},
                           PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
      $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF
                                    (ColumnFixed)}
      $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF
                                    (ColumnFixed)}
          }
  REQ   Subordinates            {SUB_ID_OF(SpecificBlock)+},
  PERM  Application-comments      {REQ #constraint-name {"8"},
                           PERM #external-data {ANY_VALUE}}}
```

7.3.3.13 FootnoteArea

```
                                :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
      $PDA|$FPDA:

        REQ  Position           {REQ #variable-position {
                            PERM #offset {ANY_VALUE},
                            PERM #separation {ANY_VALUE},
                            PERM #alignment {ANY_VALUE},
                            REQ  #fill-order {'reverse'}},

        REQ  Permitted-categories {"Footnote"}

        CASE SUPERIOR (VariableCompositeBody(Layout-Path)) OF {

        '270-degrees': -- page layout A --
            REQ  Dimension   {REQ #horizontal-dimension
                           {REQ #fixed-dimension {ANY_VALUE}
                           |REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                           {REQ #rule-b {ANY_VALUE}}},
            PERM Layout-path  {'270-degrees'}

        '0-degrees': -- page layout B --
```

```
            REQ  Dimension   {REQ #horizontal-dimension
                              {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
            REQ  Layout-path  {'0-degrees'}

        '180-degrees': -- page layouts C and D --
            REQ  Dimension   {REQ #horizontal-dimension
                              {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
            REQ  Layout-path  {'180-degrees'}
           }
           }
  REQ   Application-comments      {REQ #constraint-name {"15"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(FootnoteArea)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(FootnoteArea)}
           }
  REQ   Subordinates             {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments      {REQ #constraint-name {"15"},
                             PERM #external-data {ANY_VALUE}}}
```

7.3.3.14 BasicHeader

```
                                     :ANY-FRAME-FIXED {

GENERIC:
  REQ   Logical-source           {OBJECT_CLASS_ID_OF(CommonContent)},
  PERM   Layout-path             {'270-degrees' -- page layouts A,B,C --
                             |'180-degrees' -- page layout D --},
  REQ   Application-comments      {REQ #constraint-name {"27"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(BasicHeader)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(BasicHeader)}
```

```
            }
  REQ    Subordinates             {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments       {REQ #constraint-name {"27"},
                        PERM #external-data {ANY_VALUE}}}
```

7.3.3.15 BasicFooter

```
                                  :ANY-FRAME-FIXED {

GENERIC:
  REQ    Logical-source           {OBJECT_CLASS_ID_OF(CommonContent)},
  PERM   Layout-path              {'270-degrees' -- page layouts A,B,C --
                        |'180-degrees' -- page layout D --},
  REQ    Application-comments       {REQ #constraint-name {"33"},
                        PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(BasicFooter)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(BasicFooter)}
            }
  REQ    Subordinates             {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments       {REQ #constraint-name {"33"},
                        PERM #external-data {ANY_VALUE}}}
```

7.3.3.16 CompositeHeader

```
                                  :ANY-FRAME-FIXED {

GENERIC:
  REQ    Generator-for-subordinates  {$HeaderFooterGFS},
  PERM   Layout-path              {'270-degrees' -- page layouts A,B,C --
                        |'180-degrees' -- page layout D --},
  REQ    Application-comments       {REQ #constraint-name {"5"},
                        PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(CompositeHeader)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(CompositeHeader)}
            }
  REQ    Subordinates             {SUB_ID_OF(SourcedContentFixed)+,
                        SUB_ID_OF(ArrangedContentFixed)+,
                        SUB_ID_OF(SourcedContentVariable)+,
                        SUB_ID_OF(ArrangedContentVariable)+},
  PERM  Application-comments     {REQ #constraint-name {"5"},
                        PERM #external-data {ANY_VALUE}}}
```

7.3.3.17 CompositeFooter

```
                              :ANY-FRAME-FIXED {

GENERIC:
  REQ  Generator-for-subordinates  {$HeaderFooterGFS},
  PERM  Layout-path             {'270-degrees' -- page layouts A,B,C --
                        |'180-degrees' -- page layout D --},
  REQ  Application-comments     {REQ #constraint-name {"32"},
                        PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE  $DAC OF {
      $FDA:  PERM  Object-class  {OBJECT_CLASS_ID_OF(CompositeFooter)}
      $FPDA: REQ   Object-class  {OBJECT_CLASS_ID_OF(CompositeFooter)}
          }
  REQ  Subordinates             {SUB_ID_OF(SourcedContentFixed)+,
                        SUB_ID_OF(ArrangedContentFixed)+,
                        SUB_ID_OF(SourcedContentVariable)+,
                        SUB_ID_OF(ArrangedContentVariable)+},
  PERM  Application-comments     {REQ #constraint-name {"32"},
                        PERM #external-data {ANY_VALUE}}}
```

7.3.3.18 SourcedContentVariable

```
                              :ANY-FRAME-VARIABLE {

GENERIC:
  CASE  $DAC OF {
      $PDA|$FPDA:

        REQ  Logical-source      {OBJECT_CLASS_ID_OF(CommonContent)},
        REQ  Position          {REQ #variable-position {
                         PERM #offset {ANY_VALUE},
                         PERM #separation {ANY_VALUE},
                         PERM #alignment {ANY_VALUE},
                         PERM #fill-order {'normal'}}
      CASE SUPERIOR (CompositeHeader|CompositeFooter
                            (Layout-path)) OF {
      '270-degrees':
          REQ  Dimension     {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #rule-b {ANY_VALUE}}},
          PERM Layout-path   {'270-degrees'}
      '180-degrees':
          REQ  Dimension     {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
          REQ  Layout-path   {'180-degrees'}
          }
          }
  REQ    Application-comments      {REQ #constraint-name {"19"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                   (SourcedContentVariable)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                   (SourcedContentVariable)}
          }
  REQ    Subordinates              {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments      {REQ #constraint-name {"19"},
                             PERM #external-data {ANY_VALUE}}}
```

7.3.3.19 ArrangedContentVariable

```
                                   :ANY-FRAME-VARIABLE {

GENERIC:
  CASE   $DAC OF {
      $PDA|$FPDA:
        REQ   Generator-for-subordinates
                    {SEQ({OBJECT_CLASS_IS_OF(GenericBlock)}+)},
        REQ   Position          {REQ #variable-position {
                            PERM #offset {ANY_VALUE},
                            PERM #separation {ANY_VALUE},
                            PERM #alignment {ANY_VALUE},
                            PERM #fill-order {'normal'}},
        REQ   Dimension         {REQ #horizontal-dimension
                           {REQ #fixed-dimension {ANY_VALUE},
                           REQ #vertical-dimension
                           {REQ #fixed-dimension {ANY_VALUE}},
        CASE SUPERIOR (CompositeHeader|CompositeFooter
                             (Layout-Path)) OF {
        '270-degrees': PERM   Layout-path    {'270-degrees'}
        '180-degrees': REQ    Layout-path    {'180-degrees'}
            }
            }
    REQ    Application-comments      {REQ #constraint-name {"17"},
                           PERM #external-data {ANY_VALUE}},    CASE   $DAC OF {
       $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                (ArrangedContentVariable)}
       $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                (ArrangedContentVariable)}
            }
    REQ    Subordinates              {SUB_ID_OF(GenericBlock)+},
    PERM   Application-comments      {REQ #constraint-name {"17"},
                           PERM #external-data {ANY_VALUE}}}
```

7.3.3.20 SourcedContentFixed

```
                              :ANY-FRAME-VARIABLE {

GENERIC:
  CASE   $DAC OF {
     $PDA|$FPDA:

        REQ    Logical-source    {OBJECT_CLASS_ID_OF(CommonContent)},
        REQ    Position          {REQ #fixed-position
                           {REQ #horizontal-position{ANY_VALUE},
                            REQ #vertical-position{ANY_VALUE}}}
        CASE SUPERIOR (CompositeHeader|Compositefooter
                          (Layout-path)) OF {
      '270-degrees':
          REQ  Dimension    {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #rule-b {ANY_VALUE}}},
          PERM Layout-path  {'270-degrees'}
      '180-degrees':
          REQ  Dimension    {REQ #horizontal-dimension
                              {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                              {REQ #fixed-dimension {ANY_VALUE}},
          REQ  Layout-path  {'180-degrees'}
         }
         }
  REQ    Application-comments      {REQ #constraint-name {"18"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
      $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                   (SourcedContentFixed)}
      $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                   (SourcedContentFixed)}
         }
  REQ    Subordinates            {SUB_ID_OF(SpecificBlock)+},
  PERM   Application-comments      {REQ #constraint-name {"18"},
                             PERM #external-data {ANY_VALUE}}}
```

7.3.3.21 ArrangedContentFixed

```
                                   :ANY-FRAME-FIXED {

GENERIC:
  REQ    Generator-for-subordinates  {SEQ({OBJECT_CLASS_IS_OF
                                   (GenericBlock)}+)},
  REQ    Application-comments      {REQ #constraint-name {"16"},
                             PERM #external-data {ANY_VALUE}},
SPECIFIC:
  CASE   $DAC OF {
        $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF
                                     (ArrangedContentFixed)}
        $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF
                                     (ArrangedContentFixed)}
           }
   REQ    Subordinates              {SUB_ID_OF(GenericBlock)+},
   PERM   Application-comments       {REQ #constraint-name {<"16"},
                              PERM #external-data {ANY_VALUE}}}
```

7.3.3.22 GenericBlock

```
                                   :BLOCK {

GENERIC:
   REQ    Object-type               {'block'},
   REQ    Content-architecture-class  {$FC|$FPC|$FPR|$FPG},
   PERM   Resource                  {ANY_VALUE},
   PERM   Content-portions            {CONTENT_ID_OF
                              (character-content-portion)+
                              |CONTENT__ID_OF
                              (raster-graphics-content-portion)
                              |CONTENT_ID_OF
                              (geometric-graphics-content-portion)};
   PERM   Presentation-style          {STYLE_ID_OF(P-Style1)
                              |STYLE_ID_OF(P-Style2)
                              |STYLE_ID_OF(P-Style3)},
   PERM   User-readable-comments     {ANY_STRING},
   PERM   User-visible-name           {ANY_STRING},
   PERM   Position                  {REQ #fixed-position
                               {REQ #horizontal-position{ANY_VALUE},
                                REQ #vertical-position{ANY_VALUE}}},
   PERM   Dimension                   {REQ #horizontal-dimension
                               {REQ #fixed-dimension {ANY_VALUE}},
                              REQ #vertical-dimension
                               {REQ #fixed-dimension {ANY_VALUE}}},
   REQ    Application-comments       {REQ #constraint-name {"29"},
                              PERM #external-data {ANY_VALUE}},
SPECIFIC:
   CASE  $DAC OF {
        $FDA:  PERM  Object-class   {OBJECT_CLASS_ID_OF(GenericBlock)}
        $FPDA: REQ   Object-class   {OBJECT_CLASS_ID_OF(GenericBlock)}
```

```
                }
   PERM   Application-comments       {REQ #constraint-name {"29"},
                                 PERM #external-data {ANY_VALUE}}}
```

7.3.3.23 SpecificBlock

```
                                          :BLOCK {

SPECIFIC:
   PERM   Application-comments       {REQ #constraint-name {"30"},
                                 PERM #external-data {ANY_VALUE}}}
```

7.4 Layout style constituent constraints

7.4.1 Macro definitions

```
DEFINE(LayoutObjectClasses,  "
                OBJECT_CLASS_ID_OF(PageSet)
                |OBJECT_CLASS_ID_OF(Page)
                |OBJECT_CLASS_ID_OF(RectoPage)
                |OBJECT_CLASS_ID_OF(VersoPage)
                |OBJECT_CLASS_ID_OF(VariableCompositeBody)
                |OBJECT_CLASS_ID_OF(BasicFloat)
                |OBJECT_CLASS_ID_OF(SnakingColumns)
                |OBJECT_CLASS_ID_OF(SynchronizedColumns)
                |OBJECT_CLASS_ID_OF(ColumnFixed)
                |OBJECT_CLASS_ID_OF(ColumnVariable)
                      ")


DEFINE(SameLayoutObject, "
        REQ #sameas{<object-id-expr> ::=  PREC-OBJ(CURR-OBJ);
                  |'null'},
        RERM #within{'page'}
                    ")
```

7.4.2 Factor constraints

7.4.2.1 FACTOR: ANY-LAYOUT-STYLE

```
                                    {

  REQ    Layout-style-identifier    {ANY_VALUE},
  PERM   User-visible-name          {ANY_STRING},
  PERM   User-readable-comments     {ANY_STRING}}
```

7.4.3 Layout style constituent constraints

7.4.3.1 L-Style1

```
                                    :ANY-LAYOUT-STYLE  {

     -- this style is used for the constituent Passage only --

  CASE Document-profile(Generic-layout-structure) OF {
     'complete-generator-set':
         PERM   Layout-object-class  {OBJECT_CLASS_ID_OF(PageSet)},
         PERM   New-layout-object    {OBJECT_CLASS_ID_OF(PageSet)},
         PERM   Indivisibility       {$LayoutObjectClasses
                              |ANY_STRING|'page'|'null'}
     VOID:
         PERM   Indivisibility       {ANY_STRING|'page'|'null'}
                                      }}
```

7.4.3.2 L-Style2

```
                                    :ANY-LAYOUT-STYLE {

     -- this style is used for the constituents NumberedSegment
       and Paragraph --

  CASE Document-profile(Generic-layout-structure) OF {
     'complete-generator-set':
         PERM   Indivisibility       {$LayoutObjectClasses
                              |ANY_STRING|'page'|'null'},
         PERM   Layout-object-class  {OBJECT_CLASS_ID_OF(PageSet)},
          PERM   New-layout-object    {$LayoutObjectClasses
                                  |ANY_STRING|'page'|'null'}
      VOID:
          PERM   Indivisibility       {ANY_STRING|'page'|'null'},
          PERM   New-layout-object    {ANY_STRING|'page'|'null'}
                                      }
  PERM   Same-layout-object           {$SameLayoutObject},
  PERM   Synchronization              {ANY_VALUE}}
```

7.4.3.3 L-Style3

```
                                      :ANY-LAYOUT-STYLE {

      -- this style is used for the constituents
         BodyText, Number and FootnoteReference --

  CASE Document-profile(Generic-layout-structure) OF {
      'complete-generator-set':
          PERM   Indivisibility       {$LayoutObjectClasses,
                                  |ANY_STRING|'page'|'null'},
          PERM   New-layout-object    {$LayoutObjectClasses
                                  |ANY_STRING|'page'|'null'}
      VOID:
          PERM   Indivisibility       {ANY_STRING|'page'|'null'},
          PERM   New-layout-object    {ANY_STRING|'page'|'null'}
                                      }
  PERM   Layout-category              {ANY_STRING},
  PERM   Same-layout-object           {$SameLayoutObject},
  PERM   Concatenation                {ANY_VALUE},
  PERM   Offset                       {ANY_VALUE},
  PERM   Separation                   {PERM #leading-edge{ANY_INTEGER},
                             PERM #trailing-edge{ANY_INTEGER}},
  PERM   Block-alignment              {ANY_VALUE},
  PERM   Synchronization              {ANY_VALUE}}
```

7.4.3.4 L-Style4

```
                                      :ANY-LAYOUT-STYLE {

      -- this style is used for the constituent Footnote only --
```

```
   PERM   Indivisibility           {'page'|'null'},
   PERM   Same-layout-object            {$SameLayoutObject}}
```

7.4.3.5 L-Style5

```
                                     :ANY-LAYOUT-STYLE  {

     -- this style is used for the constituents
       BodyRaster and BodyGeometric --

   CASE Document-profile(Generic-layout-structure) OF {
      'complete-generator-set':
          PERM   New-layout-object   {$LayoutObjectClasses,
                                  |ANY_STRING|'page'|'null'}
      VOID:
          PERM   New-layout-object   {ANY_STRING|'page'|'null'}
                                       }
   PERM   Layout-category            {ANY_STRING},
   PERM   Offset                 {ANY_VALUE},
   PERM   Same-layout-object          {$SameLayoutObject},
   PERM   Separation               {PERM #leading-edge{ANY_INTEGER},
                             PERM #trailing-edge{ANY_INTEGER}},
   PERM   Block-alignment            {ANY_VALUE},
   PERM   Synchronization            {ANY_VALUE}}
```

7.4.3.6 L-Style6

```
                                     :ANY-LAYOUT-STYLE  {

     -- this style is used for the constituent FootnoteText --

   REQ   Layout-category            {"Footnote"},
   PERM   Concatenation              {ANY_VALUE},
   PERM   Offset                 {ANY_VALUE},
   PERM   Block-alignment            {ANY_VALUE},
   PERM   Separation               {PERM #leading-edge{ANY_INTEGER},
                             PERM #trailing-edge{ANY_INTEGER}}}
```

7.4.3.7 L-Style7

```
                                  :ANY-LAYOUT-STYLE  {

    -- this style is used for the constituents
       CommonText and PageNumber --

 PERM   Concatenation            {ANY_VALUE},
 PERM   Offset                {ANY_VALUE},
 PERM   Block-alignment          {ANY_VALUE},
 PERM   Separation              {PERM #leading-edge{ANY_INTEGER},
                          PERM #trailing-edge{ANY_INTEGER}}}
```

7.4.3.8 L-Style8

```
                                  :ANY-LAYOUT-STYLE  {

    -- this style is used for the constituents
       CommonRaster and CommonGeometric --

 PERM   Offset                {ANY_VALUE},
 PERM   Block-alignment          {ANY_VALUE},
 PERM   Separation              {PERM #leading-edge{ANY_INTEGER},
                          PERM #trailing-edge{ANY_INTEGER}}}
```

7.4.3.9 L-Style9

```
                                  :ANY-LAYOUT-STYLE  {

    -- this style is used for the constituent FootnoteNumber --

 REQ   Layout-category           {"Footnote"},
 PERM   Offset                {ANY_VALUE},
 PERM   Block-alignment          {ANY_VALUE},
 PERM   Separation              {PERM #leading-edge{ANY_INTEGER},
                          PERM #trailing-edge{ANY_INTEGER}}}
```

7.5 Presentation style constraints

7.5.1 Macro definitions

No macro definitions are applicable to this clause.

7.5.2 Factor constraints

7.5.2.1 FACTOR ANY-PRESENTATION-STYLE

```
                                        {

  REQ    Presentation-style-identifier {ANY_VALUE},
  PERM   User-visible-name            {ANY_STRING},
  PERM   User-readable-style          {ANY_STRING}}
```

7.5.3 Presentation style constituent constraints

7.5.3.1 P-Style1

```
                                        :ANY-PRESENTATION-STYLE  {

 -- this style is used for the constituents BodyText, Number,
    FootnoteNumber, FootnoteReference and FootnoteText --

  PERM  Presentation attributes  {
    PERM #character-attributes   {
      PERM #alignment                  {ANY_VALUE},
      PERM #character-spacing            {ANY_VALUE},
      PERM #character-fonts            {ANY_VALUE},
      PERM #character-orientation      {'0 degrees'
                                  |'90-degrees'},
      PERM #character-path             {'0-degrees'
                                  |'90-degrees'
                                  |'180-degrees'
                                  |'270-degrees'},
      PERM #code-extension-announcers  {$CDEXTAN},
      PERM #first-line-offset         {ANY_VALUE},
      PERM #graphic-character-sets     {$PERMIT-GRCHAR},
      PERM #graphic-character-subrepertoire  {ANY_VALUE},
      PERM #graphic-rendition          {$GRAPHICRENDITIONS},
      PERM #indentation             {ANY_VALUE},
      PERM #itemisation             {ANY_VALUE},
      PERM #kerning-offset           {ANY_VALUE},
      PERM #line-progression          {'90-degrees'
                          |'270-degrees'},
      PERM #line-spacing             {ANY_VALUE},
      PERM #line-layout-table         {ANY_VALUE},
      PERM #orphan-size              {ANY_VALUE},
      PERM #proportional-line-spacing  {ANY_VALUE},
      PERM #widow-size               {ANY_VALUE}}}
```

7.5.3.2 P-Style2

```
                             :ANY-PRESENTATION-STYLE  {

 -- this style is used for the constituents BodyGeometric and CommonGeometric --

   PERM  Presentation attributes  {
     PERM #geometric-graphics-attributes {
       PERM #picture-dimensions        {ANY_VALUE},
       PERM #picture-orientation       {ANY_VALUE},
       PERM #text-rendition           {PERM #fonts-list{ANY_VALUE},
                          PERM #character-set-list
                                {ANY_VALUE}}
```

7.5.3.3 P-Style3

```
                             :ANY-PRESENTATION-STYLE  {

 -- this style is used for the constituents BodyRaster and CommonRaster --

   PERM  Presentation attributes  {
     PERM #raster-graphics-attributes   {
       PERM #image-dimensions           {ANY_VALUE},
       PERM #clipping                {ANY_VALUE},
       PERM #pel-spacing               {{REQ #length {ANY_VALUE},
                            REQ #pel-spaces{ANY_VALUE}}
                           |'null'},
       PERM #spacing-ratio             {ANY_VALUE}}}
```

7.5.3.4 P-Style4

```
                                        :ANY-PRESENTATION-STYLE {

 -- this style is used for the constituents CommonText and PageNumber --

   PERM  Presentation attributes  {
     PERM #character-attributes   {
       PERM #alignment                  {ANY_VALUE},
       PERM #character-spacing          {ANY_VALUE},
       PERM #character-fonts            {ANY_VALUE},
       PERM #character-orientation      {'0-degrees'
                                 |'90-degrees'},
       PERM #character-path             {'0-degrees'
                                 |'180-degrees'
                                 |'270-degrees'},
       PERM #code-extension-announcers   {$CDEXTAN},
       PERM #first-line-offset          {ANY_VALUE},
       PERM #graphic-character-sets      {$PERMIT-GRCHAR},
       PERM #graphic-character-subrepertoire  {ANY_VALUE},
       PERM #graphic-rendition          {$GRAPHICRENDITIONS},
       PERM #indentation                {ANY_VALUE},
       PERM #itemisation                {ANY_VALUE},
       PERM #kerning-offset             {ANY_VALUE},
       PERM #line-progression           {'90-degrees'
                                 |'270-degrees'},
       PERM #line-spacing               {ANY_VALUE},
       PERM #line-layout-table          {ANY_VALUE},
       PERM #proportional-line-spacing   {ANY_VALUE}}}
```

7.6 Content portion constraints

7.6.1 Macro definitions

No macro definitions are applicable to this clause.

7.6.2 Factor constraints

7.6.2.1 FACTOR ANY-CONTENT

```
                                  :ANY-CONTENT {

    PERM   Content-identifier-logical   {ANY_VALUE},
    PERM   Content-idenifier-layout     {ANY_VALUE}}
```

7.6.3 Content portion constraints

7.6.3.1 Character content portion

```
                                  :ANY-CONTENT {

  PERM   Type-of-coding              {ASN.1{2 8 3 6 0}},
  PERM   Alternative-representation   {ANY_STRING},
  PERM   Content-information
      {CHARACTER, {#STAB  {ANY_VALUE}
                 |#SHS   {0,1,2,3,4}
                 |#SGR   {$GRAPHICRENDITIONS}
                 |#SVS   {0 1 2 4}
                 |#SLS   {ANY_VALUE}
                 |#SCS   {ANY_VALUE}
                 |#SRS   {ANY_VALUE}
                 |#HPR   {ANY_VALUE}
                 |#HPB   {ANY_VALUE}
                 |#VPR   {ANY_VALUE}
                 |#VPB   {ANY_VALUE}
                 |#JFY   {0}
                 |#CR
                 |#LF
                 |#PLD
                 |#PLU
                 |#SP
                 |#SUB
                 |#BPH
                 |#NBH
                 |#SOS
                 |#ST
```

```
                  |#$LS0
                  |#$LS1R
                  |#$LS2R
                  |#$LS3R
                  |#$SS2
                  |#$SS3
                  |#$DEG-CORE-G0
                  |#$DEG-646-G0
                  |#$DEG-ANY-G1
                  |#$DEG-ANY-G2
                  |#$DEG-ANY-G3
                  |#$DEG-EMPTY-G1
                  }...}}
```

7.6.3.2 Raster graphics content portion

```
                              :ANY-CONTENT {

  PERM  Number-of-lines              {>0},
  REQ   Number-of-pels-per-line      {>=0},
  PERM  Type-of-coding          {ASN.1{2 8 3 7 0}  -- T.6 encoding --
                      |ASN.1{2 8 3 7 1} -- T.4 one-dimensional
                                    encoding --
                      |ASN.1{2 8 3 7 2} -- T.4 two dimensional
                                    encoding --
                      |ASN.1{2 8 3 7 3} -- bitmap encoding --},
  PERM  Compression                  {ANY_VALUE},
  PERM  Alternative-representation   {ANY_STRING},
  PERM  Content-information          {RASTER}}
```

7.6.3.3 Geometric graphics content portion

```
                              :ANY-CONTENT {

  PERM  Type-of-coding               {ASN.1{2 8 3 8 0}},
  PERM  Alternative-representation   {ANY_VALUE},
  PERM  Content-information          {GEOMETRIC}}
```

8 Interchange format

Two interchange formats are supported by this profile. The Interchange Format Class A can be used by applications requiring a binary encoding based on ASN.1. The Interchange Format SDIF can be used by applications requiring a SGML based clear text encoding. This latter interchange format is an SGML application, called Office Document Language (ODL). For the purposes of interchange the ODL ENTITIES are placed in an ASN.1 wrapper, as defined by SDIF. Each encoding form has inherent advantages. Conversion of document encoded in one interchange format into the other should not produce the loss of semantic document information.

8.1 Interchange format class A

8.1.1 Interchange format

The value of the document profile attribute "Interchange format" for this interchange format is "if-a". This form of ODIF is defined in ISO 8613-5.

8.1.2 DAP identifier

The value for the document profile attribute "Document application profile" for this interchange format is represented by the following object identifier.

Editor's Note: To be supplied

8.1.3 Encoding of application comments

The encoding of the attribute "Application comments" is defined in this encoding as an octet string as specified in ISO 8613-5. This document application profile requires that the encoding within that octet string be in accordance with the ASN.1 syntax specified in the following module definition:

```
FOD_DAPSpecification
DEFINITION              ::= BEGIN
EXPORTS   Object-Class-Appl-Comm-Encoding,
        Object-Appl-Comm-Encoding;
```

```
Object-Class-Appl-Comm-Encoding      ::= SEQUENCE {
    Constraint-name        [0] IMPLICIT PrintableString,
    External-data          [1] EXTERNAL OPTIONAL }

Object-Appl-Comm-Encoding          ::= SEQUENCE {
    Constraint-name        [0] IMPLICIT PrintableString OPTIONAL,
    External-data          [1] IMPLICIT OCTETSTRING OPTIONAL }

END
```

8.1.4 Octet string lengths

The maximum length of primitive octet string in data streams which may be encoded in accordance with this DAP is 32767 octets. If it is required to encode an octet string of greater length than this, constructed type encoding must be used.

8.2 Interchange format SDIF

8.2.1 Interchange format

The document profile attribute "Interchange format" does not apply for this interchange format. This form of ODIF is defined in Annex E of ISO 8613-5. In addition, ISO 8613-6, -7 and -8 contain addition specifications for this form of ODIF.

8.2.2 DAP identifier

The value for the attribute "Document application profile" for this interchange format is represented by the following object identifier.

Editor's Note: To be supplied

8.2.3 Encoding of application comments

The encoding of the attribute "Application comments" is defined in data stream conforming to this profile with this encoding with the following DTD definition:

```
<!DOCTYPE odaac [
<!--
<!DOCTYPE doc PUBLIC "-//USA-OIW//SGML ENCODED ODA APPLICATION
COMMENTS//EN"> -->

<!ELEMENT objcappc      - O     (consname, extndata?)>
                <!-- Object class application comment -->
<!ELEMENT objdappc      - O     (consname?, extndata?)>
                <!-- Object application comment -->
<!ELEMENT consname      - O     (#PCDATA)>
                <!-- Constraint name identifier -->
<!ELEMENT othrcomm      - O     (#RCDATA)>
                <!-- Other application comments -->
<!ATTLIST extndata      loc     ENTITY      #CONREF>
                <!-- Location of other application comments -->
]>
```

Annex A (Normative) Addenda and errata

A.1 Addenda

Editor's Note: The references to the applicable addenda to be supplied.

A.2 Errata

A.2.1 Base standard errata

Editor's Note: The references to the applicable errata to be supplied.

A.2.2 Agreements errata

There is not erratum to this agreement.

Annex B (Informative) Recommendations

B.1 Conveyance of ODA over CCITT X.400-1984

This recommendation describes how ODA body parts are to be encoded for transmission over a CCITT X.400-1984 service.

An ODA body part is encoded as OdaBodyPart in the definition given below:

```
OdaBodyPart ::= SEQUENCE { OdaBodyPartParameters, OdaData }
OdaBodyPartParameters ::= SET {
        document-application-profile
                [0] IMPLICIT OBJECT IDENTIFER,
        document-architecture-class
                [1] IMPLICIT INTEGER {
                        formatted (0),
                        processable (1),
                        formatted-processable (2) }
OdaData ::=  SEQUENCE OF Interchange-Data-Element
```

Note: It is recommended to transfer an ODA document as a single body part with tag 12:

```
Oda [12] IMPLICIT OCTETSTRING
```

The content of the octet string is encoded as OdaBodyPart, defined above. However, this is out of the scope of this profile.

B.2 Conveyance of ODA over FTAM

This recommendation describes the FTAM Document Type to be used for minimal storage and transfer capabilities of ODA data streams. It is recognized that enhanced capabilities may at some point be added.

When using FTAM to transfer an ODA file, the FTAM-3, "ISO FTAM Unstructured Binary", document type should be specified.

However, since files that do not contain ODA data streams can have the same document type, it is left up to the user of application programs that remotely access files using FTAM to know that a given file contains an ODA data stream.

B.3 Conveyance of ODA over flexible diskette media

The recommended method for interchanging ODA documents between systems by the exchange of magnetically recorded Flexible Disk Cartridges is by the use of ISO 10033, Recording of Documents Conforming to ISO 8613 on Flexible Disk Cartridges Conforming to ISO 9293.

Note: It is anticipated that this standard will be encorporated as an annex to ISO 8613-1).

This international standard provides for recording each ODA document as a separate file as defined by ISO 9293, Volume and File Structure of Flexible Disk Cartridges for Information Interchange.

Note: Document encoded in ODL can be stored such that each SGML ENTITY is recorded in a separate file or in the case of an SDIF encoding, the file can be stored in a single file

B.4 Font reference

~~This recommendation applies to document application profiles that support fonts.~~ The recommended method for specifying a font reference is to be based on ISO 9541. Such a reference is to be specified by the following ASN.1 encoding:

```
Fonts-Reference     ::=     SET {

user-visible-name           [0] IMPLICIT Comment-String OPTIONAL,
user-readable-comment       [1] IMPLICIT Comment-String OPTIONAL,
reference-attributes        [2] IMPLICIT SEQUENCE OF SET {
        precedence-number           [0] IMPLICIT INTEGER OPTIONAL,
        attributes                  [1] IMPLICIT Font-Attribute-Set,
        user-readable-comment     [2] IMPLICIT Comment-String OPTIONAL }
}
```

Font sizes from 6 to 72 points (100 to 1200 BMU) are intended to be supported by implementation conforming to this informative recommendation. All other values of font sizes may additionally be supported, but implementations may also support using some form of "fallback". ~~The following notation specifies the "basic" font sizes as SMU measures.~~

~~DEFINE(BasicFontSize," {~~

~~100 117 133 150 167 183~~
~~200 217 233 250 267 283~~
~~300 317 333 350 367 383~~
~~400 417 433 450 467 483~~
~~500 517 533 550 567 583~~
~~600 617 633 650 667 683~~
~~700 717 733 750 767 783~~
~~800 817 833 850 867 883~~
~~900 917 933 950 967 983~~
~~1000 1017 1033 1050 1067 1083~~
~~1100 1117 1133 1150 1167 1183 1200}~~

~~")~~

~~DEFINE(NonBasicFontSize," {ANY_VALUE}~~
~~")~~

~~It is recommended that the~~ The minimum font properties and values from ISO 9541 that are to be specified in a Font-Attribute-Set be those specified by the following document application profile notation.

```
Font-Attribute-Set    {

PERM      Fontname              {ANY_VALUE},
PERM      Standardversion       {-- To be supplied --},
PERM      Dsnsource             {ANY_VALUE},
PERM      Fontfamily            {ANY_VALUE},
PERM      Posture               {'upright' | 'italic-forward'},
PERM      Weight                {'light' | 'medium' | 'bold'},
PERM      Propwidth             {ANY_VALUE},
PERM      Glyphcomp             {
        PERM #inclgyphcols            {ANY_VALUE},
        PERM #exclgyphcols            {ANY_VALUE},
        PERM #inclgyphs               {ANY_VALUE},
        PERM #exclgyphs               {ANY_VALUE} },
PERM         Dsnsize                  {ANY_VALUE},
PERM         Minsize                  {
        PERM #numerator               {100 .. 1200},
        PERM #denominator             {1} },
PERM         Maxsize                  {
        PERM #numerator               {100 .. 1200},
        PERM #denominator             {1} },
        -- BMU Units equivalent to range of 6..72 point sizes --
PERM         Dsngroup                 {
        PERM #group-code                       {ANY_VALUE},
        PERM #subgroup-code           {ANY_VALUE},
        PERM #specific-group-code     {ANY_VALUE} },
PERM         Structure                {ANY_VALUE},
PERM         Wrmodes                  {
        PERM #wrmodename              {ANY_VALUE},
        PERM #nomescdir               {'0-degrees' | '90-degrees' | '180-degrees' |
                                      '270-degrees'},
        PERM #esclass                 {ANY_VALUE},
        PERM #avgescx                 {ANY_VALUE},
        PERM #avgescy                 {ANY_VALUE} }
}
```

B.5 ISO 8632 (CGM) constraints for this DAP

It is recommended that geometric graphics content information contain only those elements listed in this portion of the document, in addition to the constraints imposed by ISO 8613-8. It is believed that this subset of the CGM is sufficiently implemented to enable interworking of geometric graphics for application conforming this document application profile.

Where an element has parameters, recommended constraints on the values are given. The "--" symbol indicates that there is no recommended constraint.

Requirements in ISO 8632 and ISO 8613-8 concerning mandatory elements, parameters must be fulfilled.

No requirements are placed on how an Interpreter may optionally support features not supported by this profile.

B.5.1 Delimiter elements

Begin Metafile	Metafile name recommended to be the same as file name. Support for string length upto 255.
End Metafile	–
Begin Picture	Support for string length upto 255.
Begin Picture Body	
End Picture	

B.5.2 Metafile descriptor elements

Metafile Version	Must always be 1.
Metafile Description	Support for string length upto 255. String should begin with *ISO FOD26* to identify conformance to this profile.
~~VDC Type~~	~~integer~~
~~Integer Precision~~	~~16 bit~~
Real Precision	32 bit floating point (0,9,23) or 32 bit fixed point (1,16,16).
~~Index Precision~~	~~16~~
Colour Precision	8 or 16. A MDR is required for a default other than 8.
Colour Index Precision	8 or 16. A MDR is required for a default other than 8.
Maximum Colour Index	0 .. 255
Colour Value Extent	A 3-tuple in the range [0,32767]
Metafile Element List	A suitable short-hand or a list of each element supported by this profile shall be included in the element generated.
Font List	--
Character Set list	Any registered ISO character set. At a minimum, support should be provided for ISO 6937/2 (0, 4/0) or ISO 8859/1 (0, 4/2).

B.5.3 Picture descriptor elements

~~Scaling Mode~~	~~abstract~~
~~Colour Selection Mode~~	~~indexed~~
~~Line Width Specification Mode~~	~~scaled or absolute~~
~~Marker Size Specification Mode~~	~~scaled or absolute~~
~~Edge Width Specification Mode~~	~~scaled or absolute~~
VDC Extent	Two points with (x,y) in the range [-32767,32767]
Background Colour	A 3-tuple in the range [0,32767]

B.5.4 Control elements

~~VDC Integer Precision~~	~~16~~
Transparency	--
Clip Rectangle	Any value within the VDC Range.
Clip Indicator	--

B.5.5 Graphical primitive elements

Polyline	Support for point lists with upto 255 vertices.
~~Disjoint Polyline~~	
Polymarker	Support for point lists with upto 255 vertices.
Text	Support for string length upto 255. Only the graphical characters must be supported. No requirements are placed on how an interpreter may support control characters in the string parameter.
Polygon	Support for point lists with upto 255 vertices.
Rectangle	--
Circle	--
Circular arc centre	--
Circular arc centre close	--~~(close type = pie or chord)~~
Ellipse	--
Elliptical arc	--
Elliptical arc close	--~~(close type = pie or chord)~~

B.5.6 Attribute elements

Line Type	1-5
Line Width	--
Line Colour	--
Marker Type	1-5
Marker Size	--
Marker Colour	--
Text Font Index	--
~~Text Precision~~	~~'stroke' or 'char' or 'string'~~
~~Character Expansion Factor~~	~~--~~
~~Character Spacing~~	~~--~~
Text Colour	--
Character Height	--
Character Orientation	--
Text Alignment	Horizontal: 0, 1, 2, 3; Vertical: 0, 1, 2, 3, 4, 5.
~~horizontal$D: left, centre, right,~~	~~vertical$D: base, top~~
Character Set Index	1,2
Interior Style	0,1,3,4
Fill Colour	--
Hatch Index	1-6
~~Edge Width~~	~~--~~
~~Edge Type~~	~~--~~
~~Edge Colour~~	~~--~~
~~Edge Visibility~~	~~on/off~~
Colour Table	Minimum colour table support for 64 entries.–

B.5.7 External elements

Message	The presentation of the message string may not be appropriate for all applications. No requirement for the formatted presentation of the message string has been placed on the Interpreter. Only the *No Action* action flag need be supported. Support for string length upto 255.

Table of Contents

List of Tables

List of Figures

18 Network Management

1 Introduction

(Refer to the Stable Implementation Agreements Document.)

1.1 References

The following documents are referenced in the statements of the agreements relating to NIST/OSI network management.

OSI Systems Management References:

[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 Processing Systems - Open Systems Interconnection - Systems Management - Part 10: Accounting Meter Function, ISO/IEC JTC1/SC21 N4958, June 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 Processing Systems - Open Systems Interconnection - Systems Management - Part 4: Alarm Reporting Function, ISO/IEC JTC1/SC21 N4858, June 1990.

[ARR] ISO/IEC DIS 10164-3, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 3: Attributes for Representing Relationships, ISO/IEC JTC1/SC21 N4857, June 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 9596-2, Information Processing Systems - Open Systems Interconnection - Management Information Protocol Specification - Part 2: Common Management Information Protocol, 6 December l989.

[CMIS] ISO/IEC 9595-2, Information Processing Systems - Open Systems Interconnection - Management Information Service Definition - Part 2: Common Management Information Service, 6 December 1989.

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

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

[ERF] ISO/IEC DIS 10164-5, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 5: Event Report 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 Processing Systems - Open Systems Interconnection - SMI - Part 4: Guidelines for the Definition of Managed Objects, ISO/IEC JTC1/SC21 N4852, 15 June 1990.

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

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

[MSF] ISO/IEC CD 10164-m, Information Processing Systems - Open Systems Interconnection - Systems Management - Part m: Measurement Summarization Function (Second Working Draft), ISO/IEC JTC1/SC21 N4972, July 2, 1990.

[OAAC] ISO/IEC CD 10164-9, Information Processing Systems - 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 Processing Systems - Open Systems Interconnection - Systems Management - Part 1: Object Management Function, ISO/IEC JTC1/SC21, June 1990.

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

[SARF] ISO/IEC DP 10164-7, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 7: Security Alarm Reporting Function, ISO/IEC JTC1/SC21 N6064, 20 November 1989.

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

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

[SMO] ISO/IEC DIS 10040, Information Processing Systems - 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.

[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 Processing Systems - Open Systems Interconnection - Systems Management - Part 11: Workload Monitoring Function, ISO/IEC JTC1/SC21 N4959, June 28, 1990.

Other OSI References:

[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.

[ASN1] ISO 8824, Information Processing Systems - Open System Interconnection - Specification of Abstract Syntax Notation One (ASN.1), 19 May 1987.

[BER] ISO 8825, Information Processing Systems - Open Systems Interconnection - Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), 19 May 1987.

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

[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.

[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.

[SD35] EWOS/EG/NM/90/xx, Information Technology - Profiles AOMnn OSI Management - Management Communications Protocols - Part x: AOM12 - Full CMIP for Managing & Managed Systems, 7 September 1990.

2 Scope and Field of Application

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

2.1 Outline of Part 18

The following is an outline of the information provided in these agreements (Part 18):

Clause 1 -- INTRODUCTION: This clause provides introductory text and references.

Clause 2 -- SCOPE AND FIELD OF APPLICATION (This clause): This clause describes the purpose and structure of these agreements and outlines the phased approach required because these agreements are based on a large set of evolving OSI Systems Management Standards.

Clause 3 -- STATUS: This clause describes the current status of these agreements.

Clause 4 -- ERRATA: Once this document is incorporated into a version of the Stable Implementation Agreements for Open System Interconnection Protocols, this clause will contain corrections to the stable management agreements. In addition, this clause documents interim resolutions to defects found in the management standards.

Clause 5 -- MANAGEMENT FUNCTIONS: This clause documents agreements on Systems Management Functions and on the use of other application service elements (e.g., CMISE).

Clause 6 -- MANAGEMENT COMMUNICATIONS: This clause identifies agreements on association policies, on CMIS and CMIP, and on the upper layer services required by CMIP.

Clause 7 -- MANAGEMENT INFORMATION: This clause provides agreements for management information concepts and modeling techniaues based on [MIM] and [GDMO]. Subclauses introduce the information model, list principles for naming managed objects and attributes, and provide guidelines for defining management information.

Although managed object definitions are outside the scope of this clause, a Management Information Library (MIL) is attached as an informative Annex to these agreements.

Clause 8 -- IMPLEMENTATION PROFILES/CONFORMANCE CLASSES: This clause describes the implementation profiles/conformance classes that are used to categorize management products. For each of the classes identified, this clause outlines the criteria used to determine whether a given product conforms to the class specification.

Annex A -- Management Information Library: This clause is an informative annex that provides definitions of management information - managed object classes, name bindings, attributes, actions and notifications. The intention is to progress these definitions to an International Management Information Library.

2.2 Phased Approach

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

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

Editor's Note: [It has not been decided whether the Phased IA's will be published as parts of a single clause or as separate clauses.]

Editor's Note: [There is some confusion about whether the concept of phasing merely serves to organize the work program or whether it also defines packages of functionality to which vendors will build and claim conformance or which procurers will specify. This issue needs to be resolved and the text of this clause and the conformance clause must be aligned accordingly.]

2.2.1 Alignment With Evolving Standards

In some cases, these implementation agreements are based on DIS standards. As the relevant standards progress from DIS to IS, the agreements will be aligned.

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

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

2.2.2 Definition of Phase 1

As a first step in this phased approach, the NMSIG has targeted that the initial, Phase 1, agreements will be completed and progressed to Stable status by December, 1990. These Phase 1 agreements provide limited interoperable management in a heterogeneous vendor environment. They are the beginning of a comprehensive set of implementation agreements based on the emerging OSI Systems Management standards. Furthermore, these initial agreements allow the community to gain experience with OSI management standards as they emerge.

The focus of the Phase 1 agreements is to enable a managing process provided by one vendor to interoperate with an agent process provided by a different vendor to perform limited management on a set of managed objects. Specifically, these agreements focus on the managing process/agent process interface and the techniques used to define managed objects.

The scope of Phase 1 implementation agreements is the following:

Management Functions:

Object Management Function [OMF],

State Management Function [SMF],
Attributes For Representing Relationships [ARR],
Alarm Reporting Function [ARF],
Event Report Function [ERF].

Management Information:

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

Management Communication:

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

Management Objects:

Support Objects required for the above, and other Managed Object Definitions under development by the OSI MIB Working Group

Editor's Note: [The relation of the MIL definitions to Phase 1 IA's needs to be clarified. This question is the subject of an upcoming teleconference.]

Conformance Criteria:

Conformance Classes and Conformance Criteria for the above functionality.

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

1. No agreements are provided regarding management domains.

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

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

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

2.2.3 Future Phases

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

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

3 Status

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

1 INTRODUCTION

1.1 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 are planned to be added to Part 18 of the Stable Agreements in December 1990:

2 SCOPE AND FIELD OF APPLICATION

2.1 Outline of Part 18

2.2 Phased Approach

2.2.1 Definition of Phase 1

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

6.5 Services Required by CMIP

7 MANAGEMENT INFORMATION

7.1 The Information Model

7.2 Principles of Naming

7.3 Guidelines for the Definition of Management Information

The following clauses are planned to be added to Part 18 of the Stable Agreements in March 1991:

5 MANAGEMENT FUNCTIONS AND SERVICES

5.7 Log Control Function

8 CONFORMANCE

4 Errata

(None as yet)

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 function [ERF] 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 [ERF], 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 Measurement Summarization [MSF].

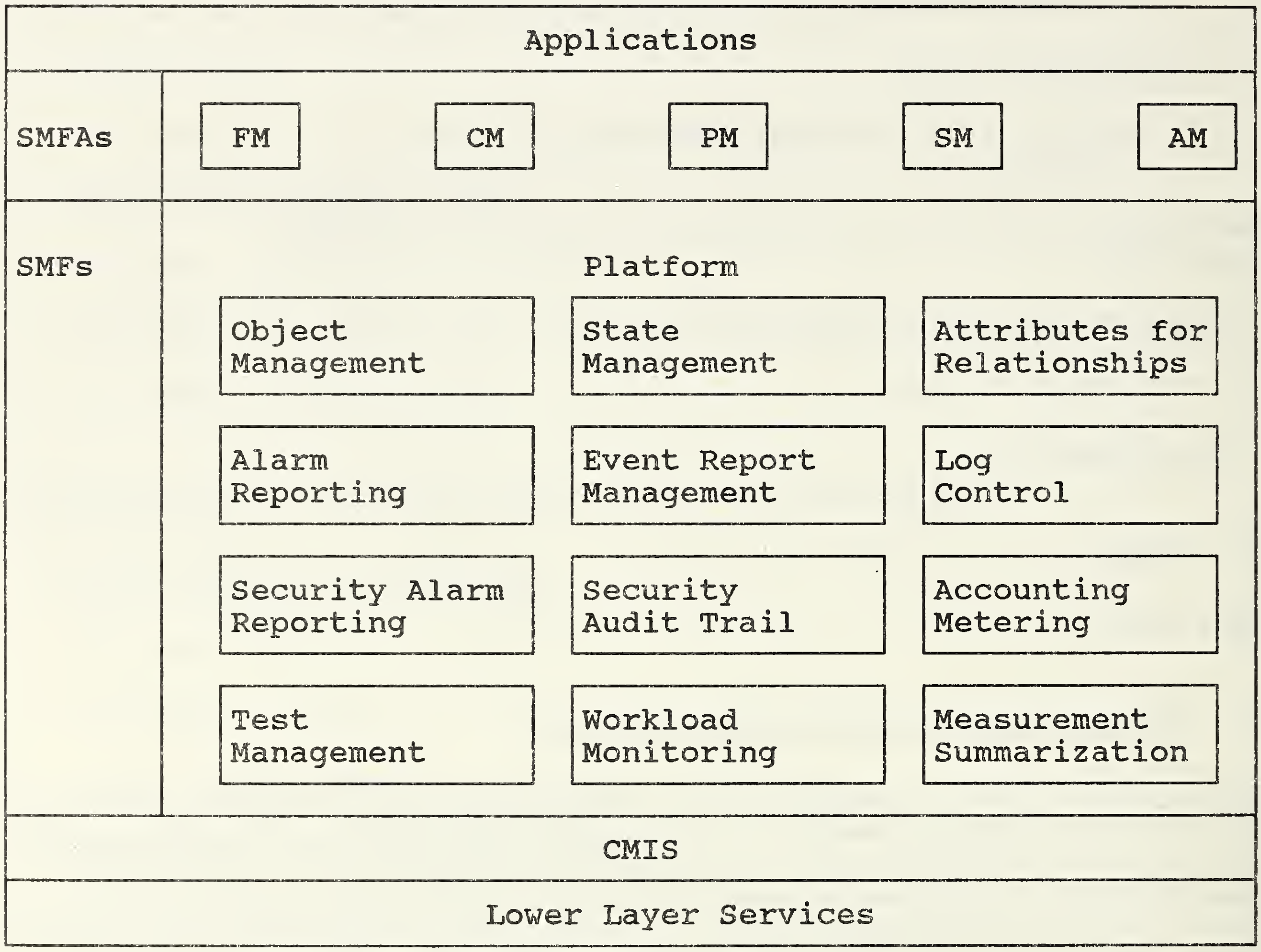


Figure 1: Functional Hierarchy of SMFs and SMFAs.

5.1 General Agreements

5.1.1 Conventions Used In SMF Agreements

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

1. For each System Management Function, a table lists the SMF services encompassed by the function, whether each SMF service is currently within the scope of these agreements, and related management support objects (if any). Although a management support object may be **related** to a SMF service, it may or may not be **required** to provide the SMF service.

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

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

M	Mandatory
O	Optional
C(p)	If Condition p exists, then parameter is mandatory; otherwise, the parameter is not applicable.
X	Excluded
I	Out Of Scope
-	Not Applicable
(=)	The value of the parameter is identical to the corresponding parameter in the interaction described by the preceding related service primitive.
U	The use of the parameter is a service-user option.

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

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

Editor's Note: [Need to improve definition of "Out Of Scope".]

5.1.2 General Agreements Referenced By Many SMF Services

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

5.1.2.1 Minimal Filter Complexity

If an implementation supports Multiple Object Selection, then it shall minimally support AND and OR with a set of two filter conditions (which shall not themselves be AND or OR), and NOT. In addition, the implementation shall support the filter conditions Equality, GreaterOrEqual, LessOrEqual, and Present. This means that a conforming implementation is not required to support compounds (AND or OR) with more than two items, and is not required to support the Substring filter condition. Additional filter items and conditions are beyond the scope of these agreements.

5.1.2.2 Mode Parameter Usage

All SMF Services mapped to CMIS M-EVENT-REPORT, M-ACTION, and M-SET shall allow either confirmed and unconfirmed Mode to be specified by the service invoker. The choice of Mode may be constrained by the managed object class definition.

5.2 Object Management Function Agreements

5.2.1 General Agreements

These agreements address the following SMF services defined by the object management standard [OMF]:

Table 1: Scope of Agreements Relating to SMF Services Defined by the Object Management Standard [OMF]

Object Management SMF Service	Within Scope Of Agreements	Related Management Support Objects
Object Creation Reporting	Yes	Event Forwarding Discriminator
Object Deletion Reporting	Yes	Event Forwarding Discriminator
Object Name Change Reporting	No	Event Forwarding Discriminator
Attribute Value Change Reporting	Yes	Event Forwarding Discriminator
PT-Create	Yes	

Object Management SMF Service	Within Scope Of Agreements	Related Management Support Objects
PT-Delete	Yes	
PT-Action	Yes	
PT-Set	Yes	
PT-Get	Yes	
PT-Event	Yes	Event Forwarding Discriminator

5.2.2 Object Creation Reporting

This subclause provides agreements pertinent to the Object Creation Reporting SMF service defined by section 9.1.1 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 2: Agreements On Parameter Usage Pertinent to the Object Creation Reporting SMF Service

SMF Object Creation Reporting Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Object Creation	M	C(=)		6.2.6
Event Time	U	-		6.2.3
Create Information				
Source Indicator	U	-		6.3.1.1
Additional Create Information	U	-		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.2.3 Object Deletion Reporting

This subclause provides agreements pertinent to the Object Deletion Reporting SMF service defined by section 9.1.2 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 3: Agreements On Parameter Usage Pertinent to the Object Deletion Reporting SMF Service

SMF Object Deletion Reporting Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Object Deletion	M	C(=)		6.2.6
Event Time	U	-		6.2.3
Delete Information				
Source Indicator	U	-		6.3.1.1
Additional Delete Information	U	-		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.2.4 Object Name Change Reporting

The Object Name Change Reporting SMF service is used by the managed system to report the renaming of a managed object instance to a managing system.

Use of the Object Name Change Reporting SMF service is beyond the scope of these agreements.

5.2.5 Attribute Value Change Reporting

This subclause provides agreements pertinent to the Attribute Value Change Reporting SMF service defined by section 9.1.4 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 4: Agreements On Parameter Usage Pertinent to the Attribute Value Change Reporting SMF Service

SMF Attr Value Change Report Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode`	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Attribute Value Change	M	C(=)		6.2.6
Event Time	U	-		6.2.3
Attribute Change Information				
Attribute Change Definition				
Attribute ID	M	-		6.3.1.1
Old Attr Value	U	-		6.3.1.1
New Attribute Value	M	-		6.3.1.1
Source Indicator	U	-		6.3.1.1
Additional Change Information	U	-		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.2.6 PT-Create

This subclause provides agreements pertinent to the PT-Create SMF service defined by section 9.1.5 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.5 are repeated here for completeness.

Table 5: Agreements On Parameter Usage Pertinent to the PT-Create SMF Service

SMF PT-Create Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Managed Object Class	M	C		6.2.6
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
Attribute List	U	C	[1]	6.2.6, 6.3.5.2
Current Time	-	U		6.2.3
Errors	-	C		6.4.4, 6.3.5.2

[1] This parameter shall be included in ALL success confirmations.

Editor's Note: [It is unclear whether attributes such as objectClass and ID 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 stating that the Managed Object Class and Instance parameters override any values provided in the Attribute List parameter.]

5.2.7 PT-Delete

This subclause provides agreements pertinent to the PT-Delete SMF service defined by section 9.1.6 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.6 are repeated here for completeness.

Table 6: Agreements On Parameter Usage Pertinent to the PT-Delete SMF Service

SMF PT-Delete Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Linked Id	-	C		6.4

SMF PT-Delete Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Base Object Class	M	-		6.2.6
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		6.2.6
Managed Object Instance	-	C		6.2.1
Current Time	-	U		6.2.3
Errors	-	C		6.3.6.1, 6.4.4

5.2.8 PT-Set

This subclause provides agreements pertinent to the PT-Set SMF service defined by section 9.1.8 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.3 are repeated here for completeness.

Table 7: Agreements On Parameter Usage Pertinent to the PT-Set SMF Service

SMF PT-Set 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	-		6.2.6
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

SMF PT-Set Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Managed Object Class	-	C		6.2.6
Managed Object Instance	-	C		6.2.1
Modification List	M	-		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Attribute List	-	U		6.2.6, 6.3.3.1, 6.3.3.3
Current Time	-	U		6.2.3
Errors	-	C		6.3.3.2, 6.4.4

5.2.9 PT-Action

This subclause provides agreements pertinent to the PT-Action SMF service defined by section 9.1.7 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.4 are repeated here for completeness.

Table 8: Agreements On Parameter Usage Pertinent to the PT-Action SMF Service

SMF PT-Action 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	-		6.2.6
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		6.2.6
Managed Object Instance	-	C		6.2.1

SMF PT-Action Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Action Type	M	C(=)		6.2.6
Action Information	U	-		
Current Time	-	U		6.2.3
Action Reply	-	C		
Errors	-	C		6.4.4

5.2.10 PT-Get

This subclause provides agreements pertinent to the PT-Get SMF service defined by section 9.1.9 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.2 are repeated here for completeness.

Table 9: Agreements On Parameter Usage Pertinent to the PT-Get SMF Service

SMF PT-Action Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Linked Id	-	C		6.6
Base Object Class	M	-		6.2.6
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
Attribute ID List	U	-		6.2.6
Managed Object Class	-	C		6.2.6
Managed Object Instance	-	C		6.2.1
Current Time	-	U		6.2.3
Attribute List	-	C		6.2.6, 6.3.2.1, 6.3.2.3

SMF PT-Action Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Errors	-	C		6.3.2.2, 6.4.4

5.2.11 PT-Event

This subclause provides agreements pertinent to the PT-Event SMF service defined by section 9.1.10 of [OMF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 10: Agreements On Parameter Usage Pertinent to the PT-Event SMF Service

SMF PT-Action Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Event Type	M	C(=)		6.2.6
Event Time	U	-		6.2.3
Event Information	U	U		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.3 State Management Function Agreements

5.3.1 General Agreements

These agreements address the following SMF services defined by the state management standard [STMF]:

Table 11: Scope of Agreements Relating to SMF Services Defined by the State Management Standard [STMF]

State Management SMF Service	Within Scope Of Agreements	Related Management Support Objects
State Change Reporting	Yes	Event Forwarding Discriminator

5.3.2 State Change Reporting

This subclause provides agreements pertinent to the State Change Reporting SMF service defined by section 9.3 of [STMF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 12: Agreements On Parameter Usage Pertinent to the State Change Reporting SMF Service

SMF State Change Report Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
State Change	M	U		6.2.6
Event Time	U	-		6.2.3
State Change Information				
Old Operational State	U	-		6.3.1.1
New Operational State	C	-		6.3.1.1
Old Usage State	U	-		6.3.1.1
New Usage State	C	-		6.3.1.1

SMF State Change Report Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Old Administrative State	U	-		6.3.1.1
New Administrative State	C	-		6.3.1.1
Old Repair Status	U	-		6.3.1.1
New Repair Status	C	-		6.3.1.1
Old Installation Status	U	-		6.3.1.1
New Installation Status	C	-		6.3.1.1
Old Availability Status	U	-		6.3.1.1
New Availability Status	C	-		6.3.1.1
Old Control Status	U	-		6.3.1.1
New Control Status	C	-		6.3.1.1
Additional State Change Info	U	-		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.4 Attributes For Representing Relationships Agreements

5.4.1 General Agreements

These agreements address the following SMF services defined by the Attributes For Representing Relationships standard [ARR]:

Table 13: Scope of Agreements Relating to SMF Services Defined by the Attributes For Representing Relationships Standard [ARR]

Attributes For Representing Relationships SMF Service	Within Scope Of Agreements	Related Management Support Objects
Relationship Change Reporting	Yes	Event Forwarding Discriminator

5.4.2 Relationship Change Reporting

This subclause provides agreements pertinent to the Relationship Change Reporting SMF service defined by section 9.3 of [ARR]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 14: Agreements On Parameter Usage Pertinent to the Relationship Change Reporting SMF Service

SMF Rel Change Report Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Relationship Change	M	U		6.2.6
Event Time	U	-		6.2.3
Relationship Change Information				
Old UserObject	U	-		6.3.1.1
New UserObject	C	-		6.3.1.1
Old ProviderObject	U	-		6.3.1.1

SMF Rel Change Report Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
New ProviderObject	C	-		6.3.1.1
Old Peer	U	-		6.3.1.1
New Peer	C	-		6.3.1.1
Old Primary	U	-		6.3.1.1
New Primary	C	-		6.3.1.1
Old Secondary	U	-		6.3.1.1
New Secondary	C	-		6.3.1.1
Old BackUp Object Instance	U	-		6.3.1.1
New Backup Object Instance	C	-		6.3.1.1
Old BackedUp Object Instance	U	-		6.3.1.1
New BackedUp Object Instance	C	-		6.3.1.1
Old Owner	U	-		6.3.1.1
New Owner	C	-		6.3.1.1
Old Member	U	-		6.3.1.1
New Member	C	-		6.3.1.1
Additional Relationship Change Info	U	-		6.3.1.1
Current Time	-	U		6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

5.5 Alarm Reporting Function Agreements

5.5.1 General Agreements

These agreements address the following SMF services defined by the alarm reporting standard [ARF]:

Table 15: Scope of Agreements Relating to SMF Services Defined by the Alarm Reporting Standard [ARF]

Alarm Reporting SMF Service	Within Scope Of Agreements	Related Management Support Objects
Alarm Reporting	Yes	Event Forwarding Discriminator & Alarm Record

5.5.2 Alarm Reporting

This subclause provides agreements pertinent to the Alarm Reporting SMF service defined by section 9.3 of [ARF]. Relevant CMIS agreements defined in subclause 6.3.1 are repeated here for completeness.

Table 16: Agreements On Parameter Usage Pertinent to the Alarm Reporting SMF Service

SMF Alarm Reporting Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M		6.4
Mode	M	-	5.1.2.2	
Managed Object Class	M	U		6.2.6
Managed Object Instance	M	U		6.2.1
Alarm Type	M	C(=)		6.2.6
Event Time	U>M	-	[2]	6.2.3
Alarm Information				
Probable Cause	M	-		6.3.1.1
Specific Problems	U	-	[3]	6.3.1.1
Perceived Severity	M	-		6.3.1.1

SMF Alarm Reporting Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Backup Object Instance	C	-	[1]	6.3.1.1
BackedUp Status	U	-		6.2.1
Trend Indication	U	-		6.3.1.1
Threshold Information	C	-		6.3.1.1
Notification Identifier	U	-	[4]	6.3.1.1
Correlated Notifications	U	-	[3]	6.3.1.1
Generic State Change	C	-		6.3.1.1
Monitored Attributes	U	-		6.3.1.1
Proposed Repair Action	U	-	[3]	6.3.1.1
Problem Text	U	-	[5]	6.3.1.1
Problem Data	U>I	-	[6]	6.3.1.1
Current Time	-	U>M	[2]	6.2.3
Event Reply	-	C		
Errors	-	C		6.4.4

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

[2] To preserve order of events, the Event Time and Current Time parameters shall be mandatory.

[3] To limit implementation complexity, the maximum number of SET items contained within the Specific Problems, Correlated Notifications, and Proposed Repair Action parameters which recipients must be able to process shall be 64.

[4] To limit implementation complexity, the maximum length of the Notification Id parameter shall be 32 bits.

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

[6] Use of the Problem Data parameter is beyond the scope of these agreements.

5.6 Event Report Management Function Agreements

5.6.1 General Agreements

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

Table 17: Scope of Agreements Relating to SMF Services Defined by the Event Report Standard [ERF]

Event Report SMF Service	Within Scope Of Agreements	Related Management Support Objects
Initiation of ERF	Yes	Event Forwarding Discriminator, Event Report Record
Termination of ERF	Yes	Event Forwarding Discriminator, Event Report Record
EFD Modification, Suspension, Resumption	Yes	Event Forwarding Discriminator, Event Report Record

Editor's Note: [The managed object class "eventReportRecord" is defined in [ERMF] but not in [DMI].]

5.6.2 Initiation Of Event Report Forwarding

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

Table 18: Agreements On Parameter Usage Pertinent to the Initiation of Event Report Forwarding SMF Service

SMF Initiation of ERF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Managed Object Class	M	C		6.2.6

SMF Initiation of ERF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
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	6.2.6, 6.3.5.2
Destination Address	U	C		6.2.6, 6.3.5.2
Backup Address	U	C		6.2.6, 6.3.5.2
Active Address	U	C		6.2.6, 6.3.5.2
Administrative State	U	C		6.2.6, 6.3.5.2
Operational State	-	C		6.2.6, 6.3.5.2
Usage State	-	C		6.2.6, 6.3.5.2
Availability Status	-	C		6.2.6, 6.3.5.2
Allomorphic List	U	C		6.2.6, 6.3.5.2
Packages	U	C		6.2.6, 6.3.5.2
Week Mask	U	C	[3]	6.2.6, 6.3.5.2
Intervals Of Day	U	C	[2]	6.2.6, 6.3.5.2
Start Time	U	C	[3]	6.2.6, 6.3.5.2
Stop Time	U	C	[3]	6.2.6, 6.3.5.2
Schedular Name	U>I	C>I	[4]	6.2.6, 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.

5.6.3 Termination Of Event Report Forwarding

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

Table 19: Agreements On Parameter Usage Pertinent to the Termination of Event Report Forwarding SMF Service

SMF Termination of ERF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Invoke Identifier	M	M(=)		6.4
Linked Id	-	C		6.4
Base Object Class	M	-		6.2.6
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		6.2.6
Managed Object Instance	-	C		6.2.1
Current Time	-	U		6.2.3
Errors	-	C		6.3.6.1, 6.4.4

5.6.4 EFD Modification, Suspension, and Resumption

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

Table 20: Agreements On Parameter Usage Pertinent to the Event Forwarding Discriminator Modification, Suspension, and Resumption SMF Service

SMF EFD 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	-		6.2.6
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		6.2.6
Managed Object Instance	-	C		6.2.1
Discriminator Construct	U	C	[1], 5.1.2.1	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Destination Address	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Backup Address	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Active Address	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4

SMF EFD Mod/Suspend/Resume Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Administrative State	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Allomorphic List	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Week Mask	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Intervals Of Day	U	C	[2]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Start Time	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Stop Time	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Schedular Name	U>I	C>I	[4]	6.2.6, 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.

5.7 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 21: 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

Log Control SMF Service	Within Scope Of Agreements	Related Management Support Objects
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 22: 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(=)			
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		

SMF Initiation of LCF Parameter	Req	Rsp	OMF PT-Create & CMIS M-CREATE Parameter	Req	Rsp
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 23: 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		6.2.6
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

SMF Initiation of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Reference Object Instance	U	-		6.2.1
Discriminator Construct	U	C	[1], 5.1.2.1	6.2.6, 6.3.5.2
Administrative State	U	C		6.2.6, 6.3.5.2
Operational State	-	C		6.2.6, 6.3.5.2
Usage State	-	C		6.2.6, 6.3.5.2
Availability Status	-	C		6.2.6, 6.3.5.2
Max Log Size	U	C		6.2.6, 6.3.5.2
Current Log Size	U	C		6.2.6, 6.3.5.2
Number Of Records	U	C		6.2.6, 6.3.5.2
Capacity Alarm Threshold	U	C		6.2.6, 6.3.5.2
Log Full Action	U	C		6.2.6, 6.3.5.2
Packages	U	C		6.2.6, 6.3.5.2
Week Mask	U	C	[3]	6.2.6, 6.3.5.2
Intervals Of Day	U	C	[2]	6.2.6, 6.3.5.2
Start Time	U	C	[3]	6.2.6, 6.3.5.2
Stop Time	U	C	[3]	6.2.6, 6.3.5.2
Schedular Name	U>I	C>I	[4]	6.2.6, 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 [ERF] 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 24: 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	-			

SMF Termination of LCF Parameter	Req	Rsp	PT-Delete & CMIS M-DELETE Parameter	Req	Rsp
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 25: 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	-		6.2.6
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		6.2.6

SMF Termination of LCF Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
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 26: 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	-			
Base Object Instance	M	-			
Scope	U	-			
Filter	U	-			
Access Control	U	-			
Synchronization	U	-			
Managed Object Class	-	C			

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	PT-Set & CMIS M-SET Parameter	Req	Rsp
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 27: 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	

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Base Object Class	M	-		6.2.6
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		6.2.6
Managed Object Instance	-	C		6.2.1
Discriminator Construct	U	C	[1], 5.1.2.1	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Administrative State	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Max Log Size	U	C	[5]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Capacity Alarm Threshold	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Log Full Action	U	C		6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Week Mask	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Intervals Of Day	U	C	[2]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4

SMF LCF Mod/Suspend/Resume Parameter	Req	Rsp	SMF Agreements	CMIS Agreements
Start Time	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Stop Time	U	C	[3]	6.2.6, 6.3.3.1, 6.3.3.3, 6.3.3.4
Schedular Name	U>I	C>I	[4]	6.2.6, 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.]

6 Management Communications

This clause identifies, in detail, use of the management communications services and protocols, based on the standards defined in [CMIS], [CMIP], [ADDRMVS/P] and [CANGETS/P].

This clause covers the agreements pertaining to the use of associations over which to carry management PDUs, agreements pertaining to the services offered to a CMIS Service User (in terms of the functions defined in clause 5), agreements pertaining to the protocol used to convey the management PDUs, and agreements pertaining to the services required of other layers in order to convey the management PDUs defined.

6.1 Association Policies

Associations are established using the procedures described in [ACSEP] with recommendations and extensions described in these implementation agreements.

6.1.1 Application Context Negotiation

These IAs specify the negotiation of application contexts as described in [SMO]. Other application contexts are outside the scope of these agreements.

6.1.2 Functional Unit Negotiation

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

6.1.3 Security Aspects of Associations

Editor's Note: [The security aspects of management associations are being pursued jointly by the NMSIG and the Security SIG. Any agreements generated as a result of this work will be added to this clause as they become available.]

6.2 General Agreements on Users of CMIS

(Refer to the Stable Inplementation 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.)

6.2.2.1 Scoping

(Refer to the Stable Implementation Agreements Document.)

6.2.2.2 Filtering

(Refer to the Stable Implementation Agreements Document.)

6.2.2.3 Synchronization

(Refer to the Stable Implementation Agreements Document.)

6.2.2.4 Multiple Replies

(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 Functional Units

(Refer to the Stable Implementation Agreements Document.)

6.2.6 CMIP Parameters

(Refer to the Stable Implementation Agreements Document.)

6.3 Specific Agreements on Users of CMIS

(Refer to the Stable Implementation Agreements Document.)

6.3.1 M-Event-Report

(Refer to the Stable Implementation Agreements Document.)

6.3.1.1 Event Argument

(Refer to the Stable Implementation Agreements Document.)

6.3.1.2 Parameter Agreements

(Refer to the Stable Implementation Agreements Document.)

6.3.2 M-Get

(Refer to the Stable Implementation Agreements Document.)

6.3.2.1 Successful Response

(Refer to the Stable Implementation Agreements Document.)

6.3.2.2 Partially Successful Response

(Refer to the Stable Implementation Agreements Document.)

6.3.2.3 Multiple Replies

(Refer to the Stable Implementation Agreements Document.)

6.3.2.4 Parameter Agreements

(Refer to the Stable Implementation Agreements Document.)

6.3.3 M-Set

(Refer to the Stable Implementation Agreements Document.)

6.3.3.1 Successful Response

(Refer to the Stable Implementation Agreements Document.)

6.3.3.2 Partially Successful Response

(Refer to the Stable Implementation Agreements Document.)

6.3.3.3 Multiple Replies

(Refer to the Stable Implementation Agreements Document.)

6.3.3.4 Add/Remove Response

(Refer to the Stable Implementation Agreements Document.)

6.3.3.5 Parameter Agreements

(Refer to the Stable Implementation Agreements Document.)

6.3.4 M-Action

(Refer to the Stable Implementation Agreements Document.)

6.3.4.1 Multiple Objects

(Refer to the Stable Implementation Agreements Document.)

6.3.4.2 Parameter Agreements

(Refer to the Stable Implementation Agreements Document.)

6.3.5 M-Create

(Refer to the Stable Implementation Agreements Document.)

6.3.5.1 Managed Object Instance

(Refer to the Stable Implementation Agreements Document.)

6.3.5.2 Attribute Values

(Refer to the Stable Implementation Agreements Document.)

6.3.5.3 Parameter Agreements

(Refer to the Stable Implementation Agreements Document.)

6.3.6 M-Delete

(Refer to the Stable Implementation Agreements Document.)

6.3.6.1 Deletion of Objects Containing Objects

(Refer to the Stable Implementation Agreements Document.)

6.3.6.2 Parameter Agreements

(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.)

6.5 Services Required by CMIP

CMIP requires the services provided by ACSE and ROSE. The conformance requirements for these services, and the underlying communication required to support them, are specified in subclause 5.12.1.

Editor's Note: [Proposed text for the ULSIG subclause 5.12.1 of the OIW Stable Agreements. No agreements beyond the standards are made except where noted.

5.12.1.x Network Management

ROSE Requirements:

The ROSE requirements are as specified in ISO 9596 section 5.2: Underlying Services, and section 6.2 Remote Operations.

Operations Classes

o 1, 2, and 5

Association Classes

o 3

ACSE Requirements:

all

Application Contexts:

o as defined by ISO/DIS 10040 ANNEX A

AE-Title:

o The structure and encoding of the Application Entity title syntax shall be implemented as defined in the amendment to ISO 8650: ISO/IEC JTC1/SC21/WG6/N 3475. Implementations shall, support the form2 of the AE-title choice, i.e., they shall encode and decode the AP-title-form2 as OBJECT IDENTIFIER and the AE-qualifier-form2 as INTEGER. The form1 is outside the scope of these agreements: this does not preclude the ACSE implementation supporting the form1.

Presentation Requirements:

Presentation Functional Units:

- o kernel

Presentation Contexts

- o at least two presentation contexts must be supported

Abstract Syntaxes

- o "ISO 8650-ACSE1" {joint-iso-ccitt(2) association-control(2) abstract-syntax(1) apdus(0) version1(1)}

- o "CMIP-PCI" {joint-iso-ccitt ms(9) cmip(1) version2(2) abstractSyntax(4)}

  Associated Transfer Syntax:

  - o "Basic Encoding of a single ASN.1 Type" {joint-iso-ccitt(2) asn1(1) basic-encoding(1)}

P-DATA Encoding:

- o 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.

Session Requirements

Session Functional Units

- o kernel
- o duplex

Version Number: 2

Maximum Size of User Data parameter field: 10,240 octets.]

6.6 CMIP PICS

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

7 Management Information

This clause, which is based on ISO standards' documents [MIM] and [GDMO], contains agreements regarding basic concepts and modelling techniques related to management information. It enumerates agreements on (i) the information model (subclause 7.1), (ii) principles for naming managed objects and their attributes (subclause 7.2), and (iii) guidelines for defining management information (subclause 7.3). It is not within the scope of this clause to make agreements about specific elements of management information or to define such specific elements of management information. Such definitions and/or agreements can be obtained via the Management Information Library (MIL) produced by the OSI MIB Working Group (a subgroup of the NMSIG).

7.1 The Information Model

This subclause contains agreements related to the information model as specified in clause 5 of [MIM].

7.1.1 Inheritance

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

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

7.1.2 Allomorphism

Allomorphism, as specified in clause 5.1.3 of [MIM], is not supported. Any other specification within the [MIM] or [GDMO] that refers to allomorphism is also not supported. "Not supported", in this context, means that an implementation that complies with the NMSIG IAs is not required to implement allomorphism.

7.1.3 Filter

The concept of filter is supported as specified in clause 5.3 of [MIM]. Restrictions on its usage are specified in subclause 6.2.2.2 and subclause 5.1.3.2 of these agreements. The restrictions in subclause 6.2.2.2 are applicable when the implementation is using "pure" CMIS. If the implementation is using services of the System Management Functions specified in clause 5, the filter restrictions specified in subclause 5.1.3.2 apply.

7.2 Principles of Naming

This subclause contains agreements about principles of naming as specified in clause 6 of [MIM].

7.2.1 Name Structure

7.2.1.1 Object Class Identification

A managed object class is identified by an ASN.1 object identifier, as specified in clause 6.3.1 of [MIM].

7.2.1.2 Object Instance Identification

The distinguished name approach is used for the identification of managed object instances, as specified in clause 6.3.2 of [MIM].

7.2.1.3 Attribute Identification

Each individual attribute of a managed object is identified by an ASN.1 object identifier, as specified in clause 6.3.4 of [MIM].

7.2.1.4 Managed Object Knowledge

Dynamic sharing of management knowledge is not supported. However, all attributes related to shared management knowledge are contained in the managed object class "top", which is defined in [DMI]. Since all managed object class definitions in the MIL are derived from "top" as defined in [DMI], these managed object classes will, by definition, contain the management knowledge attributes.

Since Allomorphism is not supported, the Allomorphic Superclasses attribute, which is one of the attributes defined in "top", will have as its value, the OBJECT IDENTIFIER of the managed object class to which it belongs.

7.3 Guidelines for the Definition of Management Information

This subclause contains agreements about guidelines for the definition of management information, as specified in [GDMO]. These guidelines apply to developers of contributions to the Management Information Library. They form a normative part of the standard; hence they must be strictly followed while defining management information.

7.3.1 Syntactical Definitions of Management Information

7.3.1.1 Managed Object Class Template

The ALLOMORPHIC SET construct of the Managed Object Class Template specified in clause 10.3.2 of [GDMO] is not supported. "Not supported", in this context, means that no managed object class definition in the Management Information Library will contain this construct.

7.3.1.2 Package Template

The following constraints apply to the Package Template specified in clause 10.4.2 of [GDMO]:

> For the ATTRIBUTE GROUPS construct, new attributes shall not be added to the group attribute from within the managed object class template because this can lead to ambiguities. Hence, the [<attribute-label>] portion of the supporting definition for the ATTRIBUTE GROUPS construct shall not be used.
>
> The REGISTERED AS construct shall be mandatory.

7.3.1.3 Attribute Template

The following constraint applies to the Attribute Template specified in clause 10.7.2 of [GDMO]:

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

7.3.2 Guidelines For Defining Behaviour

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

- a textual description of the network resource the managed object class represents, including its functional role.

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

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

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

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

- other constraints, including those involving other managed object classes.

7.3.3 Other Guidelines

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

7.3.4 Initial Value Managed Objects (IVMO)

The following text clarifies underlying concepts of Initial Value Managed Objects (IVMOs) specfied in clause 8.7 of [GDMO]:

> Initial Value Managed Objects, described in clause 8.7 of [GDMO], are most useful in cases where managed object classes that do not support the CREATE operation have been defined. Instances of such object classes would be created as a result of the normal operation of the network, but it may be desirable to be able to control the initial values of attributes of new instances of such object classes via management. IVMOs provide a mechnanism to do this. The NMSIG Transport Connection Profile managed object class defined in the [MIL] is an example of an IVMO. It represents the collection of characteristic attributes that supply default and initially advertised attribute values to be used by instances of the NMSIG Transport Connection managed object class when the instances are created. Since the NMSIG Transport Connection managed object class does not support the management CREATE operation, this IVMO serves as a mechanism which allows initial values of attributes of instances of the NMSIG Transport Connection managed object class to be controlled by management.

8 Conformance

Editor's Note: [The editor has taken the liberty of modifying some of the explanatory text in this clause for clarification of the concepts.]

8.1 Introduction

Clause 8 specifies the conformance requirements for the NMSIG 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 to test implementations claiming conformance to these agreements.

Note: [Conformance requirements for these IAs, relating to services required of the upper layers and other ASEs, are discussed in clause 6.5.]

8.2 General Requirements of Conformance

Conformance for these agreements is designed to specify a well-defined set of services/functions. 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 comprise the three conformance classification areas, respectively. Within these classification areas, particular conformance classes are specified which delineate conformance requirements for a well-defined and bounded set of services/functions (e.g., within the System Management Functions conformance classification area, a conformance class is specificed 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 each conformance class, roles (Manager, Agent, or Manager/Agent) are specified. It is required that one or more roles be supported for each conformance class to which an implementation claims conformance. The development of those conformance classes will enable:

1) users to define procurement specifications.

2) vendors to define systems capabilities and features.

3) conformance test houses to define test cases.

Implementations claiming conformance to these Implementation Agreements shall comply with the requirements stated in the following clauses.

8.3 Management Roles

During a given association, an implementation shall operate in a manager, agent or manager/agent role as specified in clause 6.

A statement of claim, within each PICS or MOCS, shall be provided stating which role(s) (Manager, Agent or Manager/Agent) an implementation supports for each conformance class.

To claim conformance to the IAs, an implementation shall be conformant to at least one role within at least one of the following areas:

o Management Communication

- System Management Functions
- Managed Object Classes

8.4 Specific Conformance Classifications

8.4.1 Management Communication

To be conformant within the Management Communication classification area, an implementation must conform to agreements in clause 6. Conformance to management communication also requires an implementor to state which optional capabilities (e.g., CMIP functional units) are supported in the implementation. These capabilities shall be stated in a PICS or in a high level statement of claim.

Editor's Note: [If a PICS Proforma (Proposed Clause 6.6 in NMSIG/90-121) is not available, what shall be used for phase 1. Issue: Manager, Agent Roles in CMIP]

No implementation claiming to be conformant to any conformance class of these agreements shall violate the protocol requirements specified in the protocol clause of these agreements. Every implementation must 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.

8.4.2 System Management Functions

To be conformant within the System Management Functions classification area, an implementation must state which functional unit(s) it supports.

To be conformant within this classification area, an implementation shall support at least one System Management Function in either a manager or agent role.

To be conformant to the Object Management Function [OMF], an implementation must conform to clause 5.2.

Editor's Note: [Is a test managed object needed, and are both functional units required?]

To be conformant to the State Management Function [STMF], an implementation must conform to clause 5.3 and all clauses referenced in 5.3.

Editor's Note: [Is a test object needed?]

To be conformant to the Attributes for Representing Relationships SMF [ARR], an implementation must conform to clause 5.4 and all clauses referenced in 5.4.

Editor's Note: [Is a test object needed?]

To be conformant to the Alarm Reporting Function [ARF], an implementation must conform to clause 5.5 and all clauses referenced in 5.5.

Editor's Note: [Is a test object needed?]

To be conformant to the Event Reporting Function [ERF], an implementation must conform to clause 5.6 plus the Event Report Record management support object required by the ERF Function.

Editor's Note: [Is a test object needed?]

Editor's Note: [What are the conformance ramifications for PDUs (information) that are outside the scope of these agreements?]

8.4.3 Managed Object

To claim conformance within the Managed Object classification area, an implementation must implement at least one of the following:

- o one or more managed objects from the OIW NMSIG MIL; or
- o any managed object not from OIW NMSIG MIL, providing this managed object is defined according to clause 7. Furthermore, this object shall require NMSIG management communication as specified in clause 6 and, as needed, one or more NMSIG SMFs as specified in clause 5. Managed object class definitions must be provided either in full or by reference to publicly available documents. Associated with any such managed object definition must be a registered managed object class identifier. All manadatory abstract syntaxes and semantics associated with that identifier must be used.

An implementation can claim conformance to a managed object if it meets all the criteria for that object class even if the implementation does not claim conformance to any superior object in the containment tree.

Editor's Note: [Name Binding/ Clarification for this requirement]

8.4.3.1 MOCS Proforma

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

Editor's Note: [The Proforma is possibly a small form to be expanded by implementors for each managed object class.]

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

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

Editor's Note: [Are conditional packages included as an implementor option or an instantiator option?]

ANNEX A -- Management Information Library (MIL)

MANAGEMENT INFORMATION LIBRARY

(MIL)

OSI MIB Working Group
Version 4.0

March 29, 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.

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 classs 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) There will be no obsolescence of any managed object class specified in the MIL.

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 OBJECT CLASSES

A.4.1 Discriminator

This managed object class is used to define the criteria for controlling management services. Refer to [ISO Doc x] for the definition of this managed object class.

A.4.2 Event Forwarding Discriminator

This managed object class is used to define the criteria that must be satisfied by potential event reports before the event reports are forwarded to a particular destination. Refer to [ISO Doc x] for the definition of this managed object class.

A.4.3 NMSIG Agent

A.4.3.1 NMSIG Agent Definition

```
nmsig-agent    MANAGED OBJECT CLASS
   DERIVED FROM  {top}
   CHARACTERISED BY
 BEHAVIOUR DEFINITIONS   agent-behaviour
      ATTRIBUTES   nmsig-agentId  GET,

REGISTERED AS {obj-class}
```

A.4.3.2 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.

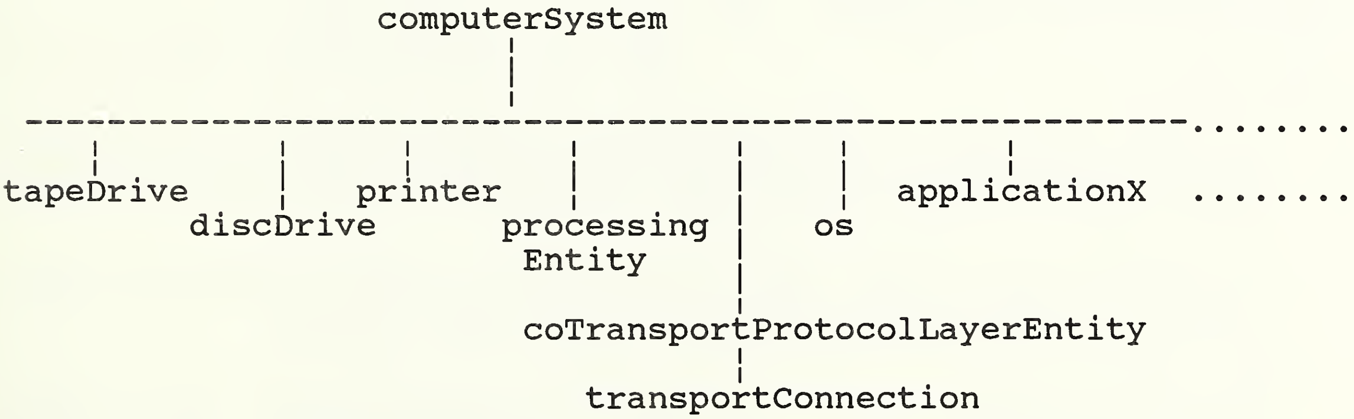
An NMSIG agent system may not support more than one instances of the NMSIG Agent managed object class. If supported, this instance is assumed to be pre-existent when the NMSIG agent system comes up; i.e., management CREATE or DELETE is not supported.

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

A.4.4 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.)



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.4.4.1 NMSIG Computer System Definition

```
nmsig-computerSystem    MANAGED OBJECT CLASS
  DERIVED FROM  {top}
  CHARACTERISED BY
     BEHAVIOUR DEFINITIONS  computerSystem-behaviour
    ATTRIBUTES   nmsig-systemId  GET,
             AdministrativeState  GET-REPLACE
             HealthState  GET,
             OperationalState   GET,
             nmsig-systemTime   GET,                          nmsig-peripheralNames
GET,
             nmsig-userFriendlyLabel  GET-REPLACE
    NOTIFICATIONS     ObjectCreationUnConfirmed,
                 ObjectDeletionUnConfirmed,
```

AttributeChangeUnConfirmed,
StateChangeUnConfirmed,
ProcessingErrorAlarmUnConfirmed,
EnvironmentalAlarmUnConfirmed,
EquipmentAlarmUnConfirmed

REGISTERED AS {obj-class}

A.4.4.2 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 managed object class supports all the values of the administrative state. It supports only the 'enabled' and 'disabled' values of the operational state.

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 CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created computer sytem instance.

The DeleteInfo field of the ObjectDeletion notification shall be NULL.

Attributes that are subject to the AttributeChange notification are: nmsig-peripheralNames, nmsig-userFriendlyLabel, HealthState.

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

A.4.5 NMSIG Connection Oriented Tranport Protocol Layer Entity

A.4.5.1 NMSIG CO Transport Protocol Layer Entity Definition

```
nmsig-coTransportProtocolLayerEntity    MANAGED OBJECT CLASS

   DERIVED FROM          {top}
   CHARACTERIZED BY
      BEHAVIOUR DEFINITIONS   coTransportProtocolLayerEntity-behaviour
     ATTRIBUTES           nmsig-coTransportProtocolLayerEntityId  GET,
                       AdministrativeState  GET-REPLACE,
                       OperationalState  GET,
                       HealthState GET,
                       nmsig-localTransportAddresses  GET,
                       nmsig-maxConnections  GET,
                       nmsig-openConnections  GET,
                       OutgoingConnectionsRequestCounter  GET,
                       IncomingConnectionsRequestCounter  GET,
                       OutgoingConnectionRejectErrorCounter  GET,
                       IncomingConnectionRejectErrorCounter  GET,
                       OutgoingDisconnectErrorCounter  GET,
                       IncomingDisconnectErrorCounter  GET,
                       nmsig-incomingNormalDisconnectCounter  GET,
                       nmsig-outgoingNormalDisconnectCounter  GET,
                       OctetsSentCounter   GET,
                       OctetsReceivedCounter  GET,
                       IncomingTemporalErrorCounter  GET,
                       OutgoingTemporalErrorCounter  GET,
                       nmsig-checksumTPDUsDiscardedCounter  GET,
                       nmsig-transportEntityType GET,
                       nmsig-productInfo GET,
                       nmsig-entityUpTime GET
    NOTIFICATIONS          ObjectCreationUnConfirmed,
                       ObjectDeletionUnConfirmed,
                       AttributeChangeUnConfirmed,
                       StateChangeUnConfirmed,
                       ProcessingErrorAlarmUnConfirmed,
                       nmsig-counterWrapUnConfirmed

REGISTERED AS         {obj-class}
```

A.4.5.2 NMSIG CO Transport Protocol Layer Entity Behaviour

coTransportProtocolLayerEntity-behaviour BEHAVIOUR

DEFINED AS

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.

This managed object class represents a "generic" view of a connection oriented transport protocol layer entity. It does not concern itself with the details of specific transport protocols like ISO TP or TCP. Transport entities that are tied to a specific protocol can be defined as its subclasses; in fact their definitions are being progressed within various standards' bodies. The purpose of defining this managed object class, however, is to provide a common base that will facilitate the high level management of similar but slightly differing resources.

The connection oriented transport protocol layer entity 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 operational 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 operational 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 CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created connection-oriented transport protocol layer entity instance.

The DeleteInfo 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 AttributeChange notification are: nmsig-localTransportAddresses, nmsig-maxConnections, nmsig-productInfo, HealthState.
Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

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

A.4.6 NMSIG Connectionless Network Protocol Layer Entity

A.4.6.1 NMSIG Connectionless Network Protocol Layer Entity Definition

```
nmsig-clNetworkProtocolLayerEntity    MANAGED OBJECT CLASS

   DERIVED FROM           {top}
   CHARACTERIZED BY
        BEHAVIOUR DEFINITIONS  clNetworkProtocolLayerEntity-behaviour
     ATTRIBUTES            nmsig-clNetworkProtocolLayerEntityId  GET,
                   AdministrativeState  GET-REPLACE,
                   OperationalState  GET,
                   HealthState  GET,
                   nmsig-localNetworkAddresses  GET,
                   nmsig-nPDUTimeToLive  GET-REPLACE,
                   PDUsSentCounter  GET,
                   PDUsReceivedCounter  GET,
                   nmsig-PDUsForwardedCounter  GET,
                   nmsig-PDUsReasmbldOKCounter  GET,
                   nmsig-PDUsReasmblFailCounter  GET,
                   nmsig-PDUsDiscardedCounter   GET,
                   nmsig-networkEntityType GET,
                   nmsig-productInfo GET,
                   nmsig-entityUpTime GET

      NOTIFICATIONS        ObjectCreationUnConfirmed,
                   ObjectDeletionUnConfirmed,
                   AttributeChangeUnConfirmed,
                   ProcessingAlarmUnConfirmed,
                   StateChangeUnConfirmed,
                   nmsig-counterWrapUnConfirmed

      PACKAGE          nmsig-clNetworkProtocolLayerEntityRedirection
              PRESENT IF connectionless network protocol layer
                    entity supports redirection of recd PDUs

REGISTERED AS      {obj-class}
```

A.4.6.2 NMSIG Connectionless Network Protocol Layer Entity Behaviour

clNetworkProtocolLayerEntity-behaviour BEHAVIOUR

DEFINED AS

The managed object class nmsig-clNetworkProtocolEntity represents an instantiation of a connectionless network protocol layer. The network layer is layer three of the OSI Reference Model. It 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.

This managed object class represents a "generic" view of a connectionless network protocol layer entity. It does not concern itself with the details of specific network protocols. Network entities that are tied to a specific network protocol can be defined as its subclasses; in fact their definitions are being progressed within various standards' bodies. The purpose of defining this managed object class, however, is to provide a common base that will facilitate the high level management of similar but slightly differing resources.

The NMSIG connectionless network protocol layer entity managed object class supports all the values of the administrative state attribute. It supports only the 'disabled' and 'enabled' values of the operational state attribute.

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 CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created connectionless network protocol layer entity instance.

The DeleteInfo 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 AttributeChange notification are: nmsig-localNetworkAddresses, nmsig-nPDUTimeToLive, nmsig-productInfo, and HealthState

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

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

A.4.6.3 NMSIG CL Network Protocol Layer Entity Redirection Package

```
nmsig-clNetworkProtocolLayerEntityRedirection  CONDITIONAL PACKAGE
          BEHAVIOUR DEFINITIONS  clNetworkProtocolLayerEntityRedirection-
                                    behaviour
          ATTRIBUTES  nmsig-PDUsRedirected  GET
```

REGISTERED AS {package}

clNetworkProtocolLayerEntityRedirection-behaviour BEHAVIOUR

DEFINED AS

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

A.4.7 NMSIG Equipment

A.4.7.1 NMSIG Equipment Definition

```
nmsig-equipment   MANAGED OBJECT CLASS
  DERIVED FROM  {top}
  CHARACTERIZED BY
    BEHAVIOUR DEFINITIONS  equipment-behaviour
    ATTRIBUTES    nmsig-equipmentId  GET,
              OperationalState  GET,
              HealthState  GET,
              AdministrativeState  GET-REPLACE,
              nmsig-locationName  GET-REPLACE,
              nmsig-contactNames  ADD-REMOVE,
              nmsig-equipmentPurpose       GET-REPLACE,
              nmsig-productInfo   GET,
              nmsig-vendorName    GET-REPLACE,
              nmsig-userFriendlyLabel  GET-REPLACE

  NOTIFICATIONS    EnvironmentalAlarmUnConfirmed,
              EquipmentAlarmUnConfirmed,
              ObjectCreationUnConfirmed,
              ObjectDeletionUnConfirmed,
              AttributeChangeUnConfirmed,
              StateChangeUnconfirmed

REGISTERED AS  {obj-class}
```

A.4.7.2 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.

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 managed object class supports all permissible values of the administrative and operational states.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created equipment instance.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted equipment instance.

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

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

A.4.8 NMSIG IEEE 802.3

A.4.8.1 NMSIG IEEE 802.3 Definition

```
nmsig-IEEE-802.3    MANAGED OBJECT CLASS
   DERIVED FROM  {top}
      CHARACTERIZED BY
         BEHAVIOUR DEFINITIONS  iEEE-802.3-behaviour
      ATTRIBUTES   nmsig-IEEE-802.3Id   GET,
                OperationalState   GET,
                AdministrativeState   GET-REPLACE,
                nmsig-macAddress   GET-REPLACE,
                nmsig-IEEE-802.3State   GET-REPLACE,
                nmsig-multicastAddressList   GET-REPLACE,
                HealthState   GET

    OPERATIONS       DELETE
                ACTIONS   nmsig-executeSelfTest

    NOTIFICATIONS   ObjectCreationUnConfirmed,
                ObjectDeletionUnConfirmed,
                AttributeChangeUnConfirmed,
                StateChangeUnconfirmed

REGISTERED AS  {obj-class}
```

A.4.8.2 NMSIG IEEE 802.3 Behaviour

```
iEEE-802.3-behaviour  BEHAVIOUR

  DEFINED AS
```

The managed object class nmsig-IEEE-802.3 represents an instantiation of an IEEE 802.3 CSMA/CD MAC. It may contain either an nmsig-IEEE-802.3-XMT managed object, an nmsig-802.3-RCV managed object, or both of these subordinate objects, as shown in the following figure.

```
+-------------------------------------------+
|  NMSIG IEEE 802.3                         |
|                                           |
| +----------------+     +----------------+ |
| |  NMSIG IEEE    |     |  NMSIG IEEE    | |
| |  802.3 XMT     |     |  802.3 RCV     | |
| +----------------+     +----------------+ |
|                                           |
+-------------------------------------------+
```

The NMSIG IEEE 802.3 managed object class supports only the 'enabled' and 'disabled' values of the operational state attribute. The 'enabled' value indicates that the underlying IEEE 802.3 resource is available for use, and the 'disabled' value indicates that the underlying IEEE 802.3 resource is not available for use.

The NMSIG IEEE 802.3 managed object class supports the DELETE operation; this operation serves to reinitialize the CSMA/CD MAC.

The NMSIG IEEE 802.3 managed object class supports an nmsig-executeSelfTest ACTION; this action causes a self test to be performed on the referenced managed object instance.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created IEEE 802.3 instance.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted IEEE 802.3 instance.

Attributes that are subject to the AttributeChange notification are: nmsig-macAddress, nmsig-multicastAddressList, HealthState.

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

A.4.9 NMSIG IEEE 802.3 RCV

A.4.9.1 NMSIG IEEE RCV Definition

```
nmsig-IEEE-802.3-RCV    MANAGED OBJECT CLASS
   DERIVED FROM  {top}
      CHARACTERIZED BY
      BEHAVIOUR DEFINITIONS  iEEE-802.3-RCV-behaviour
      ATTRIBUTES   nmsig-IEEE-802.3-RCVId   GET,
                OperationalState   GET,
                AdministrativeState   GET-REPLACE,
                HealthState   GET,
                nmsig-multicastRcvState   GET-REPLACE,
                PDUsReceivedCounter   GET,
                nmsig-PDUsFCSErrorCounter    GET,
                nmsig-PDUsAlignmentErrorCounter    GET,
                nmsig-PDUsInRangeLengthErrorCounter   GET,
                nmsig-PDUsOutRangeLengthErrorCounter   GET,
                nmsig-PDUsTooLongErrorCounter   GET
                OctetsReceivedCounter   GET,
                nmsig-multicastPDUsRcvCounter   GET,
                nmsig-broadcastPDUsRcvCounter   GET,
                nmsig-internalMACRcvErrorCounter   GET,
                nmsig-sourceAddrLastFCSErrorPDU   GET,
                nmsig-sourceAddrLastAlignmentErrorPDU   GET,
                nmsig-sourceAddrLastInRangeLengthErrorPDU   GET,
                nmsig-sourceAddrLastOutRangeLengthErrorPDU    GET,
nmsig-sourceAddrLastTooLongErrorPDU   GET,
                nmsig-FCSErrorThreshold   GET-REPLACE,
                nmsig-alignmentErrorThreshold   GET-REPLACE,
                nmsig-inRangeThreshold   GET-REPLACE,
                nmsig-outRangeThreshold   GET-REPLACE,
                nmsig-frameTooLongThreshold   GET-REPLACE,
                nmsig-internalMACRcvErrorThreshold   GET-REPLACE,
                nmsig-enablePromiscuousState  GET-REPLACE

   NOTIFICATIONS    ObjectCreationUnConfirmed,
```

ObjectDeletionUnConfirmed,
AttributeChangeUnConfirmed,
StateChangeUnConfirmed,
ProcessingAlarmUnConfirmed,
nmsig-counterWrapUnConfirmed,
CommunicationAlarmUnConfirmed

REGISTERED AS {obj-class}

A.4.9.2 NMSIG IEEE 802.3 RCV Behaviour

iEEE-802.3-RCV-behaviour BEHAVIOUR

DEFINED AS

The managed object class nmsig-IEEE-802.3-RCV represents an instantiation of an IEEE 802.3 CSMA/CD MAC receiver. This object may be contained within an nmsig-IEEE-802.3 managed object.

The NMSIG IEEE 802.3 RCV managed object class supports only the 'enabled' and 'disabled' values of the operational state attribute. The 'enabled' value indicates that the underlying IEEE 802.3 RCV resource is available for use, and the 'disabled' value indicates that the underlying IEEE 802.3 RCV resource is not available for use.

The definitive description of the counter attributes, their operation and precedence is specified in the [IEEE Doc X].

The NMSIG IEEE 802.3 RCV managed object class supports several threshold attributes; all are associated with the generation of a Communication Alarm notification.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created IEEE 802.3 RCV instance.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted IEEE 802.3 RCV instance.

Attributes that are subject to the AttributeChange notification are: nmsig-multicastRcvState, nmsig-promiscuousRcvState, nmsig-FCSErrorThreshold, nmsig-alignmentErrorThreshold, nmsig-inRangeThreshold, nmsig-outRangeThreshold, HealthState, nmsig-frameTooLongThreshold and nmsig-internalMACRcvErrorThreshold.

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

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

A.4.10 NMSIG IEEE 802.3 XMT

A.4.10.1 NMSIG IEEE 802.3 XMT Definition

```
nmsig-IEEE-802.3-XMT    MANAGED OBJECT CLASS
   DERIVED FROM  {top}
      CHARACTERIZED BY
      BEHAVIOUR DEFINITIONS  iEEE-802.3-XMT-behaviour
      ATTRIBUTES   nmsig-IEEE-802.3-XMTId   GET,
                OperationalState   GET,
                AdministrativeState   GET-REPLACE,
                HealthState  GET,
                nmsig-XmtState  GET-REPLACE,
                PDUsSentCounter   GET,
                nmsig-singleCollisionPDUsCounter   GET,
                nmsig-multipleCollisionPDUsCounter   GET,
                nmsig-lateCollisionsCounter   GET,
                nmsig-PDUsAbortedExcessiveCollisionsCounter   GET,
nmsig-carrierSenseErrorsCounter   GET,
                nmsig-collisionPDUsCounter   GET,
                OctetsSentCounter   GET,
                nmsig-multicastPDUsXmtCounter   GET,
                nmsig-broadcastPDUsXmtCounter   GET,
                nmsig-PDUsLostInternalMACXmtErrorCounter   GET,
                nmsig-PDUsExcessiveDeferralCounter   GET,
                nmsig-collisionPDUsThreshold   GET-REPLACE,
                nmsig-lateCollisionsThreshold   GET-REPLACE,
                nmsig-PDUsAbortedExcessColThreshold   GET-REPLACE,
nmsig-carrierSenseErrorsThreshold   GET-REPLACE,
                nmsig-internalMACXmtErrorThreshold    GET-REPLACE,
nmsig-excessiveDeferralThreshold   GET-REPLACE

   NOTIFICATIONS    ObjectCreationUnConfirmed,
                ObjectDeletionUnConfirmed,
                AttributeChangeUnConfirmed,
                CommunicationAlarmUnConfirmed,
                StateChangeUnConfirmed,
                ProcessingAlarmUnConfirmed,
                nmsig-counterWrapUnConfirmed
REGISTERED AS  {obj-class}
```

A.4.10.2 NMSIG IEEE 802.3 XMT Behaviour

```
iEEE-802.3-XMT-behaviour  BEHAVIOUR

  DEFINED AS
```

The managed object class nmsig-IEEE-802.3-XMT represents an instantiation of an IEEE 802.3 CSMA/CD MAC transmitter. This object may be contained within an nmsig-IEEE-802.3 managed object.

The NMSIG IEEE 802.3 XMT managed object class supports only the 'enabled' and 'disabled' values of the operational state attribute. The 'enabled' value indicates that the underlying IEEE 802.3 XMT resource is available for use, and the 'disabled' value indicates that the underlying IEEE 802.3 XMT resource is not available for use.

The NMSIG IEEE 802.3 XMT managed object class supports both the 'locked' and 'unlocked' values of the administrative state attribute. Unlocking the administrative state serves to enable transmit on the underlying IEEE 802.3 XMT resource.

The definitive description of the counter attributes, their operation and precedence is specified in the [IEEE Doc X].

The NMSIG IEEE 802.3 XMT managed object class supports several threshold attributes; all are associated with the generation of a CommunicationAlarm notification.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created IEEE 802.3 XMT instance, including those inherited from the nmsig-equipment managed object class.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted IEEE 802.3 XMT instance, including those inherited from the nmsig-equipment managed object class.

Attributes that are subject to the AttributeChange notification are: nmsig-collisionPDUsThreshold, nmsig-lateCollisionsThreshold, nmsig-PDUsAbortedExcessColThreshold, nmsig-carrierSenseErrorThreshold, nmsig-internalMACXmtErrorThreshold, nmsig-excessiveDeferralThreshold, HealthState.

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

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

A.4.11 NMSIG LAN MAC Bridge

A.4.11.1 NMSIG LAN MAC Bridge Definition

```
nmsig-LAN-MAC-Bridge    MANAGED OBJECT CLASS
   DERIVED FROM  {nmsig-equipment}
      CHARACACTERIZED BY
```

```
    BEHAVIOUR DEFINITIONS  IAN-MAC-Bridge-behaviour
    ATTRIBUTES   nmsig-packetLossRate     GET,
           nmsig-packetLossRateThreshold  GET-REPLACE

  NOTIFICATIONS    CommunicationAlarm

REGISTERED AS  {obj-class}
```

A.4.11.2 NMSIG LAN MAC Bridge Behaviour

IAN-MAC-Bridge-behaviour BEHAVIOUR

DEFINED AS

A LAN MAC bridge is a device which interconnects two or more MAC domains. A MAC domain is an instance of a MAC algorithm (e.g., a Collision Domain or a Token Domain).

The LAN MAC bridge contains two or more MAC ports each associated with a MAC Domain and operating at layer two of the OSI Model. The function of the LAN MAC bridge is to forward frames from any one MAC Domain to one or more of the other MAC domains. This managed object class represents the LAN MAC bridge device. The definition of this managed object class is based upon the IEEE 802.1 D specification.

The NMSIG LAN MAC bridge managed object class supports only the 'enabled' and 'disabled' values of the operational state attribute. The 'enabled' value indicates that the underlying LAN MAC bridge resource is available for use, and the 'disabled' value indicates that the underlying LAN MAC bridge resource is not available for use.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created LAN MAC Bridge instance.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted LAN MAC Bridge instance.

Attributes, additioal to those that have been inherited from Equipment, that are subject to the AttributeChange notification are: nmsig-packetLossRateThreshold.

A.4.12 NMSIG MAC Port

A.4.12.1 NMSIG MAC Port Definition

```
nmsig-MAC-Port   MANAGED OBJECT CLASS
  DERIVED FROM  {top}
  CHARACTERIZED BY
```

```
    BEHAVIOUR DEFINITIONS  mAC-Port-behaviour
    ATTRIBUTES   nmsig-MAC-PortId   GET,
             nmsig-MAC-PortInNonUCastPktsCounter  GET,
             nmsig-MAC-PortOutNonUCastPktsCounter  GET,
             nmsig-MAC-PortInUCastPktsCounter  GET,
             nmsig-MAC-PortOutUCastPktsCounter  GET,
             nmsig-MAC-PortOutDelayDiscPktsCounter GET,
             nmsig-MAC-PortOutQLen   GET,
             nmsig-MAC-PortInOctetRate   GET,
             nmsig-MAC-PortInOctetRateThreshold  GET-REPLACE,
             AdministrativeState  GET-REPLACE,
             OperationalState  GET,
             HealthState  GET,
             nmsig-broadcastForwardingState  GET-REPLACE,
             nmsig-multicastForwardingState  GET-REPLACE

  NOTIFICATIONS    ObjectCreationUnConfirmed,
             ObjectDeletionUnConfirmed,
             AttributeChangeUnConfirmed,
             StateChangeUnConfirmed,
             nmsig-counterWrapUnConfirmed,
             CommunicationAlarmUnConfirmed

REGISTERED AS  {obj-class}
```

A.4.12.2 NMSIG MAC Port Behaviour

mAC-Port-behaviour BEHAVIOUR

DEFINED AS

This managed object class represents a MAC Port. A MAC Port is contained in a LAN MAC Bridge. It provides the physical connection to a MAC Domain.

The NMSIG MAC Port managed object class supports only the 'enabled' and 'disabled' values of the operational state attribute. The 'enabled' value indicates that the underlying MAC Port resource is available for use, and the 'disabled' value indicates that the underlying MAC port resource is not available for use.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created MAC Port instance.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted MAC Port instance.

Attributes that are subject to the AttributeChange notification are: HealthState, nmsig-MAC-PortInOctetsRateThreshold, nmsig-broadcastForwardingState and nmsig-multicastForwardingState.

Attributes that are subject to the StateChange notification are: AdministrativeState and OperationalState.

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

A.4.13 NMSIG Network

A.4.13.1 NMSIG Network Definition

```
nmsig-network   MANAGED OBJECT CLASS

   DERIVED FROM  {top}
     CHARACTERIZED BY
     BEHAVIOUR DEFINITIONS  network-behaviour
     ATTRIBUTES   nmsig-networkId   GET,
                nmsig-networkPurpose  GET,
                nmsig-userFriendlyLabel  GET-REPLACE

     NOTIFICATIONS  ObjectCreationUnConfirmed,
                  ObjectDeletionUnConfirmed,
                  AttributeChangeUnConfirmed

REGISTERED AS  {obj-class}
```

A.4.13.2 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.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created network instnace.

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the deleted network instance.

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

A.4.14 NMSIG Processing Entity

A.4.14.1 NMSIG Processing Entity Definition

```
nmsig-processingEntity   MANAGED OBJECT CLASS

    DERIVED FROM   {nmsig-equipment}
      CHARACTERIZED BY
       BEHAVIOUR DEFINITIONS  processingEntity-behaviour
       ATTRIBUTES      nmsig-cPU-Type  GET,
                    nmsig-memorySize  GET,
                    nmsig-osInfo   GET,
                    nmsig-entityUpTime  GET

       OPERATIONS      DELETE

       NOTIFICATIONS   ProcessingAlarmUnConfirmed

REGISTERED AS  {obj-class}
```

A.4.14.2 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 NMSIG processing entity managed object class supports all the values of the administrative state. It supports only the enabled and disabled values of the operational state. An instance of the NMSIG Processing Entity managed object class must be created before any of its subordinates are created.

The CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created processing entity instance.

The DeleteInfo 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.4.15 NMSIG Root

A.4.15.1 NMSIG Root Definition

```
nmsig-root        MANAGED OBJECT CLASS

   DERIVED FROM   top
   CHARACTERIZED BY
      BEHAVIOUR DEFINITIONS  root-behaviour

REGISTERED AS {obj-class}
```

A.4.15.2 NMSIG Root Behaviour

```
root-behaviour   BEHAVIOUR

    DEFINED AS
```

This managed object class is used to represent the most superior object instance in the containment tree. The purpose of this managed object class is to serve as the common point from which all instances of managed object classes are named.

A single instance of this managed object class is always present in every system, with a distinguished name that is a null sequence (i.e. a SEQUENCE OF with a length of zero).

A.4.16 NMSIG Transport Connection

A.4.16.1 NMSIG Transport Connection Definition

```
nmsig-transportConnection          MANAGED OBJECT CLASS
   DERIVED FROM             {top}
   CHARACTERIZED BY
      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,
                 PDUsSentCounter  GET,
                 PDUsReceivedCounter  GET,
                 OctetsSentCounter  GET,
                 OctetsReceivedCounter  GET,
                   Peer GET

    OPERATIONS        DELETE   deletes contained objects

    NOTIFICATIONS       ObjectCreationUnConfirmed,
                 ObjectDeletionUnConfirmed,
                 RelationshipChangeUnConfirmed,
                 nmsig-counterWrapUnConfirmed

    PACKAGE      nmsig-transportConnectionRetransmission                PRESENT
IF transport protocol supports retransmission

REGISTERED AS  {obj-class}
```

A.4.16.2 NMSIG Transport Connection Behaviour

```
transportConnection-behaviour  BEHAVIOUR

  DEFINED AS
```

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.

This managed object class represents a "generic" view of a transport connection. It does not concern itself with the details of specific transport protocols like ISO TP or TCP. Transport connections that are tied to a specific protocol can be defined as its subclasses; in fact their definitions are being progressed within various standards' bodies. The purpose of defining this managed object class, however, is to provide a common base that will facilitate the high level management of similar but slightly differing resources.

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.

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

The DeleteInfo field of the ObjectDeletion notification shall contain all the attributes of the created transport connection instance. In addition it shall also contain a 'cause' parameter defined as follows:

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

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.4.16.3 NMSIG Transport Connection Retransmission Package

```
nmsig-transportConnectionRetransmission  CONDITIONAL PACKAGE
    BEHAVIOUR DEFINITIONS  transportConnectionRetransmission-           behaviour
    ATTRIBUTES  nmsig-maxRetransmissions  GET,
          nmsig-retransmissionTimerInitialValue GET,
          PDUsRetransmittedErrorCounter  GET,
          PDUsRetransmittedRate  GET,
          PDUsRetransmittedRateThreshold  GET-REPLACE,
          nmsig-octetsRetransmitted  GET

    NOTIFICATIONS   AttributeChange
            CommunicationAlarmUnConfirmed

REGISTERED AS  {package}
transportConnectionRetransmission-behaviour  BEHAVIOUR

  DEFINED AS
```

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

Attributes that are subject to the AttributeChange notification are: PDUsRetransmittedRateThreshold.

A.4.17 NMSIG Transport Connection Profile

A.4.17.1 NMSIG Transport Connection Profile Definition

```
nmsig-transportConnectionProfile   MANAGED OBJECT CLASS
  DERIVED FROM   {top}
    CHARACTERIZED BY
     BEHAVIOUR DEFINITIONS  trasnportConnectionProfile-behaviour
       ATTRIBUTES   nmsig-transportConnectionProfileId  GET,
                 nmsig-inactivityTimeout  GET-REPLACE,
               nmsig-maxTPDuSize  GET-REPLACE

     OPERATIONS   CREATE,
               DELETE

     NOTIFICATIONS  ObjectCreation
                 ObjectDeletion
                 AttributeChange

REGISTERED AS  {obj-class}
```

A.4.17.2 NMSIG Transport Connection Profile Behaviour

```
transportConnectionProfile-behaviour  BEHAVIOUR

  DEFINED AS
```

This managed object class 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 CreateInfo field of the ObjectCreation notification shall contain all the attributes of the created transport connection profile instance.

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

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

A.4.18 NMSIG Transport Connection Retransmission Profile

A.4.18.1 NMSIG Transport Connection Retransmission Profile Definition

```
nmsig-transportConnectionRetransmissionProfile   MANAGED OBJECT CLASS
  DERIVED FROM   nmsig-transportConnectionProfile
    CHARACTERIZED BY
      BEHAVIOUR DEFINITIONS  transportConnectionProfile-behaviour
        ATTRIBUTES  nmsig-maxRetransmissions  GET-REPLACE,
                nmsig-retransmissionTimerInitialValue GET-REPLACE

REGISTERED AS  {obj-class}
```

A.4.18.2 NMSIG Transport Connection Retransmission Profile Behaviour

```
transportConnectionRetransmissionProfile-behaviour  BEHAVIOUR

  DEFINED AS
```

This managed object class 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 AttributeChange notification are : nmsig-maxRetransmissions, nmsig-retransmissionTimerInitialValue

A.4.19 Top

This managed object class represents the root of the inheritance tree.

Refer to [ISO Doc x] for the definition of this managed object class.

A.5 NAME BINDINGS

This clause provides definitions of NAME BINDINGS for the managed object classes defined by the OSI MIB Working Group. NAME BINDINGs for managed object classes defined by other groups can be found in the document referenced under the managed object class definition in section 3.

A.5.1 Event Forwarding Discriminator Name Bindings

```
EventForwardingDiscriminator-nb-1   NAME BINDING

EventForwardingDiscriminator   IS NAMED BY  nmsig-agent
        WITH ATTRIBUTE  DiscriminatorId

REGISTERED AS   {nmsig-nb}
```

A.5.2 NMSIG Agent Name Bindings

```
nmsig-agent-nb-1   NAME BINDING

nmsig-agent IS NAMED BY nmsig-root
       WITH ATTRIBUTE  nmsig-agentId

REGISTERED AS    {nmsig-nb}
```

A.5.3 NMSIG Computer System Name Bindings

```
nmsig-computerSystem-nb-1  NAME BINDING

nmsig-computerSystem IS NAMED BY nmsig-network
   WITH ATTRIBUTE  nmsig-systemId

REGISTERED AS    {nmsig-nb}


nmsig-computerSystem-nb-2  NAME BINDING

nmsig-computerSystem IS NAMED BY nmsig-computerSystem
   WITH ATTRIBUTE  nmsig-systemId

REGISTERED AS    {nmsig-nb}


nmsig-computerSystem-nb-3  NAME BINDING
```

```
nmsig-computerSystem IS NAMED BY nmsig-root
   WITH ATTRIBUTE  nmsig-systemId

REGISTERED AS    {nmsig-nb}
```

A.5.4 NMSIG CO Transport Protocol Layer Entity Name Bindings

```
nmsig-coTransportProtocolLayerEntity-nb-1  NAME BINDING

nmsig-coTransportProtocolLayerEntity  IS NAMED BY  nmsig-computerSystem
   WITH ATTRIBUTE   nmsig-coTransportEntityId

REGISTERED AS  {nmsig-nb}


nmsig-coTransportProtocolLayerEntity-nb-2  NAME BINDING

nmsig-coTransportProtocolLayerEntity  IS NAMED BY  nmsig-equipment
   WITH ATTRIBUTE   nmsig-coTransportEntityId

REGISTERED AS  {nmsig-nb}
```

A.5.5 NMSIG CL Network Protocol Layer Entity Name Bindings

```
nmsig-clNetworkProtocolLayerEntity-nb-1   NAME BINDING

nmsig-clNetworkProtocolLayerEntity  IS NAMED BY  nmsig-computerSystem
   WITH ATTRIBUTE   nmsig-clNetworkProtocolEntityId

REGISTERED AS      {nmsig-nb}


nmsig-clNetworkProtocolLayerEntity-nb-2   NAME BINDING

nmsig-clNetworkProtocolLayerEntity  IS NAMED BY  nmsig-equipment
   WITH ATTRIBUTE   nmsig-clNetworkProtocolEntityId

REGISTERED AS      {nmsig-nb}
```

A.5.6 NMSIG Equipment Name Bindings

```
nmsig-equipment-nb-1   NAME BINDING

nmsig-equipment IS NAMED BY  nmsig-equipment
   WITH ATTRIBUTE  nmsig-equipmentId
```

REGISTERED AS {nmsig-nb}

nmsig-equipment-nb-2 NAME BINDING

nmsig-equipment IS NAMED BY nmsig-network
WITH ATTRIBUTE nmsig-equipmentId
REGISTERED AS {nmsig-nb}

nmsig-equipment-nb-3 NAME BINDING

nmsig-equipment IS NAMED BY nmsig-root
WITH ATTRIBUTE nmsig-equipmentId

REGISTERED AS {nmsig-nb}

A.5.7 NMSIG IEEE 802.3 Name Bindings

nmsig-IEEE-802.3-nb-1 NAME BINDING

nmsig-IEEE-802.3 IS NAMED BY nmsig-network
WITH ATTRIBUTE nmsig-IEEE-802.3Id

REGISTERED AS {nmsig-nb}

nmsig-IEEE-802.3-nb-2 NAME BINDING

nmsig-IEEE-802.3 IS NAMED BY nmsig-computerSystem
WITH ATTRIBUTE nmsig-IEEE-802.3Id

REGISTERED AS {nmsig-nb}

A.5.8 NMSIG IEEE 802.3 RCV Name Bindings

nmsig-IEEE-802.3-RCV-nb-1 NAME BINDING

nmsig-IEEE-802.3-RCV IS NAMED BY nmsig-IEEE-802.3
WITH ATTRIBUTE nmsig-IEEE-802.3-RCVId

REGISTERED AS {nmsig-nb}

A.5.9 NMSIG IEEE 802.3 XMT Name Bindings

nmsig-IEEE-802.3-XMT-nb-1 NAME BINDING

nmsig-IEEE-802.3-XMT IS NAMED BY nmsig-IEEE-802.3
  WITH ATTRIBUTE nmsig-IEEE-802.3-XMTId

REGISTERED AS {nmsig-nb}

A.5.10 NMSIG LAN MAC Bridge Name Bindings

nmsig-LAN-MAC-Bridge-nb-1 NAME BINDING
nmsig-LAN-MAC-Bridge IS NAMED BY nmsig-network
  WITH ATTRIBUTE nmsig-equipmentId

REGISTERED AS {nmsig-nb}

A.5.11 NMSIG MAC Port Name Bindings

nmsig-MAC-Port-nb-1 NAME BINDING

nmsig-MAC-Port IS NAMED BY nmsig-LAN-MAC-Bridge
  WITH ATTRIBUTE nmsig-MAC-PortId

REGISTERED AS {nmsig-nb}

A.5.12 NMSIG Network Name Bindings

nmsig-network-nb-1 NAME BINDING

nmsig-network IS NAMED BY nmsig-network
  WITH ATTRIBUTE nmsig-networkId

REGISTERED AS {nmsig-nb}

nmsig-network-nb-2 NAME BINDING

nmsig-network IS NAMED BY nmsig-root
  WITH ATTRIBUTE nmsig-networkId

REGISTERED AS {nmsig-nb}

A.5.13 NMSIG Processing Entity Name Bindings

```
nmsig-processingEntity-nb-1 NAME BINDING

nmsig-processingEntity IS NAMED BY nmsig-computerSystem
  WITH ATTRIBUTE  nmsig-equipmentId

REGISTERED AS {nmsig-nb}
```

A.5.14 NMSIG Transport Connection Name Bindings

```
nmsig-transportConnection-nb-1   NAME BINDING

nmsig-transportConnection
   IS NAMED BY  nmsig-coTransportProtocolLayerEntity
   WITH ATTRIBUTE   nmsig-transportConnectionId

REGISTERED AS  {nmsig-nb}
```

A.5.15 NMSIG Transport Connection Profile Name Bindings

```
nmsig-transportConnectionProfile-nb-1   NAME BINDING

nmsig-transportConnectionProfile
   IS NAMED BY nmsig-coTransportProtocolLayerEntity
   WITH ATTRIBUTE   nmsig-transportConnectionProfileId

REGISTERED AS  {nmsig-nb}
```

A.5.16 NMSIG Transport Connection Retransmission Profile Name Bindings

```
nmsig-transportConnectionRetransmissionProfile-nb-1   NAME BINDING

nmsig-transportConnectionRetransmissionProfile
   IS NAMED BY nmsig-coTransportProtocolLayerEntity
   WITH ATTRIBUTE   nmsig-transportConnectionProfileId

REGISTERED AS  {nmsig-nb}
```

A.6 ATTRIBUTES

This clause provides definitions of attributes contained in the managed object classes defined by the OSI MIB Working Group. Attribute definitions for managed object classes defined by other groups can be found in the document referenced under the managed object class definition in section 3.

A.6.1 Administrative State

Refer to [ISO Doc x] for the definition of this attribute.

A.6.2 Begin Time

Refer to [ISO Doc x] for the definition of this attribute.

A.6.3 Destination Address

Refer to [ISO Doc x] for the definition of this attribute.

A.6.4 Discriminator Construct

Refer to [ISO Doc x] for the definition of this attribute.

A.6.5 Discriminator Id

Refer to [ISO Doc x] for the definition of this attribute.

A.6.6 End Time

Refer to [ISO Doc x] for the definition of this attribute.

A.6.7 Health State

Refer to [ISO Doc x] for the definition of this attribute.

A.6.8 Incoming Connection Reject Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.9 Incoming Connection Requests Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.10 Incoming Disconnect Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.11 Incoming Temporal Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.12 NMSIG Alignment Error Threshold

```
nmsig-alignmentErrorThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
        BEHAVIOUR  alignmentErrorThreshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::= {as defined in ISO Doc X}

alignmentErrorThreshold-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the alignment error rate. The alignment error rate is defined as the number of PDUs received with alignment errors divided by the total number of PDUs received. A communication alarm notification is emitted when the alignment error rate exceeds the threshold value.

A.6.13 NMSIG Agent Id

```
nmsig-agentId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        BEHAVIOUR  agentId-behaviour
REGISTERED AS     {nmsig-attr}

agentId-behaviour  BEHAVIOUR

      DEFINED AS
```

This is the distinguishing attribute for the managed object class NMSIG Agent.

A.6.14 NMSIG Broadcast Forwarding State

```
nmsig-broadcastForwardingState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  State
        MATCHES FOR  Equality
        BEHAVIOUR  broadcastForwardingState-behaviour

REGISTERED AS   {nmsig-attr}

State ::=  ENUMERATED {off (0),
                   on (1)}


broadcastForwardingState-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies whether broadcast PDUs are being forwarded.

A.6.15 NMSIG Broadcast PDUs Rcv Counter

```
nmsig-broadcastPDUsRcvCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
        BEHAVIOUR  broadcastPDUsRcvCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

broadcastPDUsRcvCounter-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of broadcast PDUs received ok by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.16 NMSIG Broadcast PDUs Xmt Counter

```
nmsig-broadcastPDUsXmtOkCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
        BEHAVIOUR  broadcastPDUsXmtOkCounter-behaviour

REGISTERED AS     {nmsig-attr}
```

```
Count ::= {as defined in ISO Doc X}

broadcastPDUsXmtOkCounter-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of broadcast PDUs which were transmitted ok by
          the underlying NMSIG IEEE 802.3 XMT resource.
```

A.6.17 NMSIG Carrier Sense Errors Counter

```
nmsig-carrierSenseErrorsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
        BEHAVIOUR  carrierSenseErrorsCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

carrierSenseErrorsCounter-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of carrier sense Errors which were detected by the
          underlying NMSIG IEEE 802.3 XMT resource.
```

A.6.18 NMSIG Carrier Sense Errors Threshold

```
nmsig-carrierSenseErrorsThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
        BEHAVIOUR  carrierSenseErrorsThreshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

carrierSenseErrorsThreshold-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies a threshold which is applied against the carrier sense error
          rate. The carrier sense error rate is defined as the carrier sense errors detected per
          second.  A communication alarm notification is emitted when the carrier sense error
          rate exceeds the threshold value.
```

A.6.19 NMSIG Checksum TPDUs Discarded Counter

```
nmsig-checksumTPDUsDiscardedCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
        BEHAVIOUR  checksumTPDUsDiscardedCounter-behaviour

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}
checksumTPDUsDiscardedCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of TPDUs discarded due to a bad checksum.

A.6.20 NMSIG Collision PDUs Counter

```
nmsig-collisionPDUsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  collisionPDUsCounter-behaviour

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

collisionPDUsCounter-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of collision PDUs which were detected by the underlying NMSIG IEEE 802.3 XMT resource.

A.6.21 NMSIG Collision PDUs Threshold

```
nmsig-collisionPDUsThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
            BEHAVIOUR  collisionPDUsThreshold-behaviour

REGISTERED AS      {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}
```

collisionPDUsThreshold-behaviour BEHAVIOUR

DEFINED AS

This attribute specifies a threshold which is applied against the collision PDU rate. The collision PDU rate is defined as the collision PDUs detected per second. A communication alarm notification is emitted when the collision PDU rate exceeds the threshold value.

A.6.22 NMSIG CO Transport Protocol Layer Entity Id

```
nmsig-coTransportEntityId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
            BEHAVIOUR  coTransportEntityId-behaviour

REGISTERED AS     {nmsig-attr}

coTransportEntityID-behaviour  BEHAVIOUR
```

DEFINED AS

This is the distinguishing attribute for the managed object class connection oriented transport protocol layer entity.

A.6.23 NMSIG Connectionless Network Protocol Layer Entity Id

```
nmsig-clNetworkProtocolLayerEntityId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
            BEHAVIOUR  clNetworkProtocolLayerEntityId-behaviour

REGISTERED AS     {nmsig-attr}

clNetworkProtocolLayerEntityId-behaviour   BEHAVIOUR
```

DEFINED AS

This attribute is the distinguishing attribute for the managed object class clNetworkProtocolLayerEntity.

A.6.24 NMSIG Contact Names

```
nmsig-contactNames   ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX   AnyName
        MATCHES FOR  Set Comparison, Set Intersection
            BEHAVIOUR  contactNames-behaviour

REGISTERED AS     {nmsig-attr}

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

contactNames-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies name(s) of one or more contacts.
```

A.6.25 NMSIG CPU Type

```
nmsig-cPU-Type   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  PrintableString
      MATCHES FOR  Equality
          BEHAVIOUR  cPU-Type-behaviour

REGISTERED AS   {nmsig-attr}

cPU-Type-behaviour  BEHAVIOUR

    DEFINED AS

          This attribute specifies the type of the Central Processor Unit in a processing entity.
```

A.6.26 NMSIG Enable Promiscuous State

```
nmsig-enablePromiscuousState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  State
        MATCHES FOR  Equality
             BEHAVIOUR  enablePromiscuousState-behaviour

REGISTERED AS   {nmsig-attr}

State ::=  ENUMERATED {off (0),
                  on (1)}

enablePromiscuousState-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute specifies whether the IEEE 802.3 RCV is operating in promiscuous mode.

A.6.27 NMSIG Entity Up Time

```
nmsig-entityUpTime    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  entityUpTime-behaviour

REGISTERED AS      {nmsig-attr}

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.6.28 NMSIG Equipment Id

```
nmsig-equipmentId   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  equipmentId-behaviour

REGISTERED AS      {nmsig-attr}

equipmentId-behaviour  BEHAVIOUR

    DEFINED AS
```

This is the distinguishing attribute of the NMSIG equipment managed object class.

A.6.29 NMSIG Equipment Purpose

```
nmsig-equipmentPurpose  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  equipmentPurpose-behaviour

REGISTERED AS      {nmsig-attr}

equipmentPurpose-behaviour  BEHAVIOUR
```

DEFINED AS

This attribute specifies what the equipment is used for (e.g., switching, processing, etc.).

A.6.30 NMSIG Excessive Deferral Threshold

```
nmsig-excessiveDeferralThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  excessiveDeferralThreshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

excessiveDeferralThreshold-behaviour   BEHAVIOUR
```

DEFINED AS

This attribute specifies a threshold which is applied against the excessive deferral rate. The excessive deferral rate is defined as the number of excessive deferral PDUs transmitted per second. A communication alarm notification is emitted when the excessive deferral rate exceeds the threshold value.

A.6.31 NMSIG FCS Error Threshold

```
nmsig-FCSErrorThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  fCSErrorThreshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined by ISO Doc X}

fCSErrorThreshold-behaviour   BEHAVIOUR
```

DEFINED AS

This attribute specifies a threshold which is applied against the FCS error rate. The FCS error rate is defined as the number of PDUs received which had FCS errors divided by the total number of PDUs received. A communication alarm notification is emitted when the FCS error rate exceeds the threshold value.

A.6.32 NMSIG IEEE 802.3 Id

```
nmsig-IEEE-802.3Id   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  iEEE-802.3Id-behaviour

REGISTERED AS     {nmsig-attr}

iEEE-802.3Id-behaviour  BEHAVIOUR

    DEFINED AS

        This attribute is the distinguishing attribute of the NMSIG IEEE 802.3 managed object
        class.
```

A.6.33 NMSIG IEEE 802.3 RCV Id

```
nmsig-IEEE-802.3-RCVId   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  iEEE-802.3-RCVId-behaviour

REGISTERED AS     {nmsig-attr}
iEEE-802.3-RCVId-behaviour  BEHAVIOUR

    DEFINED AS

        This attribute is the distinguishing attribute of the NMSIG IEEE 802.3 RCV managed
        object class.
```

A.6.34 NMSIG IEEE 802.3 State

```
nmsig-IEEE-802.3State  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX EnableState
        MATCHES FOR  Equality
             BEHAVIOUR  iEEE-802.3State-behaviour

REGISTERED AS   {nmsig-attr}

EnableState ::=  ENUMERATED {disable (0),
                      enable (1)}

iEEE-802.3State-behaviour  BEHAVIOUR
```

DEFINED AS

This attribute specifies whether the IEEE 802.3 object is enabled or not. The 'enabled' and 'disabled' values of this attribute correspond to the 'enabled' and 'disabled' values of the OperationalState attribute. (This attribute was introduced as a GET-REPLACE attribute which can be used by management to enable or disable the underlying IEEE 802.3 resource.)

A.6.35 NMSIG IEEE 802.3 XMT Id

```
nmsig-IEEE-802.3-XMTId   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  iEEE-802.3-XMTId-behaviour

REGISTERED AS     {nmsig-attr}

iEEE-802.3-XMTId-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute is the distinguishing attribute of the NMSIG IEEE 802.3 XMT managed object class.

A.6.36 NMSIG In-Range Threshold

```
nmsig-inRangeThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  inRangeTheshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined by ISO Doc X}

inRangeTheshold-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the in-range length error rate. The in-range length error rate is defined as the number of PDUs received that had in-range length errors divided by the total number of PDUs received. A communication alarm notification with the specified severity is emitted when the in-range length error rate exceeds the threshold value.

A.6.37 NMSIG Inactivity Timeout

```
nmsig-inactivityTimeout          ATTRIBUTE
        WITH ATTRIBUTE SYNTAX INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  inactivityTimeout-behaviour

REGISTERED AS      {nmsig-attr}

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.6.38 NMSIG Incoming Normal Disconnect Counter

```
nmsig-incomingNormalDisconnectCounter        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  incomingNormalDisconnectCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

incomingNormalDisconnectCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of incoming transport connections that were disconnected due to normal reasons.

A.6.39 NMSIG Internal MAC Rcv Error Threshold

```
nmsig-internalMACRcvErrorThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  internalMACRcvErrorThreshold

REGISTERED AS     {nmsig-attr}
```

```
GaugeThreshold ::=  {as defined in ISO Doc X}

internalMACRcvErrorThreshold   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the internal MAC receive error rate. This rate is defined as the number of internal MAC receive errors detected per second. A communication alarm notification is emitted when the internal MAC receive error rate exceeds the threshold value.

A.6.40 NMSIG Internal MAC Rcv Error Counter

```
nmsig-internalMACRcvErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  internalMACRcvErrorCounter

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

internalMACRcvErrorCounter   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of internal MAC receive errors detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.41 NMSIG Internal MAC Xmt Error Threshold

```
nmsig-internalMACXmtErrorThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  internalMACXmtErrorThreshold-behaviour

REGISTERED AS      {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

internalMACXmtErrorThreshold-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the internal MAC transmit error rate. This rate is defined as the number of internal MAC transmit errors detected per second. A communication alarm notification is emitted when the internal MAC transmit error rate exceeds the threshold value.

A.6.42 NMSIG Late Collision Counter

```
nmsig-lateCollisionsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  lateCollisionsCounter-behaviour

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

lateCollisionsCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of late collisions detected by the underlying NMSIG IEEE 802.3 XMT resource.

A.6.43 NMSIG Late Collisions Threshold

```
nmsig-lateCollisionThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  lateCollisionThreshold-behaviour

REGISTERED AS      {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

lateCollisionThreshold-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the late collision rate. The late collision rate is defined as the number of late collision PDUs transmitted divided by the total number of PDUs transmitted. A communication alarm notification is emitted when the late collision rate exceeds the threshold value.

A.6.44 NMSIG Local Network Address

```
nmsig-localNetworkAddress       ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OCTET STRING
        MATCHES FOR   Equality
             BEHAVIOUR  localNetworkAddress-behaviour

REGISTERED AS       {nmsig-attr}

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.6.45 NMSIG Local Network Addresses

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

REGISTERED AS       {nmsig-attr}

LocalNetworkAddresses  ::=  SET OF OCTET STRING

localNetworkAddresses-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a set of local network addresses supported by a network protocol layer entity.

A.6.46 NMSIG Local Transport Addresses

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

REGISTERED AS  {nmsig-attr}

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

```
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.6.47 NMSIG Local Transport Connection Endpoint

```
nmsig-localTransportConnectionEndpoint      ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OCTET STRING
        MATCHES FOR  Equality
            BEHAVIOUR  localTransportConnectionEndpoint-behaviour

REGISTERED AS      {nmsig-attr}

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.6.48 NMSIG Location Name

```
nmsig-locationName   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   AnyName
        MATCHES FOR  Equality
            BEHAVIOUR  locationName-behaviour

REGISTERED AS     {nmsig-attr}

AnyName ::=  CHOICE {dn   DistinguishedName,
                      ps   PrintableString}

locationName-behaviour   BEHAVIOUR

   DEFINED AS
```

This attribute specifies the name of a location (e.g., Hilo Hawaii USA).

A.6.49 NMSIG MAC Address

```
nmsig-macAddress   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   OctetString
        MATCHES FOR  Equality
             BEHAVIOUR  macAddress-behaviour

REGISTERED AS     {nmsig-attr}

macAddress-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies a MAC address.
```

A.6.50 NMSIG MAC Port Id

```
nmsig-MAC-PortId   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  mAC-PortID-behaviour

REGISTERED AS     {nmsig-attr}

mAC-PortID-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute is the distinguishing attribute of the NMSIG MAC Port managed object
          class.
```

A.6.51 NMSIG MAC Port In Non-Unicast Packets Counter

```
nmsig-MAC-PortInNonUCastPktsCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  mAC-PortInNonUCastPktsCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

mAC-PortInNonUCastPktsCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of non-unicast (i.e., subnet broadcast or subnet multicast) packets that were received at the MAC port.

A.6.52 NMSIG MAC Port In Octet Rate

```
nmsig-MAC-PortInOctetRate  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Gauge
        MATCHES FOR  Equality, Ordering
            BEHAVIOUR  mAC-PortInOctetRate

REGISTERED AS     {nmsig-attr}

Gauge ::=  {as defined in ISO doc X}

mAC-PortInOctetRate   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the rate of octets arriving at the MAC port per second.

A.6.53 NMSIG MAC Port In Octet Rate Threshold

```
nmsig-MAC-PortInOctetRateThreshold   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  GaugeThreshold
        MATCHES FOR  Equality
            BEHAVIOUR  mAC-PortInOctetRateThreshold

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

mAC-PortInOctetRateThreshold   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the in octet rate. A communication alarm notification is emitted when the in octet rate exceeds the threshold value.

A.6.54 NMSIG MAC Port In Unicast Packets Counter

```
nmsig-MAC-PortInUCastPktsCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
```

```
            BEHAVIOUR  mAC-PortInUCastPktsCounter

REGISTERED AS     {nmsig-attr}
Count ::= {as defined in ISO Doc X}

mAC-PortInUCastPktsCounter   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of unicast packets received on the MAC port.
```

A.6.55 NMSIG MAC Port Out Delay Discarded Packets Counter

```
nmsig-MAC-PortOutDelayDiscPktsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  mAC-PortOutDelayDiscPktsCounter

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

mAC-PortOutDelayDiscPktsCounter   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of packets that were discarded at the MAC port
          because the maximum packet hold time was exceeded.
```

A.6.56 NMSIG MAC Port Out Non-Unicast Packets

```
nmsig-MAC-PortOutNonUCastPktsCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  mAC-PortOutNonUCastPktsCounter

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

mAC-PortOutNonUCastPktsCounter   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of non-unicast packets that were sent out of the MAC port.

A.6.57 NMSIG MAC Port Out Queue Length

```
nmsig-MAC-PortOutQLen  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  mAC-PortOutQLen

REGISTERED AS      {nmsig-attr}

mAC-PortOutQLen   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of packets that are currently queued for output on the MAC port.

A.6.58 NMSIG MAC Port Out Unicast Packets Counter

```
nmsig-MAC-PortOutUCastPktsCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  mAC-PortOutUCastPktsCounter

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

mAC-PortOutUCastPktsCounter   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of unicast packets that were sent out of this MAC port.

A.6.59 NMSIG Max Connections

```
nmsig-maxConnections          ATTRIBUTE
        WITH ATTRIBUTE SYNTAX INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  maxConnections-behaviour
```

```
REGISTERED AS     {nmsig-attr}

maxConnections-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the maximum number of simultaneously open transport connections allowed by the transport protocol layer entity.

A.6.60 NMSIG Max Retransmissions

```
nmsig-maxRetransmissions  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  maxRetransmissions-behaviour

REGISTERED AS     {nmsig-attr}

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.6.61 NMSIG Max TPDU Size

```
nmsig-maxTPDUSize            ATTRIBUTE
        WITH ATTRIBUTE SYNTAX INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  maxTPDUSize-behaviour

REGISTERED AS      {nmsig-attr}

maxTPDUSize-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the maximum TPDU size (in terms of octets) that can be handled by the local transport protocol layer entity.

A.6.62 NMSIG Memory Size

```
nmsig-memorySize   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  INTEGER
        MATCHES FOR   Equality, Ordering
```

```
            BEHAVIOUR  memorySize-behaviour

REGISTERED AS   {nmsig-attr}

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.6.63 NMSIG Multicast Address List

```
nmsig-multicastAddressList   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   AddressList
        MATCHES FOR  Set Comparison, Set Intersection
            BEHAVIOUR  multicastAddressList-behaviour

REGISTERED AS    {nmsig-attr}

AddressList ::=  SET OF OCTET STRING

multicastAddressList-behaviour   BEHAVIOUR

   DEFINED AS
```

This attribute specifies a multicast address list.

A.6.64 NMSIG Multicast Forwarding State

```
nmsig-multicastForwardingState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  State
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  multicastForwardingState-behaviour

REGISTERED AS   {nmsig-attr}

State ::=  ENUMERATED {off (0),
                 on (1)}

multicastForwardingState-behaviour   BEHAVIOUR

   DEFINED AS
```

This attribute specifies whether multicast PDUs are being forwarded.

A.6.65 NMSIG Multicast PDUs Rcv Counter

```
nmsig-multicastPDUsRcvCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  multicastPDUsRcvCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

multicastPDUsRcvCounter-behaviour  BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of multicast PDUs received ok by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.66 NMSIG Multicast PDUs Xmt Counter

```
nmsig-multicastPDUsXmtCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  multicastPDUsXmtCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

multicastPDUsXmtCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of multicast PDUs which were transmitted ok by the underlying NMSIG IEEE 802.3 XMT resource.

A.6.67 NMSIG Multicast Receive State

```
nmsig-multicastReceiveState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  State
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  multicastReceiveState-behaviour

REGISTERED AS   {nmsig-attr}
```

```
State ::=  ENUMERATED {off (0),
                on (1)}
```

multicastReceiveState-behaviour BEHAVIOUR

DEFINED AS

This attribute specifies the multicast receive state of the underlying NMSIG IEEE 802.3 RCV resource.

A.6.68 NMSIG Multiple Collision PDU Counter

```
nmsig-multipleCollisionPDUCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  multipleCollisionPDUCounter

REGISTERED AS     {nmsig-attr}
Count ::= {as defined in ISO Doc X}
```

multipleCollisionPDUCounter BEHAVIOUR

DEFINED AS

This attribute specifies the number of multiple collision PDUs detected by the underlying NMSIG IEEE 802.3 XMT resource.

A.6.69 NMSIG Network Entity Type

```
nmsig-networkEntityType ATTRIBUTE
      WITH ATTRIBUTE SYNTAX NetworkEntityType
      MATCHES FOR  Equality
          BEHAVIOUR  networkEntityType-behaviour

REGISTERED AS     {nmsig-attr}

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

networkEntityType-behaviour BEHAVIOUR

DEFINED AS

This attribute specifies the type of the network protocol layer entity.

A.6.70 NMSIG Network Id

```
nmsig-networkId   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  networkId-behaviour

REGISTERED AS     {nmsig-attr}

networkId-behaviour   BEHAVIOUR

    DEFINED AS
```

This is the distinguishing attribute of the NMSIG network managed object class.

A.6.71 NMSIG Network Purpose

```
nmsig-networkPurpose   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  networkPurpose-behaviour

REGISTERED AS     {nmsig-attr}

networkPurpose-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies what the network is used for (e.g., manufacturing control, airline reservation, etc.)

A.6.72 NMSIG NPDU Time To Live

```
nmsig-nPDUTimeToLive        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  nPDUTimeToLive-behaviour

REGISTERED AS     {nmsig-attr}

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.6.73 NMSIG Octets Retransmitted Error Counter

```
nmsig-octetsRetransmittedErrorCounter   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  octetsRetransmitterErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

octetsRetransmitterErrorCounter-behaviour   BEHAVIOUR
```

DEFINED AS

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

A.6.74 NMSIG OS Info

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

REGISTERED AS   {nmsig-attr}

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

ProductInfo ::=  SEQUENCE {manufacturer  PrintableString,
                      productLabel  PrintableString,
                      release       PrintableString,
                      serialNumber  PrintableString}

osInfo-behaviour   BEHAVIOUR
```

DEFINED AS

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

A.6.75 NMSIG Open Connections

```
nmsig-openConnections          ATTRIBUTE
        WITH ATTRIBUTE SYNTAX INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  openConnections-behaviour

REGISTERED AS     {nmsig-attr}

openConnections-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of currently established transport connections.

A.6.76 NMSIG Out-Range Threshold

```
nmsig-outRangeThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  outRangeThreshold-behaviour

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

outRangeThreshold-behaviour   BEHAVIOUR
    DEFINED AS
```

This attribute specifies a threshold which is applied against the out-range length error rate. This rate is defined as the number of PDUs received with out-range length errors divided by the total number of PDUs received. A communication alarm notification is emitted when the out-range length error rate exceeds the threshold value.

A.6.77 NMSIG Outgoing Normal Disconnect Counter

```
nmsig-outgoingNormalDisconnectCounter        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  outgoingNormalDisconnectCounter-behaviour

REGISTERED AS     {nmsig-attr}
```

```
Count ::= {as defined in ISO Doc X}

outgoingNormalDisconnectCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of outgoing transport connections that were disconnected due to normal reasons.

A.6.78 NMSIG Packet Loss Rate

```
nmsig-packetLossRate   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  Gauge
        MATCHES FOR  Equality, Ordering
             BEHAVIOUR  packetLossRate-behaviour

REGISTERED AS     {nmsig-attr}

Gauge ::=  {as defined in ISO Doc X}

packetLossRate-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the rate of packets dropped per second.

A.6.79 NMSIG Packet Loss Rate Threshold

```
nmsig-packetLossRateThreshold   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  packetLossRateThreshold

REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

packetLossRateThreshold    BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the packet loss rate. A communication alarm notification is emitted when the packet loss rate exceeds the threshold value.

A.6.80 NMSIG PDU Too Long Threshold

```
nmsig-PDUTooLongThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  pDUTooLongThreshold-behaviour

REGISTERED AS      {nmsig-attr}

GaugeThreshold ::=  {as defined by ISO Doc X}

pDUTooLongThreshold-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the "PDU too long" error rate. The PDU too long error rate is defined as the number of PDUs received that were too long divided by the total number of PDUs received. A communication alarm notification is emitted when the "PDU too long" error rate exceeds the threshold value.

A.6.81 NMSIG PDUs Aborted Excessive Collisions Counter

```
nmsig-PDUsAbortedExcessiveCollisionsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsAbortedExcessiveCollisionsCounter-behaviour

REGISTERED AS    {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsAbortedExcessiveCollisionsCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of PDUs which were aborted by the underlying NMSIG IEEE 802.3 XMT resource due to excessive collisions.

A.6.82 NMSIG PDUs Aborted Excessive Collisions Threshold

```
nmsig-PDUsAbortedExcessColThreshold  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GaugeThreshold
        MATCHES FOR  Equality
             BEHAVIOUR  pDUsAbortedExcessColThreshold-behaviour
```

```
REGISTERED AS     {nmsig-attr}

GaugeThreshold ::=  {as defined in ISO Doc X}

pDUsAbortedExcessColThreshold-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a threshold which is applied against the PDUs aborted due to excessive collision rate. This rate is defined as the number of PDUs aborted due to excessive collision divided by the total number of PDUs transmitted. A communication alarm notification is emitted when the PDUs aborted due to excessive collision rate exceeds the threshold value.

A.6.83 NMSIG PDUs Alignment Error Counter

```
nmsig-PDUsAlignmentErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsAlignmentErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsAlignmentErrorCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of PDUs with an alignment error detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.84 NMSIG PDUs Excessive Deferral Counter

```
nmsig-PDUsExcessiveDeferralCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsExcessiveDeferralCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsExcessiveDeferralCounter-behaviour   BEHAVIOUR
```

```
    DEFINED AS

          This attribute specifies the number of PDUs for which the underlying NMSIG IEEE
          802.3 XMT resource detected excessive deferral.
```

A.6.85 NMSIG PDUs Discarded Counter

```
nmsig-PDUsDiscardedCounter        ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsDiscardedCounter-behaviour

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsDiscardedCounter-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of PDUs that were discarded by a network
          protocol layer entity.
```

A.6.86 NMSIG PDUs FCS Error Counter

```
nmsig-PDUsFCSErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsFCSErrorCounter-behaviour

REGISTERED AS      {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsFCSErrorCounter-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies the number of PDUs with an FCS error detected by the
          underlying NMSIG IEEE 802.3 RCV resource.
```

A.6.87 NMSIG PDUs Forwarded Counter

```
    nmsig-PDUsForwardedCounter         ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  pDUsForwardedCounter-behaviour

    REGISTERED AS     {nmsig-attr}

    Count ::= {as defined in ISO Doc X}

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

A.6.88 NMSIG PDUs In-Range Length Error Counter

```
nmsig-PDUsInRangeLengthErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  pDUsInRangeLengthErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsInRangeLengthErrorCounter-behaviour   BEHAVIOUR

    DEFINED AS

              This attribute specifies the number of PDUs with an in-range length error detected
              by the underlying NMSIG IEEE 802.3 RCV resource.
```

A.6.89 NMSIG PDUs Lost Internal MAC Xmt Error Counter

```
nmsig-PDUsLostInternalMACXmtErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
            BEHAVIOUR  pDUsLostInternalMACXmtErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsLostInternalMACXmtErrorCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of PDUs which were lost by the underlying NMSIG IEEE 802.3 XMT resource due to an internal MAC transmit error.

A.6.90 NMSIG PDUs Out-Range Error Counter

```
nmsig-PDUsOutRangeLengthErrorCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsOutRangeLengthErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsOutRangeLengthErrorCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of PDUs with an out-range length error detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.91 NMSIG PDUs Reassemble Fail Counter

```
nmsig-PDUsReasmblFailCounter         ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  pDUsReasmblFailCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

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.6.92 NMSIG PDUs Reassembled OK Counter

```
nmsig-PDUsReasmbldOKCounter         ATTRIBUTE
        WITH ATTRIBUTE SYNTAX Count
        MATCHES FOR   Equality, Ordering
```

```
            BEHAVIOUR  pDUsReasmbldOKCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsReasmbldOKCounter-behaviour   BEHAVIOUR

   DEFINED AS
```

This attribute specifies the number of PDUs that were reassembled successfully by a network protocol layer entity.

A.6.93 NMSIG PDUs Too Long Error Counter

```
nmsig-PDUsTooLongErrorCounter  ATTRIBUTE
      WITH ATTRIBUTE SYNTAX   Count
      MATCHES FOR   Equality, Ordering
          BEHAVIOUR  pDUsTooLongErrorCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

pDUsTooLongErrorCounter-behaviour   BEHAVIOUR

   DEFINED AS
```

This attribute specifies the number of PDUs with a "PDU too long" error detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.94 NMSIG Peripheral Names

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

REGISTERED AS     {nmsig-attr}

PeripheralNames ::=  SET OF AnyName

AnyName ::=  CHOICE {dn   DistinguishedName,
                     ps   PrintableString}
```

```
peripheralNames-behaviour   BEHAVIOUR

    DEFINED AS
          This attribute specifies the names of auxiliary devices.
```

A.6.95 NMSIG Product Info

```
nmsig-productInfo     ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   ProductInfo
        MATCHES FOR  Equality
             BEHAVIOUR  productInfo-behaviour

REGISTERED AS     {nmsig-attr}

ProductInfo ::=  SEQUENCE {manufacturer  PrintableString,
                      productLabel  PrintableString,
                      release       PrintableString,
                      serialNumber  PrintableString}

productInfo-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies product information of the underlying resource.
```

A.6.96 NMSIG Promiscuous Receive State

```
nmsig-promiscuousReceiveState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  State
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  promiscuousReceiveState-behaviour

REGISTERED AS   {nmsig-attr}

State ::=  ENUMERATED {off (0),
                   on (1)}

promiscuousReceiveState-behaviour   BEHAVIOUR

    DEFINED AS

          This attribute specifies the promiscuous receive state of the underlying NMSIG IEEE
          802.3 RCV resource.
```

A.6.97 NMSIG Remote Network Address

```
nmsig-remoteNetworkAddress       ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OCTET STRING
        MATCHES FOR  Equality
             BEHAVIOUR  remoteNetworkAddress-behaviour

REGISTERED AS      {nmsig-attr}

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.6.98 NMSIG Remote Transport Connection Endpoint

```
nmsig-remoteTransportConnectionEndpoint      ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OCTET STRING
        MATCHES FOR  Equality
             BEHAVIOUR  remoteTransportConnectionEndpoint-behaviour

REGISTERED AS      {nmsig-attr}

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.6.99 NMSIG Retransmission Timer Initial Value

```
nmsig-retransmissionTimerInitialValue  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  INTEGER
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  retransmissionTimerInitialValue-behaviour

REGISTERED AS     {nmsig-attr}

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.6.100 NMSIG Single Collision PDUs Counter

```
nmsig-singleCollisionPDUsCounter  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   Count
        MATCHES FOR   Equality, Ordering
             BEHAVIOUR  singleCollisionPDUsCounter-behaviour

REGISTERED AS     {nmsig-attr}

Count ::= {as defined in ISO Doc X}

singleCollisionPDUsCounter-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the number of single collision PDUs detected by the underlying NMSIG IEEE 802.3 XMT resource.

A.6.101 NMSIG Source Address Last Alignment Error PDU

```
nmsig-sourceAddrLastAlignmentErrorPDU  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   OCTET STRING
        MATCHES FOR  Equality
             BEHAVIOUR  sourceAddrLastAlignmentErrorPDU-behaviour

REGISTERED AS     {nmsig-attr}

sourceAddrLastAlignmentErrorPDU-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the source address of the last alignment error PDU detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.102 NMSIG Source Address Last FCS Error PDU

```
nmsig-sourceAddrLastFCSErrorPDU  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   OCTET STRING
        MATCHES FOR  Equality
             BEHAVIOUR  sourceAddrLastFCSErrorPDU-behaviour
```

```
REGISTERED AS     {nmsig-attr}

sourceAddrLastFCSErrorPDU-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the source address of the last FCS error PDU detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.103 NMSIG Source Address Last In-Range Length Error PDU

```
nmsig-sourceAddrLastInRangeLengthErrorPDU  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   OCTET STRING
        MATCHES FOR  Equality
            BEHAVIOUR  sourceAddrLastInRangeLengthErrorPDU-behaviour

REGISTERED AS     {nmsig-attr}

sourceAddrLastInRangeLengthErrorPDU-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the source address of the last in-range length error PDU detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.104 NMSIG Source Address Last Out-Range Length Error PDU

```
nmsig-sourceAddrLastOutRangeLengthErrorPDU  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   OCTET STRING
        MATCHES FOR  Equality
            BEHAVIOUR  sourceAddrLastOutRangeLengthErrorPDU-behaviour

REGISTERED AS     {nmsig-attr}

sourceAddrLastOutRangeLengthErrorPDU-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the source address of the last out-range length error PDU detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.105 NMSIG Source Address Last Too Long Error PDU

```
nmsig-sourceAddrLastTooLongErrorPDU  ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX   OCTET STRING
        MATCHES FOR  Equality
            BEHAVIOUR  sourceAddrLastTooLongErrorPDU

REGISTERED AS     {nmsig-attr}

sourceAddrLastOutRangeLengthErrorPDU-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the source address of the last "PDU too long" error PDU detected by the underlying NMSIG IEEE 802.3 RCV resource.

A.6.106 NMSIG System Id

```
nmsig-systemId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
            BEHAVIOUR  systemId-behaviour

REGISTERED AS     {nmsig-attr}

systemId-behaviour   BEHAVIOUR

    DEFINED AS
```

This is the distinguishing attribute of the NMSIG computer system managed object class.

A.6.107 NMSIG System Time

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

REGISTERED AS     {nmsig-attr}

systemTime-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the current time clocked at the computer system.

A.6.108 NMSIG Transport Connection Id

```
nmsig-transportConnectionId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  transportConnectionId-behaviour

REGISTERED AS     {nmsig-attr}

transportConnectionId-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute is the distinguishing attribute for the managed object class transportConnection.

A.6.109 NMSIG Transport Connection Profile Id

```
nmsig-transportConnectionProfileId  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX PrintableString
        MATCHES FOR  Equality
             BEHAVIOUR  transportConnectionProfileId-behaviour

REGISTERED AS     {nmsig-attr}

transportConnectionProfileId-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute is the distinguishing attribute for the managed object class nmsig-transportConnectionProfile.

A.6.110 NMSIG Transport Connection Reference

```
nmsig-transportConnectionReference      ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  OCTET STRING
        MATCHES FOR  Equality
             BEHAVIOUR  transportConnectionReference-behaviour

REGISTERED AS     {nmsig-attr}

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.6.111 NMSIG Transport Entity Type

```
nmsig-transportEntityType ATTRIBUTE
       WITH ATTRIBUTE SYNTAX TransportEntityType
       MATCHES FOR  Equality
            BEHAVIOUR  transportEntityType-behaviour

REGISTERED AS      {nmsig-attr}

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

transportEntityType-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the type of the transport protocol layer entity.

A.6.112 NMSIG User Friendly Label

```
nmsig-userFriendlyLabel  ATTRIBUTE
         WITH ATTRIBUTE SYNTAX PrintableString
         MATCHES FOR  Equality
              BEHAVIOUR  userFriendlyLabel-behaviour

REGISTERED AS      {nmsig-attr}

userFriendlyLabel-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies a user friendly name.

A.6.113 NMSIG Vendor Name

```
nmsig-vendorName   ATTRIBUTE
         WITH ATTRIBUTE SYNTAX   AnyName
         MATCHES FOR  Equality
```

```
            BEHAVIOUR  vendorName-behaviour

REGISTERED AS     {nmsig-attr}

AnyName ::=  CHOICE {dn   DistinguishedName,
              ps   PrintableString}

vendorName-behaviour   BEHAVIOUR

    DEFINED AS
```

This attribute specifies the name of a vendor.

A.6.114 NMSIG Xmt State

```
nmsig-XmtState  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX EnableState
        MATCHES FOR  Equality, Ordering
            BEHAVIOUR  xmtState-behaviour

REGISTERED AS   {nmsig-attr}

EnableState ::=  ENUMERATED {disable (0),
                       enable (1)}

xmtState-behaviour   BEHAVIOUR
    DEFINED AS
```

This attribute specifies whether the transmitting capability of the unserlying IEEE 802.3 resource is enabled or not. The 'enabled' and 'disabled' values of this attribute correspond to the 'enabled' and 'disabled' values of the OperationalState attribute of the IEEE 802.3 XMT managed object class. (This attribute was introduced as a GET-REPLACE attribute which can be used by management to enable or disable the transmitting capability of the underlying IEEE 802.3 resource.)

A.6.115 Object Class

Refer to [ISO Doc x] for the definition of this attribute.

A.6.116 Octets Received Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.117 Octets Sent Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.118 Operational State

Refer to [ISO Doc x] for the definition of this attribute.

A.6.119 Outgoing Connection Reject Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.120 Outgoing Connections Request Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.121 Outgoing Disconnect Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.122 Outgoing Temporal Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.123 PDUs Received Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.124 PDUs Sent Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.6.125 PDUs Retransmitted Error Counter

Refer to [ISO Doc X] for the definition of this attribute.

A.7 ACTIONS

This clause provides definitions of actions supported by managed object classes defined by the OSI MIB Working Group. Action definitions for managed object classes defined by other groups can be found in the document referenced under the managed object class definition in section 3.

A.7.1 NMSIG Execute Self Test

```
nmsig-executeSelfTest  ACTION
          ACTION BEHAVIOUR  selfTestBehaviour
          WITH RESULT SYNTAX  SelfTestResult

REGISTERED AS   {nmsig-action}

selfTestBehaviour  BEHAVIOUR

          DEFINED AS

             This action requests a self test sequence be executed on the referenced managed
             object instance. This action is always confirmed. The confirmation contains the
             operational state of the managed object under test following test completion, and
             optionally indicates the success or failure of the self test.

          SelfTestResult ::= SEQUENCE
                       {
                         operationalState  OperationalState,
                         testResult        BOOLEAN OPTIONAL
                        }
```

A.8 NOTIFICATIONS

This clause provides definitions of notifications emitted by managed object classes defined by the OSI MIB Working Group. Notification definitions for managed object classes defined by other groups can be found in the document referenced under the managed object class definition in section 3.

A.8.1 Attribute Change Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.2 Communication Alarm Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.3 Equipment Alarm Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.4 Environmental Alarm Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.5 NMSIG Counter Wrap Unconfirmed

```
nmsig-counterWrapUnconfirmed   NOTIFICATION
     BEHAVIOUR  counterWrap-behaviour
     WITH DATA SYNTAX  WrapInfo

REGISTERED AS  {notification}

counterWrap-behaviour   BEHAVIOUR

      DEFINED AS

           This notification indicates that a counter has wrapped.

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

A.8.6 Object Creation Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.7 Object Deletion Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.8 Processing Error Alarm Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.9 Relationship Change Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.8.10 State Change Unconfirmed

Refer to [ISO Doc x] for the definition of this notification.

A.9 REFERENCES

This clause lists the names of documents that were referenced in the earlier clauses.

Table of Contents

List of Figures

19 REMOTE DATABASE ACCESS

1 INTRODUCTION

Remote Database Access (RDA) specifies the communications service and protocol for accessing the capabilities of a database server from a client application. Figure 19.1 depicts RDA's placement within the application layer and its relationship to supporting OSI protocols:

```
+------------------------------------------+
|                                          |
|      Remote Database Access              |
|                                          |
+---------------+--------------------------+
|               |                          |
|  ASCE         |         TP               |
|               +-----+          +---------+
|                     |          |   CCR   |
+---------------------+----------+---------+
|                                          |
|              Presentation                |
|                                          |
+------------------------------------------+
|                                          |
|                Session                   |
|                                          |
+------------------------------------------+
```

Figure 1. Placement of RDA within the Application Layer..

This is an implementation agreement for RDA developed by the Implementors Workshop sponsored by the U.S. National Institute of Standards and Technology. This document addresses both the RDA generic model, service, and protocol, as well as the SQL Specialization, ISO 9579 parts 1 and 2, respectively. It is the intent of the workshop to expand this agreement to include other parts of 9579 as they are developed.

2 SCOPE

This implementation agreement addresses remote database interaction between a database server and a client application. The database server is an open system that provides database storage facilities and supplies database processing services to clients at other open systems.

The RDA communications service provides the protocol for RDA client interaction with an RDA server. The RDA client initiates an RDA dialogue and requests RDA operations to be performed by the RDA server on behalf of a client applications. The RDA server, located within the database server, provides database services to RDA clients.

More specifically, this document describes implementation agreements in the following areas:

1. the RDA generic model, service, and protocol,

2. the RDA SQL Specialization, and

3. SQL language restrictions.

3 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this International Standardized Profile. At the time of publication, the additions indicated were valid. All documents are subject to revision, and parties to agreements based on this International Standardized Profile are warned against automatically applying any more recent additions of the documents listed below, since the nature of references made by ISPs to such documents is that they may be specific to a particular addition. Members of IEC and ISO maintain registers of currently valid International Standards and ISPs, and CCITT maintains published additions of its current recommendations.

ISO 9579-1 Information Processing Systems - Open Systems Interconnection - Remote Database Access - Part 1: Generic Model, Service, and Protocol

ISO 9579-2 Information Processing Systems - Open Systems Interconnection - Remote Database Access - Part 2: SQL Specialization

ISO/IEC/TR10000-1:1990(E) Information Technology - Framework and Taxonomy of International Standardized Profiles - Part 1: Framework

Note: Work on ISO 9579 is ongoing.

4 DEFINITIONS

5 ABBREVIATIONS

6 RDA DIALOGUE STATE MODEL AGREEMENTS

7 GENERIC RDA AGREEMENTS

7.1 Functional Units

7.2 Optional Negotiable Facilities

7.2.1 Open/Close Within Transaction

7.3 Services

7.3.1 R-BeginDialogue

7.3.1.1 Optional Parameters

7.3.1.2 Parameter Restrictions

7.3.2 R-EndDialogue

7.3.3 R-Open

7.3.4 R-Close

7.3.5 R-Execute

7.3.6 R-Define

7.3.7 R-Invoke

7.3.8 R-Drop

7.3.9 R-BeginTransaction

7.3.10 R-Commit

7.3.11 R-Rollback

7.3.12 R-Status

7.3.13 Cancel

8 SQL SPECIALIZATION AGREEMENTS

8.1 Functional Units

8.2 Optional Facilites

8.3 Services

8.4 SQL Language Agreements

8.4.1 Language/Protocal Mapping

8.4.2 Implementor Defined Items

8.4.3 SQL Functional Restrictions

8.4.4 SQL State and Error Messages

8.5 Conformance Requirements

8.6 Recommended Practices

Table of Contents

List of Tables

20 Manufacturing Message Specification (MMS)

1 Introduction

This section defines Implementors Agreements based on Manufacturing Message Specification (MMS), as defined in ISO/IEC 9506. ISO/IEC 9506 has two parts. Part 1 defines the Virtual Manufacturing Device (VMD), its subordinate abstract objects, and the services on these objects. Part 2 defines the protocol. Future parts may define companion standards.

MMS, as described in ISO/IEC 9506, is based on the following ISO documents: ACSE Service and Protocol (ISO 8649, ISO 8650), Presentation Service and Protocol (ISO 8822, ISO 8823), ASN.1 Abstract Syntax Notation and Basic Encoding Rules (ISO 8824, ISO 8825), and Session Service and Protocol (ISO 8326, ISO 8327). These services and protocols are defined architecturally in the OSI Reference Model (ISO 7498). These agreements provide detailed guidance for the implementor, and eliminate ambiguities in interpretations.

An A-Profile based on MMS and these agreements can be used over any T-Profile (see ISO TR 10000) specifying the OSI connection-mode transport service, or the transport profiles used in support of MAP (Manufacturing Automation Protocol), TOP (Technical and Office Protocols), or US GOSIP.

2 Scope and Field of Application

2.1 General

Work on implementation agreements will proceed in phases based upon grouping of services and/or contexts. Implementations are not constrained from implementing services or contexts not addressed by the current set of stable agreements. Future phases of work may affect such implementations.
Phase 1 Agreements
These agreements will be based on a subset of MMS services and protocol listed in Table 1 and defined in ISO/IEC 9506-1 and ISO/IEC 9506-2.

Table 1 Phase 1 Services

```
Initiate
Conclude
Reject
Abort

Status
GetNameList
Identify
UnsolicitedStatus
GetCapabilityList

InitiateDownloadSequence
DownloadSegment
TerminateDownloadSequence
InitiateUploadSequence
UploadSegment
TerminateUploadSequence
DeleteDomain
GetDomainAttributes

Read
Write
InformationReport
GetVariableAccessAttributes

Input
Output

CreateProgramInvocation
DeleteProgramInvocation
Start
Stop
Resume
Reset
Kill
GetProgramInvocationAttributes
```

3 References

ISO/IEC 9506-1: 1990 Industrial automation systems - Manufacturing Message Specification Part 1: Service definition

ISO/IEC 9506-2: 1990 Industrial automation systems - Manufacturing Message Specification Part 2: Protocol specification

4 Definitions

The definitions given in ISO/IEC 9506-1 are applicable to this document.

In addition the following definitions are used in this document.

MMS Implementation - a term used to describe a system which conforms to ISO/IEC 9506, acting either as client or server, when it is unnecessary to distinguish between MMS-user and MMS-provider.

MMSpdu - Any valid value of MMSpdu abstract data type as defined in Clause 7 of ISO/IEC 9506-2, except for the initiate-RequestPDU, initiate-ResponsePDU, and initiate-ErrorPDU choices, encoded in the negotiated transfer syntax.

5 Errata

None at time of publication.

6 Status

6.1 Status of Phase 1 Agreements

Phase 1 is in progress.

7 General Agreements

7.1 Max Supported PDU Size

The max_mms_pdu_size is defined as the maximum number of octets in an MMSpdu encoded using the negotiated transfer syntax. This size shall apply to all MMSpdu's with the exception of the initiate-Request PDU, initiate-Response PDU, and initiate-Error PDU. The max_mms_pdu_size shall be negotiated during connection initiation using the Local Detail Calling and Local Detail Called parameters of the MMS initiate service.

The negotiated max_mms_pdu_size shall be applied as follows:

o Any received MMSpdu which is less than or equal to the negotiated max_mms_pdu_size shall be properly parsed and processed.

- o An MMS implementation should not send an MMSpdu whose size exceeds the negotiated max_mms_pdu_size. If an MMS implementation sends an MMSpdu that exceeds the negotiated max_mms_pdu_size, then it shall be prepared to receive a reject pdu. Should an MMS implementation receive an MMSpdu that exceeds the negotiated max_mms_pdu_size, it shall either reject the MMSpdu or accept the MMSpdu as if no size violation had occurred and perform the expected processing.

- o If an MMS implementation is unable to send a service response because the response would exceed the max_mms_pdu_size, then it shall return a Service response (-) with an error class of SERVICE and an error code of OTHER.

- o When rejecting an MMSpdu because it exceeds the negotiated max_mms_pdu_size, an MMS implementation shall use a Reject PDU Type of PDU-ERROR and a Reject Code of INVALID-PDU in the resulting reject pdu.

7.2 FileName

Restrictions for the use of the type FileName in the MMS Abstract Syntax are specified in section 9.1 of part 9 of these agreements.

8 Service-Specific Agreements

8.1 Environment and General Management

8.1.1 Initiate

8.1.1.1 Negotiation of MMS Abstract Syntaxes

On the A-ASSOCIATE response, the MMS responder shall not accept more than one presentation context derived from an MMS abstract syntax. For this agreement, the term 'MMS abstract syntax' shall represent an abstract syntax from the set containing the abstract syntax defined in clause 19 of ISO/IEC 9506-2 and abstract syntaxes defined by MMS companion standards.

Tutorial Note:

There are technical problems with describing operation in multiple MMS abstract syntaxes over a single association. These problems have been identified as of 9/90, and form the basis of the prior agreement.

8.1.1.2 Max Serv Outstanding

An MMS Implementation which intends to conform only with the Client Conformance Requirements for Requester CBBs shall:

1. propose one or greater for the value of the Proposed Max Serv Outstanding Called parameter in the Initiate service when initiating the application association (calling).

2. offer one or greater for the value of the Negotiated Max Serv Outstanding Calling parameter in the Initiate service when receiving the application association initiation (called).

An MMS Implementation which intends to conform to one or more Server Conformance Requirements for Responder CBBs shall:

1. propose one or greater for the value of the Proposed Max Serv Outstanding Calling parameter in the Initiate service when initiating the application association (calling).

2. offer one or greater for the value of the Negotiated Max Serv Outstanding Called parameter in the Initiate service when receiving the application association initiation (called).

8.1.1.3 Local Detail Calling

The local detail calling parameter in the initiate request primitive shall specify the max_mms_pdu_size guaranteed to be supported by the calling MMS implementation. If the local detail calling parameter is absent from the request primitive, then the calling MMS implementation guarantees support for an unlimited max_mms_pdu_size.

If present in the request or indication primitives, the local_detail_calling parameter shall not be less than 64; however, it is recommended that at least 512 octets be supported.

8.1.1.4 Local Detail Called

The local detail called parameter in the initiate response shall specify the negotiated max_mms_pdu_size for the application association.

If the local detail calling parameter was omitted in the indication primitive, then the local_detail_called parameter:

1. may be omitted from the response, indicating that the calling MMS implementation and the called MMS implementation are prepared to support an unbounded max_mms_pdu_size, or,

2. may be specified in the response, indicating a requirement to support the specified value for max_mms_pdu_size.

If the local detail calling parameter was included in the request, then this parameter shall be present in the response and its value shall be less than or equal to the value of the local detail calling parameter of the request.

If present in the response, the local detail called parameter shall not be less than 64; however, it is recommended that at least 512 octets be supported.

8.2 VMD Support

8.3 Domain Management

8.3.1 List of Capabilities

Only one capability shall be described in each Visible String of the SEQUENCE OF.

8.3.2 Initiate Download Sequence Service

The List of Capability parameter shall follow the limitations of 8.3.1.

The syntax and semantics of the capabilities shall be defined by the Server in the PICS. Any deviation from the defined syntax and semantics shall be grounds for the Server to return a service error with Error Class = RESOURCE and Error Code = CAPABILITY-UNKNOWN.

8.3.3 Download Segment Service

A client that receives a Download Segment indication after issuing a Download Segment Result(+) with the MoreFollows parameter equal to FALSE or after issuing a Download Segment Result(-) shall issue either a service error, specifying an Error Class = SERVICE and an Error Code = PRIMITIVES-OUT-OF-SEQUENCE, or an Abort request.

8.3.4 Terminate Download Sequence Service

If a client receives a Terminate Download Sequence indication in which the Discard parameter is absent and the client has not issued a Download Segment response with the More Follows parameter = FALSE for that Domain, it shall behave as if it had received a Terminate Download Sequence indication with the Discard parameter present with error class = VMD-STATE and error code = DOMAIN-TRANSFER-PROBLEM. It is then up to the client application to determine the true state of the Domain and take any recovery action.

8.3.5 Initiate Upload Sequence Service

The List of Capability parameter shall follow the limitations of 8.3.1.

8.3.6 Upload Segment Service

A server that receives an Upload Segment indication for an Upload State Machine for which it has issued an Upload Segment Result(-) or an Upload Segment Result(+) with the MoreFollows parameter equal to FALSE, shall issue either a service error, specifying an Error Class = SERVICE and an Error Code = PRIMITIVES-OUT-OF-SEQUENCE, or an Abort request.

8.3.7 Get Domain Attributes Service

The List of Capability parameter shall follow the limitations of 8.3.1.

8.3.8 Get Capability List Service

The List of Capability parameter shall follow the limitations of 8.3.1.

8.4 Program Invocation Management

8.4.1 Start Service

A ProgramInvocationState of non-existent shall be returned in a Result(-) when a request to Start a non-existent Program Invocation is received.

8.4.2 Stop Service

A ProgramInvocationState of non-existent shall be returned in a Result(-) when a request to Stop a non-existent Program Invocation is received.

8.4.3 Resume Service

A ProgramInvocationState of non-existent shall be returned in a Result(-) when a request to Resume a non-existent Program Invocation is received.

8.4.4 Reset Service

A ProgramInvocationState of non-existent shall be returned in a Result(-) when a request to Reset a non-existent Program Invocation is received.

8.5 Variable Access

8.5.1 Scattered Access

It is strongly recommended that for services which use variable access, a Variable List Name or List of Variable be used instead of Scattered Access.

No implementations shall be required to propose or accept the VSCA Parameter CBB.

8.5.2 Floating Point

It is stongly recommended for services which use floating point types or values, that the choice of floating-point in the Data and TypeSpecification productions be used instead of the real choice.

No implementations shall be required to propose or accept the REAL parameter CBB.

8.6 Semaphore Management

Semaphore services are not considered in Phase 1.

8.7 Operator Communication

No Operator Communication agreements have been identified to date.

8.8 Event Management

Event Management services are not considered in Phase 1.

8.9 Journal Management

Journal Management services are not considered in Phase 1.

A Backwards Compatibility Agreements

A.1 Introduction

There is an installed base of real DIS 9506 based implementations. Providing support for application connectivity to both DIS and IS is desired as a migration strategy. These implementation agreements will allow IS based implementations to interoperate with DIS based implementations as described in ANNEX B. To achieve this backwards compatibility, the IS implementation shall support all of the agreements in this section.

It was found that the Abstract Syntax name object identifiers of both DIS and IS were identical. Therefore, the use of zero as the version number allows differentiation between an IS and a DIS based implementation. Since the abstract syntax name object identifier of any companion standard is different from that used by the DIS implementations, DIS implementations cannot interoperate with IS based impelementations in a companion standard context.

Note: The value of zero is a valid value for this parameter in the DIS and not in the IS.

Tutorial Note:

There are three types of implementations when considering MMS backwards compatibility.

IMP-1: An implementation based on DIS 9506 as described in Annex B.

IMP-2: An implementation based on IS 9506 with no backwards compatibilityagreements applied.

IMP-3: An implementation based on IS 9506 which includes the backwards compatibility agreements of this annex. Such an implementation can dynamically negotiate to interoperate with an IMP-1, an IMP-2, or an IMP-3 implementation.

Since the value of the minor version number is zero for DIS-based implementations, and is one or greater for implementations of ISO/IEC 9506, this value can be used to differentiate between IMP-1 and IMP-2. An IMP-3 system can interoperate with either of these systems. If an IMP-3 is the Calling system, it will offer a value of one (or greater) for the proposed version number. An IMP-1 system will respond with a value of the negotiated version number of zero, using the negotiation procedure defined in ISO/IEC 9506. The IMP-3 system will accept this response. If the IMP-3 system is the Called system and has received an Initiate request with a value of zero for the proposed version number (from an IMP-1 system), it will respond with a value of zero for the negotiated version number. By following this procedure, an IMP-3 can interoperate with an IMP-2 or with another IMP-3 viewed as an IMP-2. After association context establishment, an IMP-3 system shall behave as either an IMP-1 or an IMP-2 system as appropriate on that particular association. The remainder of this section describes additional agreements which change an IMP-2 implementation into an IMP-3 implementation.

A.2 Backwards Compatibility Agreements for Calling MMS Implementations

A calling MMS implementation shall be capable of receiving and supporting a negotiatedVersionNumber parameter in the Initiate Service confirm of zero.

A calling MMS implementation which has received a negotiatedVersionNumber parameter in the Initiate Service confirm of zero shall support the modifications described in section A.3.

A calling MMS implementation shall be capable of receiving an Application Context Name parameter value appropriate to an IMP-1 or IMP-2 in the A-Associate confirm.

A calling MMS implementation which has received a negotiatedVersionNumber of zero shall be capable of receiving and supporting an InitiateResponse which has been encoded according to the modifications described in Appendix B, specifically the capability of receiving and supporting a negotiatedParameterCBB containing exactly 7 bits.

If a calling MMS implementation receives an Initiate confirm primitive with a negotiatedVersionNumber parameter equal to zero, the calling MMS implementation shall support the VLIS conformance building block if the implementation claims support for any service which contains one or more parameters which indicate the VLIS CBB in this service definition.

A.3 Backwards Compatibility Agreements for Called MMS Implementations

A called MMS implementation shall be capable of receiving and supporting a proposedVersionNumber parameter in the Initiate Service indication of zero.

A called MMS implementation which has received a proposedVersionNumber parameter in the Initiate Service indication of zero shall support the modifications in section A.3.

A called MMS implementation shall be capabile of receiving an Application Context Name parameter appropriate to an IMP-1 or IMP2 in the A-Associate indication.

A called MMS implementation shall be capable of receiving and supporting an InitiateRequest which has been encoded according to the modifications described in Appendix B, specifically the capability of receiving and supporting a proposedParameterCBB containing exactly 7 bits.

If a called MMS implementation receives an Initiate indication primitive with a proposedVersionNumber parameter equal to zero, the called MMS implementation shall support the VLIS conformance building block if the implementation claims support for any service which contains one or more parameters which indicate the VLIS CBB in this service definition.

A.4 General Backwards Compatibility Agreements

A.4.1 VMD Logical Status

If the current VMD State is SUPPORT-SERVICES-ALLOWED and the association minor version number is zero, then the vmdLogicalStatus parameter shall have a value of STATE-CHANGES-ALLOWED in a Status response or in an unsolicitedStatus request.

A.4.2

Further agreements are required to complete this section.

B DIS 9506 Modifications Required For Backwards Compatibility

(Refer to the Stable Agreements, dated September, 1990.)

Editor's Note: The text of the entire working document has been marked as "pending stable".

21. Character Set Usage in OSI Applications

This International Standardized Profile is defined within the context of Functional Standardization, in accordance with the principles specified by ISO TR 10000, "Taxonomy Framework and Directory of Profiles." The context of Functional Standardization is one part of the overall field of Information Technology (IT) standardization activities, covering base standards, profiles, and registration mechanisms. A profile defines a combination of base standards that collectively perform a specific well-defined IT function. Profiles standardize the use of options and other variations in the base standards, and provide a basis for the development of uniform, internationally recognized system tests.

This International Standardized Profile was developed in close cooperation between the three International OSI Workshops: the NIST OSI Implementors Workshop (NIST OIW), the European Workshop for Open Systems (EWOS), and the AsiaOceania Workshop (AOW). The text is harmonized between these three Workshops and was ratified by the Workshops' plenary assemblies.

This International Standardized Profile contains an informative Annex A - Character Set Technology.

21.1. Scope

This International Standardized Profile describes Information Processing Character Set agreements covering character set usage in referencing Application Service Elements and OSI Applications. These agreements are based upon ISO Character Set International Standards and CCITT Character Set Recommendations. The informative Annex A summarizes the character set practices within referencing Application Service Elements and OSI Applications including all relevant encoding information drawn from the appropriate ISO Registers, ISO Standards, and CCITT Recommendations.

21.1.1. Recording Additional Character Sets

This International Standardized Profile does not prevent Application Service Elements from adding new graphic character sets or control function sets. When new character sets are to be added, however, they shall be recorded in this International Standardized Profile.

21.1.2. General Applicability of Character Sets

For the purpose of this International Standardized Profile when new character sets are to be added, efforts shall be made to obtain agreement on their uses among Application Service Elements so that they are generally applicable.

21.1.3. Minimum Number of Character Sets

The number of character sets supported will be kept to the minimum possible so as to maximize interoperability.

21.2. References

The following International Standards and CCITT Recommendations are referenced in this International Standardized Profile:

International Information Exchange for Videotex, CCITT Recommendation T.100, 1985.

International Alphabet No. 5, CCITT Recommendation T.50, 1985.

Coded Character Sets for Telematic Services, CCITT Recommendation T.51, 1985.

Character Repertoire and Coded Character Sets for the International Teletex Service, CCITT Recommendation T.61, 1985.

Information processing — 8-bit single-byte coded graphic character sets — Part 7: Latin/Greek alphabet, DIS 8859-7, 1987.

Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques, IS 2022, 1986.

Data processing — Procedure for registration of escape sequences, IS 2375, 1985.

Information processing — ISO 8-bit code for information interchange — Structure and rules for implementation, IS 4873, 1986.

Information Processing — ISO 7-bit and 8-bit coded character sets — Additional control functions for character-imaging devices, IS 6429, 1983.

Information Processing — ISO 7-bit coded character set for information interchange, IS 646, 1983.

Information processing — Coded character sets for text communication — Part 1: General introduction, IS 6937/1, 1983.

Information Processing — Coded character sets for text communication — Part 2: Latin alphabetic and non-alphabetic graphic characters, IS 6937/2, 1983.

Text Communication — Registration of graphic character subrepertoires, IS 7350, 1984.

Information Processing Systems — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1), IS 8824, 1987.

Information Processing Systems — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), IS 8825, 1987.

Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1, IS 8859-1, 1987.

International Register of Coded Character Sets to be Used With Escape Sequences, International Register of Coded Character Sets, 1989.

21.3. Definitions

21.3.1. character data:

Character data is defined to be graphic characters and control functions as defined by ISO 2022 and the appropriate International Standards.

21.3.2. composite graphic symbol:

A composite graphic symbol is defined for the purposes of this International Standardized Profile as a non-spacing diacritical in combination with an alphabetic as in ISO 6937.

21.4. Abbreviations

21.4.1. ASN.1:

ASN.1 is an abbreviation for Abstract Symbolic Notation One.

21.4.2. IRV

IRV is an abbreviation for International Reference Version.

21.5. Position within the Taxonomy

<<The formal position of this International Standardized Profile within the taxonomy is currently unknown.>>

It may be referenced from the ISP for any application service element or OSI application.

21.6. Conformance

Implementations claiming conformance to this ISP must designate one or more of the Character Set Profiles defined herein.

Imaging of Graphic Characters is not required by this ISP. Imaging conformance may be defined in the specific Upper Layers Requirements section of the referencing ISP. If no imaging requirements are specified, then there are no conformance requirements.

21.6.1. Processed Character Data

Processed character data is character data which must be processed by the Application Service Element or OSI Application, for example, store and forward character data.

Senders of character data must not produce invalid character codes or invalid designating or invoking escape sequences.

21.6.1.1. Non-supported Character Sets

If an implementation receives a designating escape sequence for a character set that it is not able to interpret, then it shall regard that sequence as "invalid data." If possible, it will signal this error in a way that is appropriate to the protocol definition. For applications for which there is no protocol, then no error need be returned. It will not be required to interpret any following characters that are within that data item.

21.6.1.2. Reserved Character Codes

If an implementation receives a coded character that is specified in the standard to be "reserved for future standardization," it shall not be considered an error. An imaging device shall indicate receipt of such a reserved character to the user in am implementation defined way, e.g. by making available a character that need not be distinguishable from one of the other characters specified in the standard.

If receivers reject or discard invalid character codes, an appropriate error code must be returned.

21.6.1.3. Validation of Character Codes

Character codes within the scope of a standard for which there is no definition in the code table are defined to be invalid character codes. An invalid escape sequence is any designating or invoking escape sequence which is not defined in these agreements.

Implementations must conform to the following statement.

- Originators of data shall not produce invalid character codes or invalid designating or invoking escape sequences.

21.6.2. Unprocessed Character Data

Unprocessed character data is character data which is not processed by the Application Service Element or OSI Application, for example, character matching.

21.6.2.1. Validation of Character Codes

Character codes within the scope of a standard for which there is no definition are defined to be invalid character codes. An invalid escape sequence is any designating or invoking escape sequence which is not defined in these agreements.

Implementations must conform to the following statements.

- Receivers need not validate character codes or designating or invoking escape sequences.
- Senders who do not originate data need not validate character codes.

21.7. General Agreements

The agreements recorded in this section cover all character set usage except where explicitly noted to the contrary. Additional agreements specific to individual character sets are recorded in the individual character set profiles.

21.7.1. Encoding

The following agreements cover various aspects of character encoding.

21.7.1.1. Overprint, Composite Characters

A composite graphic symbol is considered as one character for purposes of comparison and character string length computation.

With the exception of composite graphic symbols, sequences of graphic characters and control functions which would result in the presentation of two or more graphic characters in a single character position shall not be used. So for example, the sequence "a BACKSPACE ¨" must be processed as three characters rather than as the single character ä.

21.7.1.2. Code Extension Facilities for GeneralString and GraphicString

This section constitutes the prior agreement on code extension required by ISO 2022.

For ASN.1 GeneralString and GraphicString types, the assumed extension facilities are as though the following escape sequences from ISO 2022 have been applied: ESC 2/0 4/3, ESC 2/0 4/9, and ESC 2/0 5/10. These sequences indicate:

- 8-bit environment;
- the G0, and G1 graphic sets shall be used;
- the designating escape sequences also invoke the G0 and G1 sets into columns 02 to 07 and 10 to 15 respectively;
- no locking shift functions shall be used;
- the graphic character sets may comprise 94 and/or 96 characters,
- a G2 set shall be used; and,
- characters from G2 may be accessed by use of the single-shift 2 control function.

Designating ESCAPE sequences in a data stream are permitted. No Announcers of extension facilities may be used within these ASN.1 string types.

21.7.1.3. Initial Conditions for TeletexString

For TeletexString (T61String) the initial condition is described in CCITT T.61 Annex A, Clause A.2.

21.7.2. Comparisons

This section contains agreements concerning comparison of characters during processing.

21.7.2.1. Matching Characters

A character submitted for matching with another character does not have to be drawn from the same coded character set. However, the match is restricted to characters taken from any pair of coded character sets for which equality or inequality is defined. The identifications of such pairs of coded character sets are shown in the following list. The result of comparing characters from a pair of different coded character sets not in this list is undefined.

(ISO 646,	ISO 6937-2)
(ISO 646,	ISO 8859-1)
(ISO 6937-2,	ISO 8859-1)

Character matching is defined for characters, not their coded representations. The character must take into account any code extension techniques. For example, the character named "SMALL LETTER a WITH DIAERESIS" of ISO 8859 must match the character named "small a with diaeresis or umlaut mark" of ISO 6937 even though the former character is encoded in a single octet and the latter in two octets.

Two characters are said to be equal if, and only if, their names are identical. The names are recorded in the registration of the character sets in the **International Register of Coded Character Sets to be used with Escape Sequences** and not the character set International Standard or Recommendation.

In the case of ISO 6937-2 the names of the composite graphic symbols are specified in the standard itself. However in the present edition there are some systematic differences between the naming conventions used in the standard and those used in the ISO Character Set Register as shown below:

ISO 6937 name:	capital A with acute accent
ISO Register Name:	CAPITAL LETTER A WITH ACUTE ACCENT

In this case, two characters are equal if, and only if, their names differ only by the inclusion of the word LETTER in the ISO Register Name. For those characters whose names do not follow this convention, the following list defines the match:

ISO 6937 Name	ISO Register Name

<<Editor's Note: to be filled in>

If a character set registration does not provide character names then matching will be defined by exact matching on an octet by octet basis.

<<Editor's Note: The problem of matching Oriental language character sets is for further study.>>

In comparing strings all control functions except code designation and invocation extension facilities shall be ignored. SPACE is treated as a graphic character in such comparisons.

In comparing strings when a character code is encountered for which no other match is defined, matching will be defined by exact matching on an octet by octet basis.

21.7.2.2. CaseIgnore Comparisons

In character comparisons in which case is ignored, the matching rules of clause 21.7.2.1 are relaxed in that the characters are equal if their names as defined in clause 21.7.2.1 differ only by one name having SMALL where the other name has CAPITAL.

21.7.2.3. Ordering and Comparing Characters

An agreement on comparison, other than equality or inequality, between characters requires a definition of a collating sequence. This document contains no such agreements.

The collating sequence of letters, accented letters and other graphic symbols is not currently defined in any International Standard or Recommendation.

Preferred collating sequences might vary between countries.

21.7.2.4. Comparing Encoded ASN.1 Character Strings

In this section a character string is considered to be a sequence of characters some of which may be composed of multiple bytes depending upon the character set encodings which are specified. Comparing two character strings gives the same result independent of each character string's encoding, for example, the comparison is independent of the Basic Encoding Rules for ASN.1:

- as constructed or primitive form, or,
- as definite or indefinite length form.

21.8. Character Set Profiles

A Character Set Profile summarizes implementation agreements specific to a particular character set. Character Set Profiles are identified in the following manner:

CSn-m

where:
CS means Character Set
n = 1 designates a profile for a graphic character set
n = 2 designates a profile for a control function set
m is a number uniquely identifying the Character Set Profile.

The values of n and m are defined in this agreement. Names of Character Set Profiles are also defined in this International Standardized Profile.

This section covers agreements about Character Set Standards and Recommendations including:

- subrepertoires supported,
- standardized options selected,
- component character sets and their registrations in the **International Register of Coded Character Sets to be used with Escape Sequences** where there is a choice to be made, or the standard does not specify it, and,
- the designation of component character sets within the ISO 2022 Code Extension Model where there is a choice to be made.

The General Agreements of the preceding section apply to each of these Character Set Profiles.

21.8.1. CS1-1 ISO 646 Graphic Character Set

21.8.1.1. Base Standard

International Standard 646 - 1983, *Information Processing — ISO 7-bit coded character set for information interchange.*

<<Editor's Note: These agreements will be based on the new DIS 646.>>

21.8.1.2. Subrepertoire or Version

International Reference Version

21.8.1.3. Standard Options Selected

Composite graphic symbols are covered by General Agreements.

21.8.1.4. Character Set Components and Designated Position

IRV of ISO 646 number 2 in G0

<<Editor's Note: This will change to number 6.>>

Space is in 2/0

21.8.1.5. Other Agreements

None.

21.8.2. CS1-2 JIS X0208

<<Editor's Note: to be defined.>>

21.8.3. CS1-3 CCITT Recommendation T.61 Graphic Character Sets Basic Teletex Profiles

21.8.3.1. Base Standard

CCITT Recommendation T.61 - 1985, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

<<Editor's Note: These references will be updated as soon as the 1989 versions are published.>>

21.8.3.2. Subrepertoire or Version

None

21.8.3.3. Standard Options Selected

None

21.8.3.4. Character Set Components and Designated Position

Teletex Primary Graphic Set 102 in G0

Teletex Supplementary Graphic Set 103 in G2

SPACE in 2/0

21.8.3.5. Other Agreements

Support for CCITT Recommendation T.61 as an ASN.1 GeneralString is outside of this International Standardized Profile.

Support of the graphic set components of T.61 as an ASN.1 GraphicString is outside the scope of this International Standardized Profile.

Use of CCITT Recommendation T.61 except where mandated by standards is outside the scope of this International Standardized Profile. Exceptions to this rule for specific Application Service Element protocol elements must be documented by the referencing Application Service Elements or OSI Applications.

21.8.4. CS1-4 ISO 8859-1 Latin Alphabet No. 1

21.8.4.1. Base Standard

International Standard 8859-1 - 1987, *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*

21.8.4.2. Subrepertoire or Version

Not applicable.

21.8.4.3. Standard Options Selected

Not applicable.

21.8.4.4. Character Set Components and Designated Position

ASCII Graphic Character Set number 6 in G0

Right hand part of Latin Alphabet No. 1 number 100 in G1

21.8.4.5. Other Agreements

None.

21.8.5. CS1-5 ISO 6937-2 Coded Character Sets for Text Communication

21.8.5.1. Base Standard

International Standard 6937/2 - 1983, *Information Processing — Coded character sets for text communication — Part 2: Latin alphabetic and non-alphabetic graphic characters.*

<<Editor's Note: Includes Addendum 1 as soon as it is published.>>

21.8.5.2. Subrepertoire or Version

Full number 0

Minimum number 1

Teletex number 3

Western European Data Processing number 9

21.8.5.3. Standard Options Selected

Not applicable

21.8.5.4. Character Set Components and Designated Position

IRV of ISO 646 number 2 in G0

<<Editor's Note: This will change to number 6.>>

Supplementary set of Latin Text Processing number 142 in G2

21.8.5.5. Other Agreements

For subrepertoires 2 and 5, the supplementary set may be omitted at the discretion of the sending application.

21.8.6. CS1-6 ISO 8859/7 Greek Supplementary Set

<<Editor's Note: to be defined.>>

21.8.7. CS1-7 CCITT Recommendation T.61 Graphic Character Sets Basic Teletex Profiles (1984)

21.8.7.1. Base Standard

CCITT Recommendation T.61 - 1981, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

21.8.7.2. Subrepertoire or Version

None

21.8.7.3. Standard Options Selected

None

21.8.7.4. Character Set Components and Designated Position

Teletex Primary Graphic Set 102 in G0

Teletex Supplementary Graphic Set 103 in G2

SPACE in 2/0

21.8.7.5. Other Agreements

Support for CCITT Recommendation T.61 as an ASN.1 GeneralString is outside of this International Standardized Profile.

Support of the graphic set components of T.61 as an ASN.1 GraphicString is outside the scope of this International Standardized Profile.

Use of CCITT Recommendation T.61 except where mandated by standards is outside the scope of this International Standardized Profile. Exceptions to this rule for specific Application Service Element protocol elements must be documented in the referencing Application Service Elements or OSI Applications.

This profile is intended for use with the X.400-1984 implementation agreements only.

21.8.8. CS 1-8 CCITT Recommendation T.61 Graphic Character Sets

<<Editor's Note: to be defined.>>

21.8.9. Korean National Character Set

<<Editor's Note: to be defined.>>

21.8.10. CS2-1 ISO 646 C0 Control Functions

21.8.10.1. Base Standard

International Standard 646 - 1983, *Information Processing — ISO 7-bit coded character set for information interchange.*

21.8.10.2. Subrepertoire or Version

None.

21.8.10.3. Standard Options Selected

None.

21.8.10.4. Character Set Components and Designated Position

ISO 646 C0 Set number 1 in C0

DELETE in 7/15

21.8.10.5. Other Agreements

When a single format effector for vertical (or horizontal) movement is optionally permitted to effect a combined vertical and horizontal movement, implementations shall not use this single format effector for effecting the combined vertical and horizontal movement.

21.8.11. CS2-2 ISO 6429 Additional Control Functions

21.8.11.1. Base Standard

International Standard 6429 - 1983, *Information Processing — ISO 7-bit and 8-bit coded character sets — Additional control functions for character-imaging devices.*

21.8.11.2. Subrepertoire or Version

None.

21.8.11.3. Standard Options Selected

None.

21.8.11.4. Character Set Components and Designated Position

C1 Control Set of ISO 6429-1983 number 77 in C1

21.8.11.5. Other Agreements

None.

21.8.12. CS2-3 CCITT Recommendation T.61 Control Sets

21.8.12.1. Base Standard

CCITT Recommendation T.61 - 1985, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

<<Editor's Note: These references will be updated as soon as the 1989 versions are published.>>

21.8.12.2. Subrepertoire or Version

None.

21.8.12.3. Standard Options Selected

Teletex optional repertoire of control functions is not selected.

21.8.12.4. Character Set Components and Designated Position

Teletex Primary Set of Control Functions number 106 in C0

Teletex Supplementary Set of Control Functions number 107 in C1

21.8.12.5. Other Agreements

None.

21.8.13. CS2-4 CCITT Recommendation T.61 Control Sets (1984)

21.8.13.1. Base Standard

CCITT Recommendation T.61 - 1981, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

21.8.13.2. Subrepertoire or Version

None.

21.8.13.3. Standard Options Selected

Teletex optional repertoire of control functions is not selected.

21.8.13.4. Character Set Components and Designated Position

Teletex Primary Set of Control Functions number 106 in C0

Teletex Supplementary Set of Control Functions number 107 in C1

21.8.13.5. Other Agreements

This profile is intended for use with the X.400-1984 implementation agreements only.

Annex A

Character Set Technology

(This Annex does not form part of these agreements.)

A.1. Introduction

This Annex presents information from Information Processing Character Set Standards which is relevant to the implementation of OSI Services. The intent is to collect into one place the most relevant information for implementors from character set standards specified in OSI and OSI related standards.

A.2. Scope

Material in this Annex is drawn from ISO and CCITT Character Set standards and Recommendations. Topics covered include Character Set Extension Techniques and Character Set Encodings. ASN.1 Basic Encoding Rules are reviewed also. Rationale for the implementation agreements in the ISP is provided where appropriate.

A.3. Field of Application

This annex covers character set information for ASN.1 Basic Encoding Rules as used by OSI services. It also includes information pertaining to OSI Interchange Formats such as Office Document Architecture.

A.4. Character Set Standards

The following character set standards have some relevance to this material.

International Information Exchange for Videotex, CCITT Recommendation T.100, 1985.

International Alphabet No. 5, CCITT Recommendation T.50, 1985.

Coded Character Sets for Telematic Services, CCITT Recommendation T.51, 1985.

Character Repertoire and Coded Character Sets for the International Teletex Service, CCITT Recommendation T.61, 1985.

Information processing — 8-bit single-byte coded graphic character sets — Part 7: Latin/Greek alphabet, DIS 8859-7, 1987.

Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques, IS 2022, 1986.

Data processing — Procedure for registration of escape sequences, IS 2375, 1985.

Information processing — ISO 8-bit code for information interchange — Structure and rules for implementation, IS 4873, 1986.

Information Processing — ISO 7-bit and 8-bit coded character sets — Additional control functions for character-imaging devices, IS 6429, 1983.

Information Processing — ISO 7-bit coded character set for information interchange, IS 646, 1983.

Information processing — Coded character sets for text communication — Part 1: General introduction, IS 6937/1, 1983.

Information Processing — Coded character sets for text communication — Part 2: Latin alphabetic and non-alphabetic graphic characters, IS 6937/2, 1983.

Text Communication — Registration of graphic character subrepertoires, IS 7350, 1984.

Information Processing Systems — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1), IS 8824, 1987.

Information Processing Systems — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), IS 8825, 1987.

Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1, IS 8859-1, 1987.

International Register of Coded Character Sets to be Used With Escape Sequences, International Register of Coded Character Sets, 1989.

A.5. Introduction to Character Set Standards

A brief introduction to reading a character set standard is presented here for the uninitiated. Most of the character set standards described in this Annex use the term "bit combinations" to refer to the ordered string of bits which compose a character. Most implementations of these standards allocate an 8-bit byte to a character and consequently tend to intermix the notions of bytes and characters. In the OSI environment, 8-bit bit combinations are normally referred to as "octets."

A character set standard generally presents its character encodings in a table composed of 16 rows and 8 or 16 columns depending on whether a 7-bit or an 8-bit character set is being defined. A given character code is generally referenced by naming its column and then its row. Thus in ISO 646 the capital letter A is referred to as 4/1. Some standards precede single digits with a zero so that in ISO 8859/1 the capital letter A is referred to as 04/01. This positional notation is especially important in the consideration of the code extension techniques. Code extension techniques describe characters in terms of their position only, without regard for any possible previously assigned interpretations.

A.6. Definitions

The following definitions drawn from relevant character set standards are provided to assist in understanding the material in this annex. These definitions were drawn from International Standards which were current at the time of drafting this document. Any conflict between these definitions and those of the relevant International Standards shall be resolved by using the definition in the International Standard.

bit combination: An ordered set of bits that represents a character or is used as a part of the representation of a character.

byte: A bit string that is operated upon as a unit and the size of which is independent of redundancy or framing techniques.

character: A member of a set of elements used for the organization, control or representation of data.

code extension: The techniques for the encoding of characters that are not included in the character set of a given code.

control character: A control function the coded representation of which consists of a single bit combination.

control function: An action that affects the recording, processing, transmission or interpretation of data and that has a coded representation consisting of one or more bit combinations.

graphic character: A character, other than a control function, that has a visual representation normally handwritten, printed or displayed.

A.7. ISO 2022 Information Processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques

This International Standard was originally written to establish extension techniques for the 7-bit codes of ISO 646. It has been revised twice so that it now also provides the basic framework for an 8-bit code family which is compatible with the 7-bit codes. The four interrelated clauses cover

- the extension of the 7-bit code remaining in a 7-bit environment;
- the structure of a family of 8-bit codes;
- the extension of an 8-bit code remaining in an 8-bit environment;
- the relationship between the 7-bit code and an 8-bit code.

The middle two clauses are of special relevance to this document although portions of the others should be read and understood in order to set the context for the relevant material.

Some underlying assumptions from the standard are recorded here in order to understand the context of these agreements. Clause 2 notes that code extension techniques are designed to be used for data to be processed serially in a forward direction.

A.7.1. Structure of a Family of 8-bit codes

Clause 7 of the standard describes a family of 8-bit codes obtained from the 7-bit set. The family of 8-bit codes is obtained by the addition of one bit to each of the bit combinations of the 7-bit code producing a set of 256 8-bit combinations. The characters of the 7-bit code are assigned to the 128 bit combinations for which the eighth bit is set to ZERO. The 128 additional bit combinations for which the eighth bit is set to ONE are available for assignment. The 8-bit code table of clause 7.1 is a 16 by 16 array of columns numbered 00 to 15 and rows numbered 0 to 15. Columns 08 and 09 are provided for control characters and columns 10 to 15 for graphic characters.

The following figure shows the basic code structure for 8-bit character codes. This structure is followed by the standards described in this annex.

8-bit Code Structure

<table>
<tr><th></th><th>00</th><th>01</th><th>02</th><th>03</th><th>04</th><th>05</th><th>06</th><th>07</th><th>08</th><th>09</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>0</td><td colspan="2" rowspan="16">A set of 32 control characters</td><td>SP</td><td colspan="5"></td><td colspan="2" rowspan="16">A set of 32 control characters</td><td>10/0</td><td colspan="5"></td></tr>
<tr><td>1</td><td colspan="6" rowspan="14">A set of 94 or 96 graphic characters</td><td colspan="6" rowspan="14">A set of 94 or 96 graphic characters</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td colspan="5"></td><td>DEL</td><td colspan="5"></td><td>15/15</td></tr>
</table>

The family concept is described in clause 7.2 as

a) a set of 32 additional control characters can be selected for columns 08 and 09;

b) a set of 94 or 96 additional graphic characters can be selected for columns 10 to 15. If a set of 94 graphic characters is invoked in columns 10 to 15, positions 10/0 and 15/15 shall not be used.

Three control functions were provided by ISO 646 for purposes of code extension. ISO 2022 uses these three and adds 7 more for use in the 8-bit environment. For reference purposes the corresponding characters from the 7-bit environment are shown also. The following table shows these control functions.

7-bit Name	Abbreviation	8-bit Name	Abbreviation
ESCAPE	ESC	ESCAPE	ESC
SHIFT-OUT	SO	LOCKING-SHIFT ZERO	LS0
SHIFT-IN	SI	LOCKING-SHIFT ONE	LS1
LOCKING-SHIFT TWO	LS2	LOCKING-SHIFT TWO	LS2
LOCKING-SHIFT THREE	LS3	LOCKING-SHIFT THREE	LS3
SINGLE-SHIFT TWO	SS2	SINGLE-SHIFT TWO	SS2
SINGLE-SHIFT THREE	SS3	SINGLE-SHIFT THREE	SS3
		LOCKING-SHIFT ONE RIGHT	LS1R
		LOCKING-SHIFT TWO RIGHT	LS2R
		LOCKING-SHIFT THREE RIGHT	LS3R

A.7.2. Elements of Code Extension in an 8-bit Environment

The elements of code extension in an 8-bit environment are shown in the following table taken from Clause 8.1 of the standard:

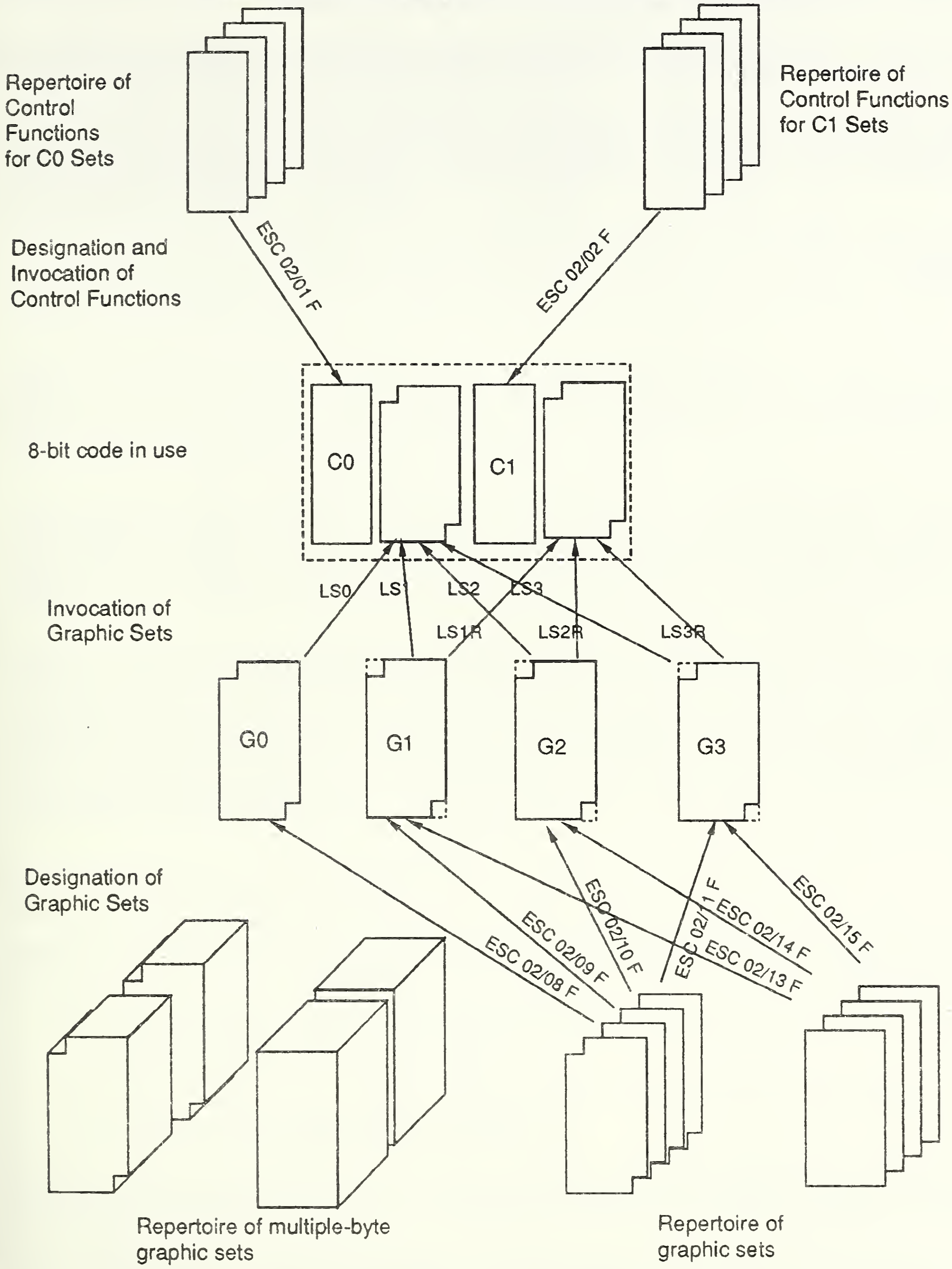
Set	Description	Columns occupied
C0	32 control characters	00 to 01
C1	32 control characters	08 to 09
G0	94 graphic characters	02 to 07
G1	94 or 96 graphic characters	02 to 07 or 10 to 15
G2	94 or 96 graphic characters	02 to 07 or 10 to 15
G3	94 or 96 graphic characters	02 to 07 or 10 to 15

A.7.3. Multiple Character Sets

<<Describe multi-level designation and invocation here.>>

The standard defines a graphic character set extension strategy in which a designating escape sequence is used to select up to four graphic character sets from the International Character Set Register. An invocation sequence is then used to select up to two graphic sets from the designated sets for concise access to the characters. The following figure shows the technique for the 8-bit environment.

Code Extension in an 8-bit Environment


Repertoire of Control Functions for C0 Sets
Repertoire of Control Functions for C1 Sets
Designation and Invocation of Control Functions
ESC 02/01 F
ESC 02/02 F
8-bit code in use
C0
C1
Invocation of Graphic Sets
LS0
LS1
LS2
LS3
LS1R
LS2R
LS3R
G0
G1
G2
G3
Designation of Graphic Sets
ESC 02/08 F
ESC 02/09 F
ESC 02/10 F
ESC 02/11 F
ESC 02/13 F
ESC 02/14 F
ESC 02/15 F
Repertoire of multiple-byte graphic sets
Repertoire of graphic sets

The standard defines two terms for use in describing code extension practices: to designate and to invoke. They are defined as follows:

to designate: To identify a set of characters that are to be represented, in some cases immediately and in others on the occurrence of a further control function, in a prescribed manner.

to invoke: To cause a designated set of characters to be represented by the prescribed bit combinations whenever those bit combinations occur, until an appropriate code extension function occurs.

Designation of a character set is usually achieved by employing an escape sequence defined by the standard along with values assigned by a registration authority. In many cases, designation of a character set also implies invocation. In other cases a character set must be explicitly invoked usually by using a shift function.

The following table defines the use of the locking shift functions in an 8-bit environment for extension of the graphic set.

Function	Abbreviation	Set Invoked	Columns affected
LOCKING-SHIFT ZERO	LS0	G0	02 to 07
LOCKING-SHIFT ONE	LS1	G1	02 to 07
LOCKING-SHIFT ONE RIGHT	LS1R	G1	10 to 15
LOCKING-SHIFT TWO	LS2	G2	02 to 07
LOCKING-SHIFT TWO RIGHT	LS2R	G2	10 to 15
LOCKING-SHIFT THREE	LS3	G3	02 to 07
LOCKING-SHIFT THREE RIGHT	LS3R	G3	10 to 15

The meanings of control characters in columns 00, 01, 08 and 09 shall not be affected by the occurrence of these locking shift functions.

Clause 6.4 states that at the beginning of any information interchange, except where interchanging parties have agreed otherwise, all designations shall be defined by the use of appropriate escape sequences, and the shift status shall be defined by the use of the appropriate locking shift functions.

A.7.4. Announcement of Extension Facilities

A code extension facility consists of the elements of code extension employed as well as the means by which these elements are designated and invoked. Thus the control function sets, the graphic character sets, and the character shifting codes must be specified. Specification of control function sets and graphic character sets also specifies the designation and invocation sequences required to use their codes.

Clause 9 of ISO 2022 describes how the various extension facilities are to be made known. If an announcement is to be embedded in the interchanged information, the form is described. The announcement may be omitted by agreement between the interchanging parties. Some restrictions are imposed on the defined announcer sequences. For example the sequence ESC 02/00 04/03 specifies that 1) the G0 and G1 sets shall be used in an 8-bit environment only, 2) the designating escape sequences also invoke the G0 and G1 sets into columns 02 to 07 and 10 to 15, respectively, and 3) no locking shift functions shall be used.

A.7.5. Composite Graphic Characters

Clause 6.1.8 of the standard addresses methods for the representation of additional graphic characters by the combination of two or more graphic characters in the same position. Two methods are provided for:

a) graphic characters having implicit forward motion (spacing characters) used in conjunction with BACKSPACE or CARRIAGE RETURN;

b) graphic characters having no implicit forward motion (non-spacing characters) used in combination with spacing graphic characters.

Method b allows for the specification of characters with diacritical marks. The technique is known colloquially as the "dead key" approach. A non-spacing accent grave character is immediately followed by the character it modifies.

A.7.6. International Register of Coded Character Sets to be used with Escape Sequences

ISO 2375 specifies procedures to be used to assign meanings to the final bit combinations of escape sequences defined in ISO 2022. The International Register of Coded Character Sets to be used with escape sequences is the document which records these assignments. The current International Registration Authority for ISO 2375 is the European Computer Manufacturers Association (ECMA).

A.8. Character Sets

Several character set standards are described here. The standards chosen for description are each used by one or more known OSI applications. The usage of these standards is summarized in tabular form.

A.8.1. ISO 646 *7-bit coded character set for information processing interchange* and CCITT Recommendation T.50 *International Alphabet No. 5*

This International Standard specifies a set of 128 characters with their coded representation. The 128 bit combinations of the 7-bit code represent control characters and graphic characters. The allocation of characters to bit combinations is based on the following principles:

- the bit combinations 0/0 to 1/15 represent 32 control characters;
- the bit combination 2/0 represents the character SPACE, which is interpreted as both a control character and a graphic character;
- the bit combinations 2/1 to 7/14 represent up to 94 graphic characters;
- the bit combination 7/15 represents the control character DELETE.

The 7-bit code table consists of 128 positions arranged in 8 columns and 16 rows. The columns are numbered from 0 to 7, and the rows are numbered 0 to 15.

Most of these characters are mandatory and unchangeable, but provision is made for some flexibility to accommodate national and other requirements. The standard provides guidance on how to exercise the options offered in order to define specific national versions and application-oriented versions. It further specifies an International Reference Version in which all options have been exercised.

<<Editor's Note: A revision of ISO 646 which has achieved DP status revises this table.>>

X3.4-1977 ASCII

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SUB	*	:	J	Z	j	z
11	VT	ESC	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	|
13	CR	GS	-	=	M	]	m	}
14	SO	RS	.	>	N	^	n	~
15	SI	US	/	?	O	_	o	DEL

ISO 646-1983 IRV

	0	1	2	3	4	5	6	7
0	NUL	TC7	SP	0	@	P	`	p
1	TC1	DC1	!	1	A	Q	a	q
2	TC2	DC2	"	2	B	R	b	r
3	TC3	DC3	#	3	C	S	c	s
4	TC4	DC4	¤	4	D	T	d	t
5	TC5	TC8	%	5	E	U	e	u
6	TC6	TC9	&	6	F	V	f	v
7	BEL	TC10	'	7	G	W	g	w
8	FE0	CAN	(	8	H	X	h	x
9	FE1	EM	)	9	I	Y	i	y
10	FE2	SUB	*	:	J	Z	j	z
11	FE3	ESC	+	;	K	[	k	{
12	FE4	IS4	,	<	L	\	l	|
13	FE5	IS3	-	=	M	]	m	}
14	SO	IS2	.	>	N	^	n	‾
15	SI	IS1	/	?	O	_	o	DEL

ISO 646 International Reference Version

A.8.2. ISO 8859 *Information Processing — 8-bit single-byte coded character sets*

This International Standard is a multiple part standard. Each part specifies a set of up to 191 graphic characters and the coded representation of each of these characters by means of a single 8-bit byte. The use of control functions for the coded representation of composite characters is prohibited. Each set is intended for a group of languages. Part 1 of ISO 8859 specifies a set of 191 graphic characters identified as Latin alphabet No. 1. This set of graphic characters is suitable for use in a version of an 8-bit code according to ISO 2022.

The standard specifically notes that it is not intended for use with CCITT defined Telematic services. If information coded according to ISO 8859 is to be transferred to such services, it will have to conform at the coding interface to their requirements.

ISO 8859/1-1987 Latin Alphabet No. 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð
1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
8			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
11			+	;	K	[	k	{			«	»	Ë	Û	ë	û
12			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
13			–	=	M	]	m	}			SHY	½	Í	Ý	í	ý
14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
15			/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

ISO 8859/1 - 1987 Latin Alphabet No. 1

A.8.3. ISO 6937 *Information Processing — Coded Character Sets for Text Communication*

This International Standard specifies repertoires of graphic characters and control functions, and their coded representation for use in text communication. This International Standard consists, at present, of two parts, as follows:

- ISO 6937/1, General Introduction.
- ISO 6937/2, Latin Alphabetic and non-alphabetic graphic characters.

The specifications are based on the 7-bit coded character set specified in ISO 646, the 7-bit and 8-bit code extension techniques of ISO 2022, and the definitions of additional control functions given in ISO 6429.

ISO 6937 was developed in parallel with CCITT Recommendations which in the standard are referred to as S.61 and S.100. These CCITT Recommendations were moved to a new section in 1984 and were renumbered T.61 and T.100. This 1984 designation is being carried forward in the 1988 CCITT Recommendations.

A.8.3.1. ISO 6937/1 *Information Processing — Coded Character Sets for Text Communication — Part 1: General Introduction*

Annex A of this International Standard describes a method of identification of graphic characters and control functions which is used in other parts of the standard to define the characters of the standard.

A.8.3.2. ISO 6937/2 *Information Processing — Coded Character Sets for Text Communication — Part 2: Latin Alphabetic and Non-alphabetic Graphic Characters*

This part of the standard

a) defines a repertoire of Latin alphabetic and non-alphabetic characters for the communication of text in European languages;

b) specifies coded representations for the graphic characters;

c) specifies rules for the definition and use of graphic character subrepertoires.

A graphic subrepertoire is a subset of the defined character repertoire. Because the number of characters defined by this standard is so large, this subsetting facility allows for the use of well defined subsets of the characters available. Rules for the definition of subrepertoires are defined in clause 5. The procedure for registration of subrepertoires is given in ISO 7350. Three standard subrepertoires are defined in Annex A of the standard.

Graphic characters which represent accented letters and umlauts are specified using a two byte sequence composed of the diacritical character immediately followed by the character modified. The allowable combinations are carefully defined in the standard and only these combinations are permitted.

ISO 6937/2-1983 Addendum 1
Full Repertoire

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			SP	0	@	P	`	p			NBSP	°		―	Ω	ĸ
1			!	1	A	Q	a	q			¡	±	`	¹	Æ	æ
2			"	2	B	R	b	r			¢	²	´	®	Đ	đ
3			#	3	C	S	c	s			£	³	^	©	ª	ð
4			¤	4	D	T	d	t			$	×	~	™	Ħ	ħ
5			%	5	E	U	e	u			¥	µ	¯	♪		ı
6			&	6	F	V	f	v				¶	˘	¬	Ĳ	ĳ
7			'	7	G	W	g	w			§	·	˙	¦	Ŀ	ŀ
8			(	8	H	X	h	x				÷	¨		Ł	ł
9			)	9	I	Y	i	y			‘	’			Ø	ø
10			*	:	J	Z	j	z			“	”	˚		Œ	œ
11			+	;	K	[	k	{			«	»	¸		º	ß
12			,	<	L	\	l	|			←	¼	_	⅛	Þ	þ
13			-	=	M	]	m	}			↑	½	˝	⅜	Ŧ	ŧ
14			.	>	N	^	n	‾			→	¾	˛	⅝	Ŋ	ŋ
15			/	?	O	_	o	DEL			↓	¿	ˇ	⅞	ŉ	SHY

ISO 6937-2 Latin Alphabetic and non-Alphabetic Characters

A.8.4. CCITT Recommendation T.51 *Coded Character Sets for Telematic Services*

This Recommendation specifies a primary set and a supplementary set of graphic characters which are to be the respective supersets of various primary and supplementary character sets to be used in various telematic services. The Recommendation also describes those code extension mechanisms which are relevant to existing telematic services.

A.8.5. CCITT Recommendation T.61 *Character Repertoire and Coded Character Sets for the International Teletex Service*

This Recommendation contains detailed definitions of the repertoires of graphic characters and control functions to be used in the basic International Teletex service, and their coded representations for communication.

A.9. ASN.1 Character String Types

Character String Types are sequences of zero, one or more characters from some specified character set. ISO 8824 defines 8 such types: NumericString, PrintableString, TeletexString (T61String), VideotexString, VisibleString (ISO646String), IA5String, GraphicString, GeneralString.

A.9.1. Universal Class Numbers and Registration Numbers

The type of each character string is identified by a Universal Class number. Universal Class numbers are assigned by ISO 8824. No other standard or private user may define these numbers. The character sets associated with each type are identified by the ISO Character Set Registration Numbers as shown in the following table:

Name of Character String Type	Universal Class Number	ISO Character Set Registration Numbers
NumericString	18	Not Registered
PrintableString	19	Not Registered
TeletexString (T61String)	20	87, 102, 103, 106, 107 + SPACE + DELETE
VideotexString	21	1, 72, 73, 102, 108, 128, 129 + SPACE + DELETE
VisibleString (ISO646String)	26	2 + SPACE
IA5String	22	1, 2 + SPACE + DELETE
GraphicString	25	All G sets + SPACE
GeneralString	27	All G sets and all C sets + SPACE + DELETE

NumericString and PrintableString do not have Registration Numbers assigned to them since their character sets are defined in table 4 and 5 respectively of ISO 8824.

A.9.2. Initial States

Some character string types allow multiple character sets through code extension techniques. For these types, at the beginning of each string there are initial default character sets to be designated in G0 and/or C0 and/or C1 and for each character set there is an assumed escape sequence. The following table drawn from ISO 8825 describes these initial states.

Name of Character String Type	Initial G0 (Reg. No.)	Initial C0 (Reg. No.)	Initial C1 (Reg. No.)	Initial ESC Seq and Lock Shift Function	Code Extension
NumericString	2	None	None	ESC 2/8 4/0 LS0	No
PrintableString	2	None	None	ESC 2/8 4/0 LS0	No
TeletexString (T61String)	102	106	107	ESC 2/8 4/0 LS0 ESC 2/1 4/5 ESC 2/2 4/8	Yes
VideotexString	102	1	73	ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1	Yes
VisibleString (ISO646String)	2	None	None	ESC 2/8 4/0 LS0	No
IA5String	2	1	None	ESC 2/8 4/0 LS0 ESC 2/1 4/0	No
GraphicString	2	None	None	ESC 2/8 4/0 LS0	Yes
GeneralString	2	1	None	ESC 2/1 4/0 LS0 ESC 2/1 4/0	Yes

For example, VideotexString initial G0 set is Primary Teletex Graphic Set (ISO Registration Number 102), initial C0 set is ISO 646 C0 set (ISO Registration Number 1), initial C1 set is Attribute Control Set for Videotex (ISO Registration Number 73), initial escape sequence and locking shift function is ESC 2/8 7/5 LS0, and ESC 2/2 4/1, and code extensions are permitted.

A.10. Use of ASN.1 OctetString as a Character String

<<Editor's Note: Add a description of ODA treatment of character sets.>>

A.11. Escape Sequences for Character Set Designation

This information is extracted from the ISO Register. In some cases, the defaults supplied by ASN.1 make the use of these escape sequences unnecessary. In some cases, this information is carried by application protocol elements.

Graphic Set Designation

Set No.	G0	G1	G2	Name
2	ESC 2/8 4/0			ISO 646 IRV
6	ESC 2/8 4/2			ISO 646 USA
87	ESC 2/4 2/8 4/2	ESC 2/4 2/9 4/2		JIS X0208
100		ESC 2/13 4/1	ESC 2/14 4/1	ISO 8859/1 Right Hand Part
102	ESC 2/8 7/5			CCITT T.61 Primary
103			ESC 2/10 7/6	CCITT T.61 Supp
126		ESC 2/13 4/6		ISO 8859/7 Greek
142			ESC 2/14 4/10	ISO 6937/2 Ad1 Supp

Control Set Designation

Set No.	C0	C1	Name
1	ESC 2/1 4/0		ISO 646 C0
106	ESC 2/1 4/5		CCITT T.61 Primary
107		ESC 2/2 4/8	CCITT T.61 Suppl.

<<Editor's Note: Add 6429 designation.>>

<<Editor's Note: Add DIS 10538 amd DIS 10367?>>

Table of Contents

List of Tables

0 INTRODUCTION

This is the definition of a single specification for two Open Document Architecture (ODA) Document Application Profiles (DAPs) named National Institute of Standards and Technology (NIST) ODA Raster DAP. The two DAPs differ only in the encoding of the data stream. One uses the ASN.1 based ODIF encoding. The other uses the SGML/SDIF based ODIF endcoding. When this document refers to *this profile*, it is referring to either of the DAPs defined by this specification.

This DAP is suitable for interchanging documents in formatted form. The documents contain only raster images. This DAP has been prepared by the ODA Special Interest Group of the NIST Open Systems Interconnection (OSI) Implementors Workshop. The DAP is defined in accordance with ISO 8613-1 and CCITT T.411 and follows the standardized proforma and notation defined in the proposed Draft Addendum to ISO 8613-1 Annex F (to be published). The DAP is based on ODA as defined in ISO 8613 and the Draft Addendum to ISO 8613, Part 7.

1 SCOPE AND FIELD OF APPLICATION

This DAP specifies an interchange format suitable for transfer of structured documents between equipment designed for raster processing. Such documents contain only bi-tonal raster graphics content, such as engineering drawings and illustrations, although there is no restriction on the minimum size of the image.

This document defines a DAP that allows large format raster documents to be interchanged in a formatted form in accordance with ISO 8613.

It is assumed that, when negotiation is performed by the service using this DAP, all non-basic features are subject to negotiation.

This DAP is independent of the processes carried out in an end system to create, edit, or reproduce raster documents. It is also independent of the means to transfer the document which, for example, may be by means of communication links or exchanged storage media.

The features of a document that can be interchanged using this DAP fall into the following categories:

- o Page format features - these concern how the layout of each page of a document will appear when reproduced;

- o Raster graphics layout and imaging features - these concern how the document content will appear within pages of the reproduced document; and

- o Raster graphics coding - these concern the raster graphics representations and control functions that make up the document raster graphics content.

2 REFERENCES

The following references are required in order to implement this DAP:

ISO 8613-1 - Information processing - Text and Office Systems; Open Document Architecture (ODA) and Interchange Format Part 1: Introduction and General Principles (1989)

ISO 8613-2 - Information processing - Text and Office Systems; Open Document Architecture (ODA) and Interchange Format Part 2: Document Structures (1989)

ISO 8613-4 - Information processing - Text and Office Systems; Open Document Architecture (ODA) and Interchange Format Part 4: Document Profile (1989)

ISO 8613-5 - Information processing - Text and Office Systems; Open Document Architecture (ODA) and Interchange Format Part 5: Open Document Interchange Format (1989)

ISO 8613-7 - Information processing - Text and Office Systems; Open Document Architecture (ODA) and Interchange Format Part 7: Raster Graphics Content Architectures (1989)

ISO 8613-7 - Draft Addendum: Tiled Raster Graphics Addendum to ISO 8613, Part 7 (January 1990)

ISO 8613-1 - Draft Addendum: Document Application Profile Proforma and Notation (to be published)

ISO 8824 - Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)

ISO 8825 - Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

ISO 8879 - Information processing - Text and office systems - Standard Generalized Markup Language (SGML).

ISO 9069 - Information processing - SGML support facilities - SGML Document Interchange Format (SDIF).

ISO 9070 - Information processing - SGML support facilities - Registration procedures for public owner identifiers.

CCITT T.6 - Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, (1988)

CCITT T.411 Open Document Architecture (ODA) and Interchange Format - Introduction and general principles (1989)

CCITT T.412 Open Document Architecture (ODA) and Interchange Format - Document structures (1989)

CCITT T.414 Open Document Architecture (ODA) and Interchange Format - Document profile (1989)

CCITT T.415 Open Document Architecture (ODA) and Interchange Format - Open document interchange format (1989)

CCITT T.417 Open Document Architecture (ODA) and Interchange Format - Raster graphics content architecture (1989)

3 DEFINITIONS AND ABBREVIATIONS

The definitions given in ISO 8613-1 are applicable to this document.

4 RELATIONSHIP TO OTHER DAPS

Functionally, this DAP is similar to the CCITT Recommendation T.503, A Document Application Profile for the Interchange of Group 4 Facsimile Documents.

5 CONFORMANCE

In order to conform to this DAP, a data stream representing a document must meet the requirements specified in subclause 5.1.

Subclause 5.2 specifies the requirements for implementations that originate and/or receive data streams conforming to this DAP.

5.1 Data stream conformance

The following requirements apply to the encoding of data streams that conform to these agreements.

- o The data stream shall be encoded in accordance with the ASN.1 encoding rules defined in ISO 8825 or the SGML encoding rules defined in ISO 8879;

- o The data stream shall be structured in accordance with the interchange format defined in clause 8 of this DAP;

- o The document shall be structured in accordance with only the formatted document architecture class specified in clause 7 of this DAP. In addition, the document shall contain all mandatory constituents specified for that class and may optionally contain constituents permitted for that class as specified in clause 7;

- o Each constituent shall contain all those attributes specified as required for that constituent in this profile. Other attributes may be specified provided they are permitted for that constituent;

- o The attributes shall have values within the range of permissible values specified in this profile;

- o The encoded document shall be structured in accordance with the abstract document architecture defined in ISO 8613-2;

- o The encoded document shall be structured in accordance with the characteristics defined in clause 6 of this DAP and shall contain only those features defined in clause 6.

5.2 Implementation conformance

This subclause states the requirements for implementations claiming conformance to this DAP.

An implementation claiming to originate and/or receive data streams conforming to this DAP must complete a conformance test to be defined by a test plan to be developed by DoD and NIST.

A conforming receiving implementation must be capable of receiving any data stream conforming to this DAP. "Receiving" means not rejecting a data stream conforming to this DAP and usually, but not always, involves recognizing and further processing the data stream elements. The explicit meaning of "receiving" is determined by a conformance test plan to be developed by DoD and NIST.

6 CHARACTERISTICS SUPPORTED BY THIS DAP

This clause describes the characteristics of documents that can be represented by data steams conforming to this profile. This clause also describes how these characteristics are represented in terms of divisional components of the data streams.

6.1 Overview

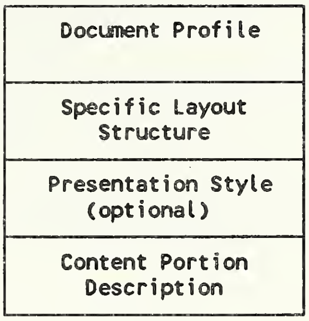
This DAP describes the features of ISO 8613 that are needed to support the interchange of documents containing only raster graphics content. It specifies interchange formats for the transfer of structured documents with simple layout structures.

This DAP describes documents that can be interchanged in the formatted form, which facilitates the reproduction of a document as intended by the originator.

Only one category of content is allowed within the document, namely, a raster graphics content in the formatted processable form, which facilitates the reproduction of the document content as intended by the originator or facilitates the revision of the document content..

This subclause describes the layout features that can be represented in documents conforming to this DAP. The features are described in terms that are typical of the user-perceived capabilities and semantics found in a raster document interchange environment.

For the purpose of interchange, a document is represented as a collection of **constituents**, each of which is a set of attributes. The constituents that make up a formatted document are defined below in this subclause and are illustrated in the following figure.



Document Profile

Specific Layout Structure

Presentation Style (optional)

Content Portion Description

Constituents defined as **required** must occur in any document that conforms to this profile. Constituents listed as **optional** may or may not be present in the document, depending on the requirements of the particular document.

The required constituents include:

- a document profile,
- layout object descriptions representing a specific layout structure, and
- content portion description.

The only optional constituent is the presentation style.

6.2 Logical Constituents

Not applicable.

6.3 Layout Constituents

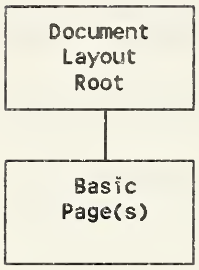
This subclause describes the features of the layout objects that can be represented in documents conforming to this DAP.

6.3.1 Overview of the Layout Characteristics

The document structure allows the document content to be laid out and presented in one or more pages. Each page consists of only a single raster graphics content representing an engineering drawing, illustration, or other raster scanned image.

A specific layout structure of the document conforming to this application profile consists of a two-level hierarchy of a document layout root and a set of basic pages. The basic page contains the content information.

The following is a document layout structure derived from this DAP:



6.3.2 DocumentLayoutRoot

A DocumentLayoutRoot is the top level in a document layout structure. A DocumentLayoutRoot may consist of a sequence of one or more BasicPage constituent constraints.

6.3.3 BasicPage

A BasicPage is a basic layout object that corresponds to the area used for presenting the raster image content of the document.

6.3.3.1 Page Dimensions

A wide variety of page dimensions are supported including large format raster documents. The dimensions of the pages may be specified as any value, in BMU measurement units, including the larger sizes produced from foldout-size images and roll paper. These sizes apply to both portrait and landscape orientations.

Dimensions equivalent to or less than the actual (nominal) page sizes of ANSI E in both portrait and landscape orientations are basic values. Larger dimensions (F-K) including those produced from roll paper are non-basic and their use must be indicated in the document profile. Although ISO A0-A4 sizes are not generally used, the A1-A4 sizes do fall within the range of the ANSI E sizes and therefore could be considered basic values (See table 2). A0 size is a non-basic value.

The default dimensions are the Common Assured Reproduction Area (CARA) of North American Letter (A). Any default page dimensions may be specified in the document profile subject to the maximum dimensions defined above by using the Page-dimensions attribute. The Page-position attribute may be used to specify the position of the pel array image on the page. Although actual page dimensions may be used allowing for the raster content to completely fill a page leaving no borders, it is advised that the assured reproduction area (ARA) listed in table 1 be used wherever feasible. See ISO 8613-2, subclause 7.3, General rules for positioning pages on presentation surfaces.

6.3.3.2 Nominal Page Sizes

The nominal page sizes that may be specified are listed in Table 1. These may be specified in portrait or landscape orientations. All values of nominal page size up to ANSI E size are basic. All sizes larger than

ANSI E size and roll paper are non-basic and their use in a document must be indicated in the document profile using the Medium-type attribute (See table 2).

Any of the nominal page sizes defined in Table 1, subject to the restriction specified above, may be specified as the default value in the document profile.

Table 1 also includes the recommended assured reproduction area (ARA). Information loss may occur when a document is reproduced if the dimensions of the BasicPage exceed the ARA for the specified nominal page size.

Table 1 Dimensions for Various Page Sizes

Page Type	Size	Size (BMU)	ARA (BMU)
- Metric	(mm)		
ISO-A4	210X297	9920 x 14030	9240 x 13200
ISO-A3	297X420	14030 x 19840	13200 x 18480
ISO-A2	420X594	19840 x 28060	18898 x 27118
ISO-A1	594X840	28060 x 39680	26173 x 37843
ISO-A0	840X1188	39680 x 56120	37843 x 54283
- ANSI, North American(NA)	(inches)		
NA-A	8.5X11	10200 x 13200	9240 x 12400
NA-B	11X17	13200 x 20400	12744 x 19656
NA-C	17X22	20400 x 26400	19500 x 25800
NA-D	22X34	26400 x 40800	25800 x 39600
NA-E	34X44	40800 x 52800	39600 x 52200
NA-F	28X40	33600 x 48000	31400 x 47400
NA-G	11X90	13200 x 108000	12400 x 106800
NA-H	28X143	33600 x 171600	31400 x 170400
NA-J	34X176	40800 x 211200	39600 x 210000
NA-K	40X143	48000 x 171600	47400 x 170400
NA-Legal	8.5X14	10200 x 16800	9240 x 15480
- Foldouts			
Small	11X14	13200 x 16800	12744 x 15480
NA-B	11X17	(same as NA-B above)	

These page sizes are for the portrait orientation.

Table 2 Layout Attributes

Attributes	Basic Values	Default Values	Non-Basic Values
Page Dimensions*	CARA NA A-F, CARA NA-Legal ISO A4-A1 Small Foldout	CARA NA-A	ARA NA G-K ISO A0 11" roll
Medium-type* (Nominal page size)	NA A-F, NA-Legal, ISO A4-A1 Small Foldout	NA-A	NA G-K ISO A0 11" roll

* see Table 1

6.4 Document Layout Control

A document layout structure contains only a basic page with content information.

Each raster graphics content must be allocated to a single basic page.

A page containing tiled raster graphics content may consist of as many tiles as is necessary to represent the image in digital form.

6.5 Content Layout and Imaging Control

A document may contain only raster graphics content portions as specified in ISO 8613-7.

6.5.1 Raster Graphics Content Architecture

Only the formatted processable raster graphics content architecture class is supported by this profile. The content architecture class associated with a basic page is specified using the document architecture class attribute Content-architecture-class. The default value that must be specified in the document profile is formatted processable raster content architectures.

When using raster graphics content, only one content portion may be associated with a basic page.

6.5.2 Raster Graphics Encoding Methods

Three encoding methods, CCITT T.6 (untiled), Tiled, and Bitmap are supported by this profile as basic values. Neither the CCITT T.4 one dimensional method nor the CCITT T.4 two dimensional method is supported.

The CCITT Recommendation T.6 Group 4 compression algorithm shall be used in all cases, tiled and untiled, except where it is more efficient to retain an image or tile image in bitmap format or to specify a tile as being either all background or all foreground.

When the coding type is specified as CCITT T.6, the encoding must be compressed and the "code extension" technique in T.6 encoding is not used. That is to say that uncompressed data cannot occur within a T.6 encoded data stream. Thus, there is no need for a default value of the Compression attribute and this attribute will not appear in the description of the raster content.

In a content portion, it is required that the Number-of-pels-per-line parameter of the Coding-attributes attribute be specified. The use of the Number-of-lines parameter is optional. The value of these parameters shall be a positive number. Otherwise, no constraints are placed on these parameters by this profile. This profile places no constraints on the size of the pel arrays that may be used as long as the size does not exceed the page dimension size.

The type of coding method used is specified by the attribute Type-of-coding. The use of this attribute is mandatory in the Document-architecture-defaults of the document profile to define the default value of either T.6 encoding (untiled) or Tiled encoding. The use of this attribute in the description of the content portions is non-mandatory. If this attribute is not specified for a particular content portion, then the default value specified in the Document-architecture-defaults of the document profile is used.

If the Tiled encoding method is used, the default value of 512 for the Number-of-pels-per-tile-line and Number-of-lines-per-tile must be used. No other values are supported, therefore these two attributes do not need to be specified. If the Tile-types attribute is not present, then all tiles will be T.6 encoded. If it is present, then there must be a value specified for each tile in which case only null background, null foreground, T.6 encoded, or bitmap encoded values are supported. T.4 one dimensional and T.4 two dimensional encodings are not supported. There are no restrictions on the use of the Tiling-offset other than that specified in ISO 8613-7 Addendum.

See table 3 for a tabulated list of the attributes and their basic, default, and non-basic values.

6.5.3 Raster Presentation

Raster presentation is controlled by the presentation attributes specified in ISO 8613-7. This DAP provides for additional constraints on these presentation attributes as specified below.

The basic Pel-path values supported by this profile are 0 and 90 degrees. The Pel-path values of 180 and 270 degrees are non-basic.

The basic Line-progression value supported by this profile is 270 degrees. The Line-progression value of 90 degrees is non-basic.

The basic Pel-spacing values supported by this profile are the ratios equal to 6 and 4 BMU between adjacent pels. This corresponds to equivalent resolutions of 200 and 300 pels per 25.4mm (1 in.), respectively when the BMU is interpreted as 1/1200 inch. Values for Pel-spacing other than these ratios are non-basic, i.e., 5, 3, 2, and 1 BMU. These correspond to equivalent resolutions of 240, 400, 600, and 1200 pels per 25.4mm (1 in.).

There are no restrictions on the use of the Clipping attribute. The Spacing-ratio and Image-dimensions attributes are not supported.

See table 4 for a tabulated list of the attributes and their basic, default, and non-basic values.

Table 3 Content Coding Attributes

Attributes	Basic Values	Default Values	Non-Basic Values
Number-of-pels-per-line	any positive integer	None	None
Number-of-lines	any positive integer	None	None
Tiling-offset*	(any non-neg integer < 512, any non-neg integer < 512)	(0,0)	None
Tile-types*	T.6 encoded bitmap encoded null background null foreground	T.6 encoded	None
Type-of-coding	T.6 encoding (untiled) bitmap (untiled) tiled	T.6 encoding	None

* Only used if Type-of-coding is "tiled"

Table 4 Presentation Attributes

Attributes	Basic Values	Default Values	Non-Basic Values
Pel-path	0, 90 deg	0 deg	180, 270 deg
Line-progression	270 deg	270 deg	90 deg
Pel-spacing	6, 4 BMU (200, 300)	4 BMU (300)	5,3,2,1 BMU
Clipping	Two Coord. Pairs (any non-negative integer, any non-negative integer)	(0,0), (N-1, L-1)	None

6.6 Miscellaneous Features

Specification of the attribute Application-comments is optional. When used in conjunction with the Type-of-coding of 'Tiled', it contains a sequence of positive integers, one for each tile in the content portion. The sequence of integers is a set of indices representing the octet offsets to the beginning of the respective tiles, starting from the location of the first tile. The first tile will be at offset zero (0). The integers will be sequenced in the same order as the tiles. The tiles will be sequenced primarily in the Pel-path and secondarily in the Line-progression direction as defined by the presentation attributes.

6.7 Document Management Features

Every document interchanged in accordance with this DAP must include a document profile containing information which relates to the document as a whole. The document profile used in this DAP must identify the contents as raster graphics data.

The features specified by the document profile are listed below. A definition of the information contained in these features is given in the corresponding attribute definitions in ISO 8613-4.

Presence of document constituents:

- o specific layout structure;
- o presentation styles (optional).

Document characteristics:

- o document application profile;

o document application profile defaults;

o document architecture class;

o content architecture class;

o interchange format class;

o ODA version date;

o raster graphics content defaults.

Non-basic document characteristics:

o page dimensions;

o medium type;

o raster graphics presentation features.

The attributes applicable to the document profile are defined in Table 5. The folowing notation is used in the class column of this table:

o m mandatory attribute

o nm non-mandatory attribute

o d defaultable attribute

Capital letters (M, NM, and D) are used for groups of attributes.

Table 5 Document Profile Attributes

Attribute	Class	Permissible Values
Specific-layout-structure	m	present
Presentation-styles	nm	present
Document-characteristics	M	
Document-architecture-class	m	formatted
Document-application-profile	m	{-- See clause 8 for a definition of the permitted values for this attribute. --}
Content-architecture-classes	m	{2 8 2 7 2}
Interchange-format-class	m	B
ODA-version	m	ISO 8613, 1989-07-04
Document-architecture-defaults	M	
Content-architecture-class	m	formatted processable
Type-of-coding	nm	T.6 Encoding (default) Tiled Encoding
Page-dimensions	nm	See list in table 1, (Default value is NA-A, 9240 x 13200 BMU)
Medium-types	nm	See list in table 1, (Default value is NA-A, 9240 x 13200 BMU)
Page-position	nm	any coordinate pair within page
Raster-gr-content-defaults	NM	
Pel-path	nm	0, 90, 180, 270 degrees (0 is normal default)
Line-progression	nm	90, 270 degrees

		(270 is normal default)
Clipping	nm	any coordinate pair within page
Pel-spacing	nm	6 BMU (200 pels/in.) 5 BMU (240 pels/in.) 4 BMU (300 pels/in.) 3 BMU (400 pels/in.) 2 BMU (600 pels/in.) 1 BMU (1200 pels/in.) (Normal default is 4 BMU)
Non-basic-doc-characteristics	NM	
Page-dimensions	nm	See table 1, NA-F through NA-K, roll paper
Medium-types	nm	See table 1, NA-F through NA-K, roll paper
Raster-gr-presentation-features	NM	
Pel-path	nm	180, 270 degrees
Line-progression	nm	90 degrees
Pel-spacing	nm	5 BMU (240 pels/in.) 3 BMU (400 pels/in.) 2 BMU (600 pels/in.) 1 BMU (1200 pels/in.)
Document-management-attributes	M	
Document Reference	m	Any string of characters

7 SPECIFICATION OF CONSTITUENT CONSTRAINTS

7.1 Document Profile Constraints

7.1.1 Macro Definitions

```
-- Basic page dimensions. --
DEFINE(BasicPageDimension,"
    { #horizontal       { <=40800 },      #vertical         { <=52800},
-- Any size equal to or smaller than the actual page size of ISO A1 and ANSI E portrait. --
    | #horizontal       { <=52800 },      #vertical         { <=40800 } }
-- Any size equal to or smaller than the actual page size of ISO A1 and ANSI E landscape. --
")

-- Non-basic page dimensions. --
DEFINE(NonBasicPageDimensions,"
    { #horizontal       {40801..48000},   #vertical         {52801..211200}
-- Any size larger than the range of basic values in ANSI E portrait and equal to or smaller than the full
size of ANSI K portrait. --
    | #horizontal       {52801..211200},  #vertical         {40801..48000}}
-- Any size larger than the range of basic values in ANSI E landscape and equal to or smaller than the
full size of ANSI K landscape. --
")

DEFINE(NominalPageSizes,"

-- ISO Page Sizes --

    #horizontal         {9920},           #vertical         {14030}
-- ISO A4 Portrait (210mm x 297mm) --
    | #horizontal       {14030},          #vertical         {9920}
-- ISO A4 Landscape (297mm x 210mm) --
    | #horizontal       {14030},          #vertical         {19843}
-- ISO A3 Portrait (297mm x 420mm) --
    | #horizontal       {19843},          #vertical         {14030}
-- ISO A3 Landscape (420mm x 297mm) --
    | #horizontal       {19843},          #vertical         {28063}
-- ISO A2 Portrait (420mm x 594mm) --
    | #horizontal       {28063},          #vertical         {19843}
-- ISO A2 Landscape (594mm x 420mm) --
    | #horizontal       {28063},          #vertical         {39732}
-- ISO A1 Portrait (594mm x 841mm) --
    | #horizontal       {39732},          #vertical         {28063}
-- ISO A1 Landscape (841mm x 594mm) --
    | #horizontal       {39732},          #vertical         {56173}
-- ISO A0 Portrait (841mm x 1189mm) --
    | #horizontal       {56173},          #vertical         {39732}
-- ISO A0 Landscape (1189mm x 841mm) --

-- ANSI Page Sizes --

    | #horizontal       {10200},          #vertical         {13200}
-- ANSI A Portrait (8.5in x 11in) --
```

```
     | #horizontal      {13200},          #vertical        {10200}
  -- ANSI A Landscape (11in x 8.5in)  --
     | #horizontal      {10200},          #vertical        {16800}
  -- ANSI Legal Portrait (8.5in x 14in)  --
     | #horizontal      {16800},          #vertical        {10200}
  -- ANSI Legal Landscape (14in x 8.5in)  --
     | #horizontal      {13200},          #vertical        {20400}
  -- ANSI B Portrait (11in x 17in)  --
     | #horizontal      {20400},          #vertical        {13200}
  -- ANSI B Landscape (17in x 11in)  --
     | #horizontal      {20400},          #vertical        {26400}
  -- ANSI C Portrait (17in x 22in)  --
     | #horizontal      {26400},          #vertical        {20400}
  -- ANSI C Landscape (22in x 17in)  --
     | #horizontal      {26400},          #vertical        {40800}
  -- ANSI D Portrait (22in x 34in)  --
     | #horizontal      {40800},          #vertical        {26400}
  -- ANSI D Landscape (34in x 22in)  --
     | #horizontal      {40800},          #vertical        {52800}
  -- ANSI E Portrait (34in x 44in)  --
     | #horizontal      {52800},          #vertical        {40800}
  -- ANSI E Landscape (44in x 34in)  --
     | #horizontal      {33600},          #vertical        {48000}
  -- ANSI F Portrait (28in x 40in)  --
     | #horizontal      {48000},          #vertical        {33600}
  -- ANSI F Landscape (40in x 28in)  --
     | #horizontal      {13200},          #vertical        {108000}
  -- ANSI G Portrait (11in x 90in)  --
     | #horizontal      {108000},         #vertical        {13200}
  -- ANSI G Landscape (90in x 11in)  --
     | #horizontal      {33600},          #vertical        {171600}
  -- ANSI H Portrait (28in x 143in)  --
     | #horizontal      {171600},         #vertical        {33600}
  -- ANSI H Landscape (143in x 28in)  --
     | #horizontal      {40800},          #vertical        {211200}
  -- ANSI J Portrait (34in x 176in)  --
     | #horizontal      {211200},         #vertical        {40800}
  -- ANSI J Landscape (176in x 34in)  --
     | #horizontal      {48000},          #vertical        {171600}
  -- ANSI K Portrait (40in x 143in)  --
     | #horizontal      {171600},         #vertical        {48000}
  -- ANSI K Landscape (143in x 40in)  --

  -- Foldouts --

     | #horizontal      {13200},          #vertical        {16800}
  -- Foldout Portrait (11in x 14in)  --
     | #horizontal      {16800},          #vertical        {13200}
  -- Foldout Landscape (14in x 11in)  --
```

```
    | #horizontal       {13200},            #vertical          {>= 16801}
-- Any portrait size larger than the typical foldout size (11in x 14in) including 11 inch roll paper --
    | #horizontal       {>= 16801},         #vertical          {13200}
-- Any landscape size larger than the typical foldout size (14in x 11in) including 11 inch roll paper --
")

DEFINE(FDA,"            formatted (0)")

DEFINE(DAC,"
Document-profile{#Document-characteristics
 {#Document-architecture-class}}  ")

DEFINE(FPR,"            {2 8 2 7 2}")  -- Raster formatted processable --
```

7.1.2 Constituent Constraints

7.1.2.1 DocumentProfile

```
{

-- Presence of document constituents --

$FDA:  REQ    Specific-layout-structure       {'present'},
       PERM   Presentation-styles             {'present'};

-- Document characteristics --

REQ    Document-application-profile           {-- See clause 8 for a definition of the permitted values
                                              for this attribute. --},

REQ    Doc-appl-profile-defaults              {

-- Document architecture defaults --

       REQ    #content-architecture-class     {$FPR},
       PERM   #dimensions                     {$BasicPageDimensions
                                              $NonBasicPageDimensions},
       PERM   #medium-type                    {
              REQ   #nominal-page-size        {$NominalPageSizes},
              REQ   #side-of-sheet            {ANY_VALUE} },
       PERM   #type-of-coding                 {'T6 encoding'
                                              | 'tiled encoding'},
       PERM   #page-position                  {ANY_VALUE},
       PERM   raster-gr-contents-defaults     {
              PERM   #pel-path                {ANY_VALUE},
              PERM   #line-progression        {ANY_VALUE},
               PERM  #pel-spacing           {ANY_RATIO = 6/1 4/1},
               DIS   #compression           {'uncompressed'},
               PERM  #clipping              {ANY_VALUE},

REQ    Document-architecture-class          {$FDA},
REQ    Content-architecture-classes         {$FPR},
REQ    Interchange-format-class             {-- See clause 8 for a definition of the permitted values
                                            for this attribute. --},
REQ    ODA-version
       {#standard-or-recommendation  {<character-string-constraint>
       ::= "ISO 8613"},
       #publication-date        {<character-string-constraint>
       ::= "1989-07-04"} },

-- Non-basic document characteristics --

PERM   #Page-dimensions                     {$NonBasicPageDimensions},
PERM   #Medium-types                        {
       REQ     #nominal-page-size           {$NominalPageSizes},
       REQ     #side-of-sheet               {ANY_VALUE},
PERM   #Ra-gr-presentation-features         {
       PERM    #pel-path                    {'180-degrees'
                                            '270-degrees'},
       PERM    #line-progression            {'90-degrees'},
       PERM    #pel-spacing                 {ANY_RATIO <> 6/1 4/1},
       DIS     #compression                 {'uncompressed'},


-- Document management attributes --

REQ  Document-reference                     {ANY_VALUE}};
```

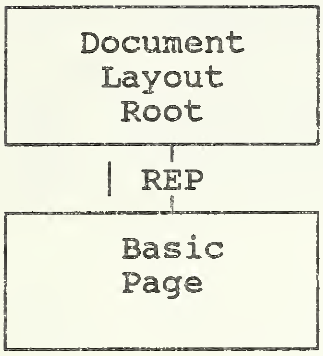
7.2 Logical Constituent Constraints

No logical constituents applicable in this subclause.

7.3 Layout Constituent Constraints

7.3.1 Diagrams of Relationships of Layout Constituents

The notation used for the structure diagrams is that specified in Annex A of ISO 8613-2.



7.3.2 Macro Definitions

None Applicable.

7.3.3 Factor Constraints

```
FACTOR:       ANY-LAYOUT           {

SPECIFIC:
PERM  Object-type                         {VIRTUAL},
PERM  Object-identifier                          {ANY_VALUE},
PERM Subordinates                         {VIRTUAL},
PERM User-visible-name                    {ANY_VALUE},
PERM User-readable-comment                {ANY_VALUE},
}
```

```
FACTOR:       ANY-PAGE     :ANY-LAYOUT {

SPECIFIC:
PERM  Object-type                         {'BASIC-PAGE'},
PERM  Dimensions                          {$BasicPageDimensions |
                                          $NonBasicPageDimensions},
PERM Page-position                        {ANY_VALUE},
}
```

7.3.4 Constituent Constraints

7.3.4.1 LayoutDocumentRoot

```
LayoutDocumentRoot          : ANY-LAYOUT          {

SPECIFIC:
```

```
REQ    Object-type                           {'DOCUMENT_LAYOUT_ROOT'},
REQ    Subordinates                          {SUB_ID_OF(BasicPage)+},
}
```

7.3.4.2 BasicPage

```
BasicPage                    : ANY-PAGE      {

SPECIFIC:
REQ    Object-type                           {'BASIC_PAGE'},
PERM   Medium-type                           {#nominal-page-size
                                             {NON_BASIC}, #side-of-sheet
                                             {ANY_VALUE}};
PERM   Application-comments                  {SEQ_INTEGERS},
                                  -- See subclause 8.2 --
PERM   Content-portions                      {ANY_VALUE},
PERM   Dimensions                            {#horizontal{
                                             #fixed{ANY_VALUE}},
                                             #vertical{#fixed{ANY_VALUE}}
                                             },
PERM   Position                              {#fixed{ANY_VALUE}},
PERM   Presentation-style                    {STYLE_ID_OF(PStyle3},
PERM   Presentation-attributes               {
       PERM  #raster-attributes              {
             PERM  Pel-path                  {ANY_VALUE},
             PERM  Line-progression          {ANY_VALUE},
             PERM  Pel-spacing               {ANY_VALUE},
             PERM  Clipping                  {ANY_VALUE} } }; }
```

7.4 Layout Style Constraints

No layout style constraints applicable in this subclause.

7.5 Presentation Style Constraints

7.5.1 Macro Definitions

```
DEFINE(R-Pres-Attr,"
PERM   Pel-path                              {ANY_VALUE},
PERM   Line-progression                      {ANY_VALUE},
PERM   Pel-spacing                           {ANY_VALUE},
PERM   Clipping                              {ANY_VALUE},
 ")
```

7.5.2 Factor Constraints

```
FACTOR:         ANY-PRESENTATION-STYLE {
REQ    Presentation-style-identifier          {ANY_VALUE},
PERM   User-readable-comments                 {ANY_VALUE},
PERM   User-visible-name                      {ANY_VALUE},
}
```

7.5.3 Constituent Constraints

7.5.3.1 PStyle3

```
PStyle3         :ANY-PRESENTATION-STYLE  {

REQ    Content-architecture-class             {$FPR},
PERM   Presentation-attributes                {$R-Pres-Attr},
}
```

7.6 Content Portion Constraints

7.6.1 Raster Graphics Content Portion

```
DEFINE(T6,                                    "ASN.1 {2 8 3 7 0}")
DEFINE(Bitmap,                                "ASN.1 {2 8 3 7 3}")
DEFINE(Tiled,                                 "ASN.1 {2 8 3 7 5}")

PERM   Content-identifier-layout              {CONTENT_ID_OF(raster-content-portion)},
PERM   Type-of-coding                         {$T6 | $Bitmap | $Tiled},
PERM   Coding-attributes                      {
PERM   #Number-of-lines                       {ANY_VALUE},
REQ    #Number-of-pels-per-line               {ANY_VALUE},
PERM   #Number-of-pels-per-tile-line          {512},
PERM   #Number-of-lines-per-tile              {512},
PERM   #Tiling-offset                         {ANY_VALUE},
PERM   #Tile-types                            {'null background' |
                                              'null foreground' |
                                              'T.6 encoded' |
                                              'bitmap encoded'}
                                              },
PERM   Content-information                    {RASTER},
```

7.7 Additional Usage Constraints

No other usage constraints are currently defined.

8 INTERCHANGE FORMAT

Two interchange formats are supported by this profile. The Interchange Format Class A can be used by applications requiring a binary encoding based on ASN.1. The Interchange Format SDIF can be used by applications requiring a SGML based clear text encoding. This latter interchange format is an SGML application, called Office Document Language (ODL). For the purposes of interchange, the ODL ENTITIES are placed in an ASN.1 wrapper, as defined by SDIF. Each encoding form has inherent advantages. Conversion of document encoded in one interchange format into the other should not produce the loss of semantic document information.

8.1 Interchange format class A

8.1.1 Interchange format

The value of the document profile attribute "interchange format" for this interchange format is "if-b". This form of ODIF is defined in ISO 8613-5.

The encoding is in accordance with the Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), as defined in ISO 8825.

8.1.2 DAP identifier

The value for the document profile attribute "Document application profile" for this interchange format is represented by the following object identifier.

Editor's Note: To be supplied.

8.1.3 ASN.1 Generation Constraints

The following are additional constraints imposed on the ASN.1 generation beyond those defined in ISO 8824 and ISO 8825.

8.1.4 Encoding of Application Comments

ISO 8613-5 define the encoding of the attribute Application Comments as an octet string. This DAP requires that the encoding within that octet string be in accordance with the ASN.1 syntax specified in the following module definition.

```
NISTDAPSpecification
```

```
DEFINITION                          ::=      BEGIN
EXPORTS Object-Appl-Comm-Encoding;

Object-Appl-Comm-Encoding  ::=  IMPLICIT SEQUENCE OF
                                INTEGER
END
```

8.2 Encoding of Raster Content Information

The encoding of raster content information in the bitmap encoding scheme is that specified in subclause 9.3 of the raster graphics content architecture part of ISO 8613-7, that is, the first pel in the order of bits is allocated to the most significant bit of an octet. The encoding of the code words in the Group 4 facsimile encoding scheme is such that the first or only bit of the first code word shall be placed in the least significant bit of the first octet. Subsequent bits of the first and following code words are placed in the direction of more significant bits in the first and following octets.

8.3 Interchange format SDIF

8.3.1 Interchange format

The document profile attribute "Interchange format" does not apply for this interchange format. This form of ODIF is defined in Annex E of ISO 8613-5. In addition, ISO 8613-6, -7, and -8 contain additional specifications for this form of ODIF.

8.3.2 DAP identifier

The value for this attribute "Document application profile" for this interchange format is represented by the following object identifier.

Editor's Note: To be supplied.

8.3.3 Encoding of application comments

The encoding of the attribute "Application comments" is defined in a data stream conforming to this profile with the following DTD definition:

```
<!DOCTYPE odaac [
<!--
<!DOCTYPE doc PUBLIC "-//USA-OIW//SGML ENCODED ODA APPLICATION COMMENTS//EN">
-->

<!ELEMENT objappc   - O (#RCDATA)>
        <!-- Object application comment -->
]>
```

Editor's Note: The above DTD definitions must be verified by a SGML expert and modified as required.

ANNEX: A (NORMATIVE) Draft Addendum to ISO 8613-7

Annex A is the Draft Addendum to ISO 8613-7, Tiled Raster Graphics Addendum, dated January 1990.

Editor's Note: To be supplied as a separate document.

ANNEX: B (INFORMATIVE) Errata of Changes

This September 1990 (Working) document incorporates all the changes approved at the September 1990 ODA SIG meeting. A summary of these changes are listed below.

- A technical change to add an option for using SGML/SDIF based data stream encoding. These required changes to the following clauses (subclauses): 0, 2, 5.1, 6.7 (table 5), 7.1.2.1, and 8.
- Editorial changes to clauses 0-6 resulting from a preliminary DoD review and comment period.
- Editorial changes to clauses 7 and 8 resulting from review and comments by ODA SIG members at the September 1990 meeting.

Table of Contents

23 REFERENCES

Editor's Note: In this document, references are maintained in the individual sections as appropriate. Additional references for all of the subject covered in this document may be found in the aligned references section of the Stable Implementation Agreements Document, Version 3 dated September 1990.

READER RESPONSE FORM

Please retain my name for the next mailing of the NIST/OSI Implementors Workshop.

NAME: ______________________________

ADDRESS: ______________________________

____

____

PHONE NO.:______________________________

Mail this page to:

National Institute of Standards and Technology
NIST Workshop for Implementors of OSI
Brenda Gray, Registrar
Building 225, Mail Stop B-217
Gaithersburg, MD 20899

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990— 281-557/20735

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4448

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
NOVEMBER 1990

4. TITLE AND SUBTITLE

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS--SEPTEMBER, 1990

5. AUTHOR(S)

TIM BOLAND, Editor

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

□ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document records Working Agreements on Implementation details of Open Systems Interconnection Protocols among the organizations participating in the NIST/OSI Workshop Series for Implementors of OSI Protocols. These decisions are documented to facilitate organizations in their understanding of the status of agreements. This is a standing document that is updated after each workshop (about 4 times a year).

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

NIST/OSI WORKSHOP, LOCAL AREA NETWORKS: NETWORK PROTOCOLS: OPEN SYSTEMS INTERCONNECTION:

13. AVAILABILITY

☒ UNLIMITED
□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
612

15. PRICE
A99

ELECTRONIC FORM