NIST Special Publication 500-177

Computer
Systems
Technology

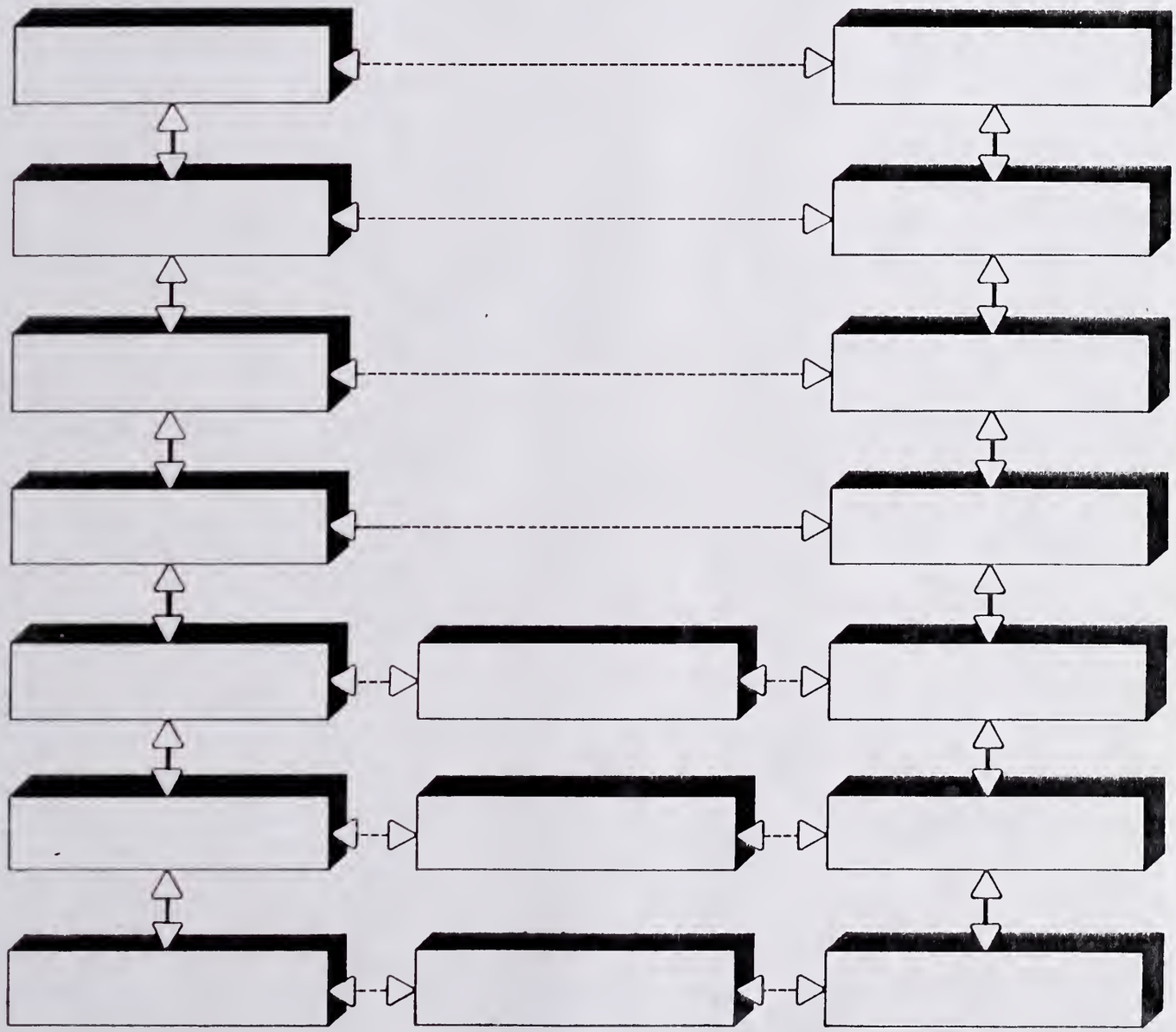
U.S. DEPARTMENT OF
COMMERCE
National Institute of
Standards and
Technology

NIST

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 3 Edition 1 December 1989

Based on the Proceedings of the NIST Workshop for Implementors of OSI

Workshop Chairman
Tim Boland, NIST



NIST Special Publication 500-177

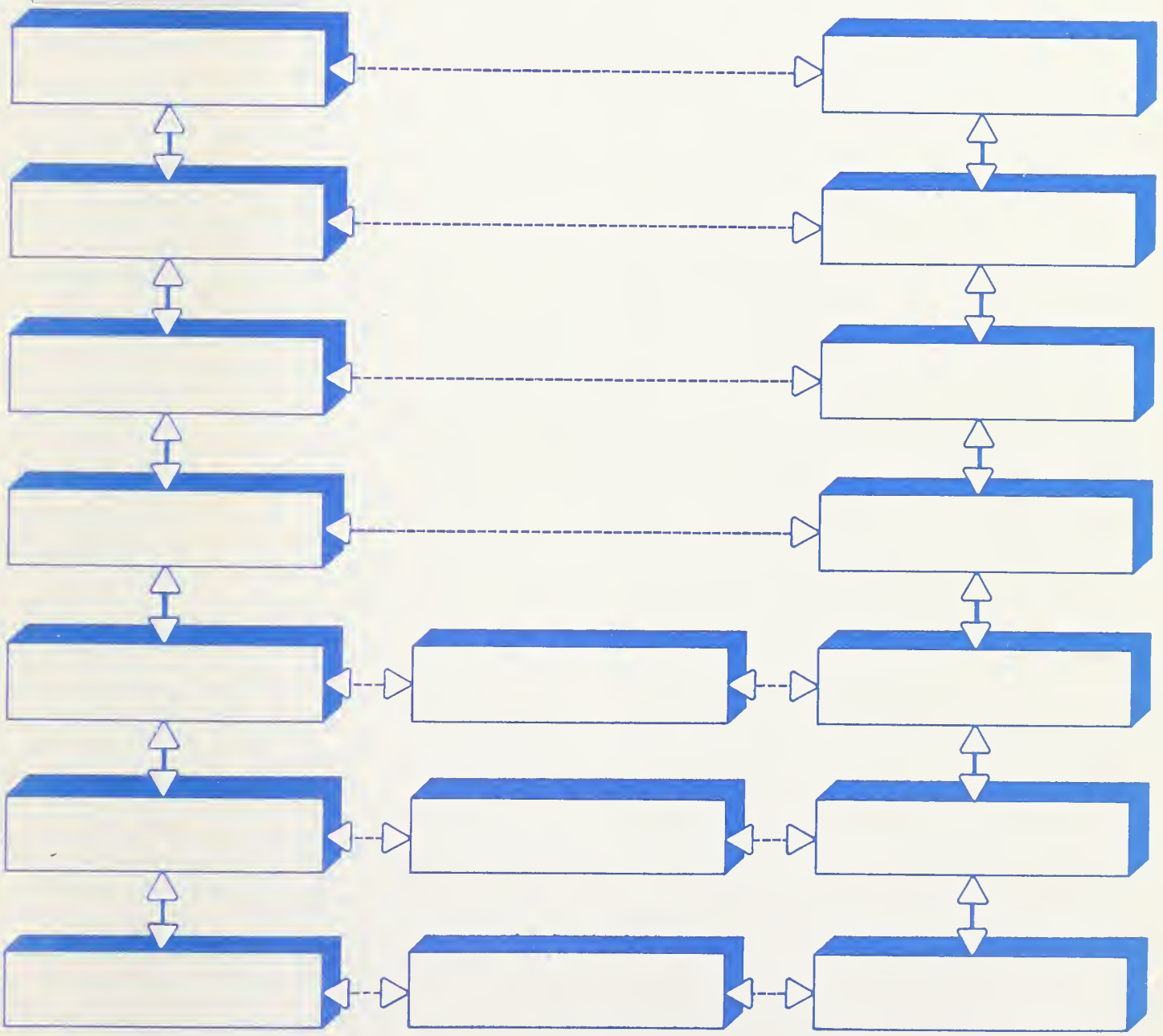
Computer Systems Technology

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 3 Edition 1 December 1989

Based on the Proceedings of the NIST Workshop for Implementors of OSI

Workshop Chairman
Tim Boland, NIST



NIST Special Publication 500-177

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 3 Edition 1 December 1989

Based on the Proceedings of the NIST Workshop for Implementors of OSI

Workshop Chairman
Tim Boland, NIST

Special Interest Group (SIG) Chairs

Directory Services	Chris Moore	Touch Communications
FTAM	Klaus Truoel	DFN
Lower Layers	Fred Burg	AT&T
Manufacturing Message Spec.	Herbert Falk	SISCO
Network Management	Paul Brusil	MITRE
ODA	Frank Dawson	IBM
Registration Authority	Einard Stefferud	NMA-Northrup
Remote Database Access	Richard Gerhardt	General Motors
Security	James M. Galvin	Trusted Info. Sys.
Transaction Processing	Andrew P. Schwartz	IBM Corp.
Upper Layers	David Chappell	Cray Research
Virtual Terminal	Cyndi Jung	3COM
X400	Barbara Nelson	RETIX

Supersedes NIST/SP-500/162

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

Issued March 1990

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) (formerly the National Bureau of Standards) has a unique responsibility for computer systems technology within the Federal government. NIST's National Computer Systems Laboratory (NCSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. NCSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. NCSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports NCSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-177
Natl. Inst. Stand. Technol. Spec. Publ. 500-177, 679 pages (Mar. 1990)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1990

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

Table of Contents

List of Figures

List of Tables

1. GENERAL INFORMATION

1.1 PURPOSE OF THIS DOCUMENT

This document records current stable implementation agreements of OSI protocols among the organizations participating in the NIST Workshop for Implementors of OSI. Stable in the context of this document means that:

1) The agreements are based on final standards (e.g., ISO-IS or CCITT Recommendations) or nearly final (e.g., ISO-DIS) with no significant changes expected, and,

2) The agreements have been approved by the NIST Workshop Plenary for progression from the Working Agreements document to this document after a period of review. These agreements are considered final; the only changes allowed will be clarifications, and certain Technical and Alignment errata. These changes must have the strong support of vendors, and be justifiable.

For these reasons, the agreements are considered advanced enough for use in product and test suite development. This means that readers can use this text as a basis for procurement references for OSI products. All of the text in this document is considered stable as defined above.

Future releases of these Stable Agreements will add and/or extend functionality offered by this edition and version. When required, new versions will be introduced on a yearly basis. It is the NIST Workshop intent that new versions of this Stable Agreements document will be compatible with the present version. If this proves impractical, the agreements will attempt to provide mechanisms and guidelines which maximize interoperability. Furthermore, it is the intent that these stable agreements be maintained via the Errata process as long as is appropriate. For the subject area, interworking information and other useful advice to the reader is given as appropriate. Specific "defect report" information (including extent of applicability) is provided in the ERRATA portions of each Chapter.

Agreements text is either in this Stable Document (Stable) or in the aligned Working Document (not yet stable). It is a goal that the same text not appear in the same position in both documents at once (except for sec. 1). New editions of a version reflect very recent stable functionality as well as editorial, technical, and alignment errata, all applied to the previous edition.

The intended audience for this document is composed of those individuals who are interested in Stable Implementation Agreements for OSI protocols. Each section of the document covers a different subject area, and the sections are presented so as to present a consistent and unified approach. The structure of each section, whenever possible, is divided into the following subsections:

- Introduction,
- Scope and Field of Application,
- Status,
- Errata,
- Protocol and Service Agreements,
- Conformance, and
- Appendices.

The corresponding and aligned document, "Working Implementation Agreements for OSI Protocols dated September 1990," records agreements which are not yet considered stable, in the sense described above. This document will be referenced as the "Working Agreements Document." This Stable document is aligned with the Working Agreements Document in the sense that the structures are identical, and pointers are given in this Stable Document to work in the Working Agreements Document which could become stable in the future.

The benefit of this document to the reader is that it gives a complete accounting of current stable agreements. Minor changes (Errata) to these agreements will be issued in replacement page format as separate editions. These errata will only be applied to the current version.

Version 3 (this version) is backwards compatible with Version 2 to the maximum extent possible. Version 3 includes all of the material from Version 2, (modified by errata) as well as now stable material from the previous year; important new functional additions from Version 2 are in the areas of certain aspects of Lower Layers network technology, X.3 Virtual Terminal profile, Workshop Registration procedures, new Upper Layer and Directory Services agreements, and new FTAM and 1988-Based X.400 agreements.

1.2 PURPOSE OF THE WORKSHOP

In February, 1983, at the request of industry, NIST 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.

1.3 WORKSHOP ORGANIZATION

The Workshop organizes its work through Special Interest Groups (SIGs) that prepare technical documentation. An executive committee of SIG chairpersons led by the overall Workshop chairperson administers the Workshop. NIST invites highly qualified technical leaders from participating organizations to assume leadership roles in the SIGs. The SIGs are encouraged to coordinate with standards organizations and user groups, and to seek widespread technical consensus on implementation agreements through international discussions and liaison activities.

The Workshop meets four times a year at the National Institute of Standards and Technology in Gaithersburg, Maryland where each SIG is required to convene its meeting. In addition, a plenary assembly of all Workshop delegates is convened for consideration of SIG motions and other Workshop business. SIGs are also encouraged to hold interim meetings at varied locations around the world.

The Workshop is an open public forum. Registration materials, documents, and Workshop schedules are available from:

National Institute of Standards and Technology
NIST Workshop for Implementors of OSI
Building 225, Room B-217
Gaithersburg, Maryland 20899

1.4 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 Workshop and associated agreements have been endorsed by various activities and groups. See the aligned section of the Working Agreements Document for more on this subject.

1.5 RELATIONSHIP OF THE WORKSHOP TO THE NIST

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. The NIST organizes, administers, and makes technical contributions to the Workshop. The NIST bears no other relation to the workshop.

1.6 STRUCTURE AND OPERATION OF WORKSHOP

1.6.1 Plenary

The main body of the workshop is a Plenary Assembly. Any organization may participate. Representation is international. The NIST prefers for the business of Workshops to be conducted informally since there are no corresponding formal commitments within the Workshop to implement the decisions reached. For more information, consult the aligned section of the Working Agreements Document.

1.6.2 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 during the Workshop week. As technical work is completed by a SIG, it is presented to the Plenary for disposition. For more information on SIGs (including SIG charters), consult the aligned section of the Working Agreements Document.

1.7 POINTS OF CONTACT

For information concerning the workshop, write to:

> Chair, NIST Workshop for Implementors of OSI
> at the address given in section 1.3.

Individual points of contact are given in the aligned section of the Working Agreements Document.

1.8 PROFILE CONFORMANCE

This section presents general concepts for profile conformance. These concepts shall be observed when writing Implementation Agreements.

1.8.1 General Principle

Conformance to an OSI Profile (Implementation Agreements, Functional Standards) implies conformance to the referenced Base Standards.

Therefore, a Profile shall not specify any requirements that would contradict or cause non-conformance to the Base Standards to which it refers (see TR 10000-1, clauses 6.1, 6.3.1). The conformance requirements defined in ISO/IEC TR 10000-1 fully apply.

1.8.2 Constraints

Base standards usually provide options for PDUs, parameters, encoding choices, value ranges, etc.

A profile may make specific choices of these options and ranges of values. For the promotion of interoperability, pragmatic constraints or minimum requirements may be imposed (e.g., the limitation of Search operations, selection of encoding choices, value ranges, byte ranges for encoding). These minimum requirements of restrictions shall not contradict the conformance requirements of the respective base standards.

1.8.2.1 Sending/Encoding Entity

In order to promote interworking, reasonable restrictions or minimum requirements may be specified in a profile as described above.

1.8.2.2 Receiving/Decoding Entity

Minimum requirements of receiving/decoding capability for alternatives, permissible values, etc. may be specified in a profile. A profile shall not specify the behavior of a receiving/decoding entity when receiving data which is outside the scope of or excluded by the Profile for senders.

A Profile Conformance Test shall be limited by the scope of the profile specification and shall not probe beyond its boundaries. That means, the capability of a receiver/decoder would be tested only in the range of choices or values which are specified for the sending/encoding entity (i.e., for interworking between systems both being conformant to the Profiles).

1.8.3 Classification of Conformance

Conformance requirements of a profile shall be related to conformance requirements of a base standard as written in clause 6.5 and annnex C of ISO/IEC TR 10000-1. For the conformance classes, the following terminology shall be used unless otherwise specified by the base standard or equivalent conformance requirements for a profile as required by the ISO/IEC Technical Committee that is responsible for the base standard:

m	mandatory
o	optional
c	conditional
x	excluded
i	out of scope
-	not applicable

2. SUB NETWORKS

Editor's Note: The term "Ongoing" used in this document refers to the Working Implementation Agreements dated September 1990.

2.1 INTRODUCTION

This chapter provides agreements about subnetwork services used in support of the OSI Network Layer.

2.2 SCOPE AND FIELD OF APPLICATION

These agreements cover subnetwork types including local area networks, packet switched networks, circuit switched networks, ISDN, and others.

2.3 STATUS

This version was completed in March 1990. Updated material has been moved into section 2.5.2 from the Working Implementation Agreements Document to replace existing text.

2.4 ERRATA

2.5 LOCAL AREA NETWORKS

2.5.1 IEEE 802.2 LOGICAL LINK CONTROL

The following decisions have been reached with respect to this protocol.

1. Link Service Access Point (LSAP)

   The code below shall be used to address systems using Network Layer protocols identified by ISO TR 9577 (e.g., ISO 8473, ISO 8208). Note that bit zero is transmitted first.

   The most significant bit is bit 7, thus this bit pattern represents hexadecimal FE.

0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1

Figure 2.1. LSAP bit pattern.

2. Type and Class

Only the connectionless type 1, class 1 IEEE 802 link service will be used.

2.5.2 IEEE 802.3 CSMA/CD ACCESS METHOD

The following implementation agreements have been reached with respect to the IEEE 802.3 CSMA/CD Access Method and Physical Layer Specifications:

- o The address length shall be 48 bits
- o For a data packet of LLC data length of n octets, the length of the pad field is

  max(0, minFrameSize-(8n+2(addressSize)+48)) bits.

The following implementation agreements have been reached with respect to 10 BROAD 36 Networks:

1. Single Cable Networks

   - The translator frequency shall be 192.25 Mhz
   - The channel allocations are

Reverse Channels	Forward Channels
T12, T13, T14	L, M, N
T13, T14, 2'	M, N, O
T14, 2', 3'	N, O, P
2', 3', 4'	O, P, Q
3', 4', 4A'	P, Q, R
4', 4A', 5'	Q, R, S

2. Dual Cable Networks

For nontranslated dual cable networks forward and reverse frequencies are the same. Permissible channel allocations are:

T12, T13, T14
T13, T14, 2'
T14, 2', 3'
2', 3', 4'
3', 4', 4A'
4', 4A', 5'
L, M, N
M, N, O
N, O, P
O, P, Q
Q, R, S

3. When both IEEE 802.4 and IEEE 802.3 10 BROAD 36 networks coexist on the broadband cable system the preferred channel allocations are:

	Reverse	Forward
IEEE 802.3	T12, T13, T14	L, M, N
IEEE 802.4	6', FM1'	T, U
channels reserved for future use	3', 4'	P, Q
	4A', 5'	R, S

2.5.3 IEEE 802.4 TOKEN BUS ACCESS METHOD

The following options are agreed to with respect to Draft J of token bus:

- Data Rate:
  - 10 Mb (Broadband)
  - 5 Mb (Carrierband)
- Addressing: 48 bit
- The lmeOption, Priority Mechanism, shall be implemented

- Broadband Channel Assignments

Forward	Reverse
P	3'
Q	4'
R	4A'
S	5'
T	6'
U	FM1'

2.5.4 IEEE 802.5 TOKEN RING ACCESS METHOD

The following implementation agreements have been reached with respect to the IEEE Standard 802.5, Token Passing Ring Access Method and Physical Layer specification.

- The data signalling rate shall be 4 Mbit/s
- The address length shall be 48 bits
- The message priority (PM) of the AMP data unit shall be 7
- The ALL_STATIONS_THIS_RING_ADDRESS shall be X'COOOFFFFFFFF'
- The TRR value shall be 4 milliseconds
- The THT value shall be 8.9 milliseconds
- The TQP value shall be 20 milliseconds
- The TVX value shall be 10 milliseconds
- The TNT value shall be 2.6 milliseconds
- The TAM value shall be 7 seconds
- The TSM value shall be 15 seconds

- The MAC Information field (I-field) shall be defined as follows:

Starting Sequence	I-Field	End Sequence

and the:

1) Starting Sequence includes: SD, AC, FC, DA, SA
2) Ending Sequence includes: FCS, ED, FS

Figure 2.2. I-Field Format.

- With the above timer and MAC I-field definitions, the following limits are defined:

  - Protocol limits the I-field to a maximum of 4425 bytes, and

  - All stations shall support I-fields that have a minimum of one byte and a maximum of at least 2000 bytes.

2.5.5 FIBER DISTRIBUTED DATA INTERFACE (FDDI)

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

(Refer to the Ongoing Implementation Agreements Document)

2.5.5.2 TOKEN RING PHYSICAL LEVEL (PHY,X3.148-1988)

(Refer to the Ongoing Implementation Agreements Document)

2.5.5.3 PHYSICAL LAYER MEDIA DEPENDENT (PMD, X3.166-1989)

(Refer to the Ongoing Implementation Agreements Document)

2.6 WIDE AREA NETWORKS

2.6.1 CCITT RECOMMENDATION X.25

The procedures required to describe the DTE side of a DTE/DCE interface for systems attached to sub-networks providing an X.25 interface shall be as defined in ISO 7776 and ISO 8208 and as supplemented below. (These procedures shall also apply to a DTE operating on a DTE/DTE interface).

2.6.2 ISO 7776

ISO 7776 is used as the Layer 2 Protocol with the agreements defined below.

1 The address assignments are:

DTE = A (=11000000 binary)
DCE = B (=10000000 binary)

On a DTE/DTE interface, one of the DTEs, by a prior agreement, shall use the DCE address.

2 The modulus shall be 8.

3 A window size (k) of 7 shall be supported. In addition, other window sizes may also be supported.

4 The Multilink Procedures are excluded.

2.6.3 ISO 8208

The elements of ISO 8208 applicable for use depend on the OSI role of ISO 8208 (ie., provision of CONS, support of CLNP). Independent of the role, ISO 8208 is used as the Layer 3 protocol, with the following agreements:

1 Virtual Call Service,

2 any mutually agreed window and packet size, however, all DTEs must be capable of supporting a window size of 2, a packet size of 128 octets, and a sequence number modulus of 8,

3 a DTE must be capable of receiving the Flow Control Parameter Negotiation Facility and responding appropriately (per ISO 8208), and

4 The Basic RPOA Selection Facility shall be implemented and its use or non-use selectable on a per virtual call basis.

When ISO 8208 is used to support CONS, the optional user facilities in Clause 5.1 of ISO 8878 shall also be supported.

When ISO 8208 is used to support CLNP (when providing the CLNS), Permanent Virtual Circuit Service may also be used.

2.7 INTEGRATED SERVICES DIGITAL NETWORKS (ISDN)

2.7.1 Introduction

This section defines Implementation Agreements for packet-data transfer in an ISDN context. The agreements provide a set of procedures for accessing an ISDN so that end systems implemented according to these agreements can obtain ISDN services and can successfully interoperate.

The agreements are not meant to preclude vendors from implementing additional procedures as long as they do not create system interoperability problems. Capabilities will vary from ISDN to ISDN and procedures beyond those included here may be necessary to request and utilize network services more effectively and fully.

The agreements cover two fundamental ISDN services for X.25 packet mode ISDN terminals, namely,

CASE I: The ISDN provides a circuit-mode (Layer 1) connection either on demand ("switched") or permanently ("dedicated circuit"). A general description of the corresponding ISDN 64 Kbps circuit-mode bearer service is described in CCITT Recommendation I.231. The circuit-mode connection is between an X.25 ISDN terminal and (i) a PSPDN, or (ii) another X.25 ISDN terminal. The circuit-mode connection to a PSPDN corresponds to CASE A of CCITT Recommendation X.31.

CASE II: The ISDN provides the X.25 virtual circuit service. A general description of this service is given in CCITT Recommendation I.232. This case corresponds to CASE B of CCITT Recommendation X.31.

Figures 2.3 and 2.4 give the agreed stacks for X.25 packet transfer over D and B channels, respectively. Some particular aspects are given below.

1. The packet data transfer is on a B channel of a Basic Access or a Primary Rate Interface. In CASE II, it can be on a D channel of a Basic Access Interface.

2. The layer 2 procedures are LAPB (ISO 7776) on a B channel and LAPD (CCITT Recommendation Q.921) on a D channel.

3. X.25 PLP (ISO 8208) procedures are used, including the setting up and clearing of virtual calls.

4. Q.921 and Q.931 procedures on a D channel are used for access signaling, when appropriate, to select the B or D channel for packet data transfer and for establishing and releasing a physical path in the ISDN.

5. Refer to chapter 3 for the specification of methods for providing OSI Network Services.

2.7.2 Implementation Agreements

This section gives Implementation Agreements for individual ISDN-related protocols. The relevant protocol stacks are given in figures 2.3 and 2.4.

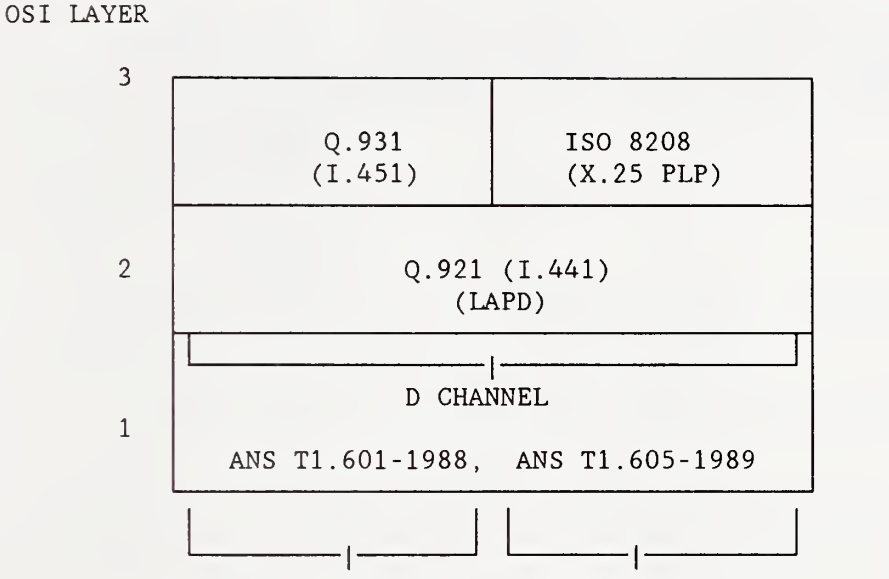


Figure 2.3. Protocol Layers at S, T and U reference points when D Channel is used in ISDN.

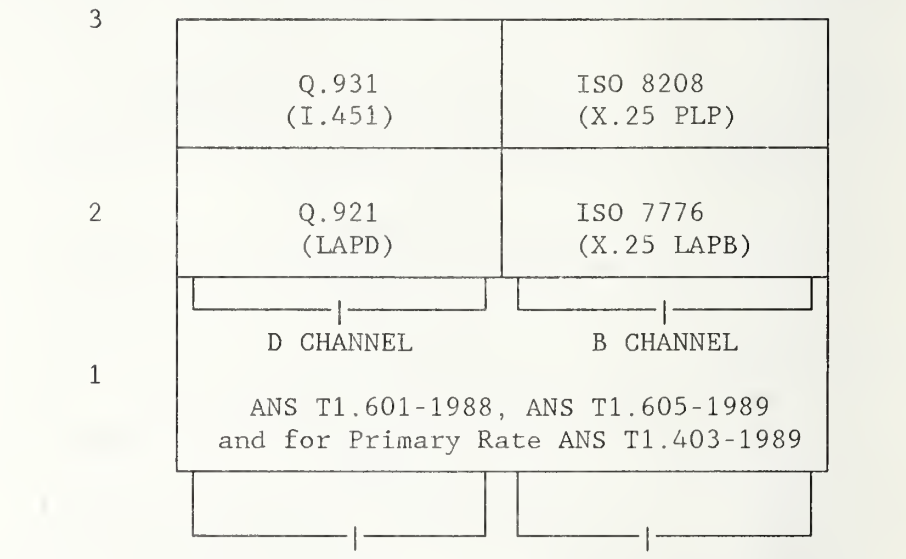


Figure 2.4. Protocol Layers at S, T and U reference points when B Channel is used in ISDN.

2.7.2.1 Physical Layer, Basic Access at "U"

ANS T1.601-1988, "Integrated Services Digital Network-Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT-Layer 1 Specification" applies.

2.7.2.2 Physical Layer, Basic Access at S and T

ANS T1.605-1989, "Integrated Services Digital Network-Basic Access Interface at S and T Reference Points-Layer 1 Specifications" applies.

2.7.2.3 Physical Layer, Primary Rate at "U"

The physical layer is governed[1] by ANS T1.403-1989, "Carrier-to-Customer Installation Metallic Interface," and CCITT Recommendation I.431-1988, "Primary Rate User-Network Interface - Layer 1 Specification subject to the exceptions given below.

The following portions of ANS T1.403.1989 shall be deleted.

- Section 5.3.1: The bit rate tolerance option of +/- 200 bps.
- Section 5.6: The minimum pulse density of this section.
- Section 6.1: The superframe format.
- Section 6.3: The complete section.
- Section 8.0: The reference to the SF format.
- Section 8.3: The text in paragraph 8.3.1.1 and footnote 7 (8.3.1.2).
- Section 8.4.1: Footnote 9.
- Section 9/ figure 9: Provisions for the use of the RJ48M connector.
- Table 2: This table.
- Table 3: The illustration in table 3 of "Robbed-Bit Signaling."

---

1 ANSI accredited subcommittee T1E1 is developing a standard for the ISDN primary rate interface at reference points "S" and "T" as well as "U". One of the accepted guidelines for the standard is consistency with ANS T1.403-1989. It is intended that, when this new ISDN-unique standard is adopted, this agreement will be modified to reference it and will be extended to cover interfaces at reference points "S" and "T" as well as "U". The current standard mentioned is at the default letter ballot level of the draft document: T1E1-2/89-046R4, "ISDN Primary Rate Customer Installation Interfaces Layer 1 Specification."

The following portions of ANS T1.403-1989 shall be modified :

- Section 5.3.2 — The text of this section is replaced by:

  "The line code is B8ZS except as noted in section 7," and

- Section 7.0: — The reference to the pulse density requirements of section 5.6 is inappropriate. The text is replaced by:

  "The provisions of Clear Channel Capability (CCC) depends upon the use of the B8ZS line code, though the use of ZBTSI is one interim method that may be employed by agreement of the network and the user"

The provisions of ANS T1.403-1989 shall be supplemented by the provisions of CCITT Recommendation I.431 - section 4.4.

2.7.2.4 Data Link Layer, D-Channel

CCITT Recommendation Q.921 (I.441), "ISDN User-Network Interface Data Link Layer Specification" applies.

2.7.2.5 Signaling

CCITT Recommendations Q.931 (I.451), "ISDN User-Network Interface Layer 3 Specification" applies.

The following agreements have been reached concerning the use of Q.931.

1 On a Basic Rate Interface supporting the ISDN virtual circuit service, all of Q.931 section 6 except for 6.1.1 and 6.2.1 (the sections covering the circuit-switched access case) shall apply. The following sections also apply: 2.2, packet mode access connection states; 3.2, messages for packet mode access connection control; 4-4.5, section specifying general information element

handling and encoding; 4.7, information elements for packet communications.

2 On a Primary Rate Interface supporting the ISDN virtual circuit service all of Q.931 section 6 shall apply except for 6.1.1 and 6.2.1 (the sections specifying the circuit switched access case), 6.1.2.2, 6.2.2.2 and 6.4.2 (the sections specifying D-Channel ISDN Virtual Circuit service case). The following sections also apply: 2.2, packet mode access connection states; 3.2 messages for packet mode access connection control; 4-4.5, sections specifying general information element handling and encoding; 4.7, information elements for packet communications.

3 On a Basic or Primary Rate Interface supporting the unrestricted 64-Kbps circuit-mode service, Q.931 sections 6.1.1, 6.2.1, 6.4.1 and 6.4.3 shall apply. The following sections also apply: 2.1, circuit mode connection states; 3.1, messages for circuit mode connection control; 4-4.5, sections specifying general information element handling and encoding.

2.7.2.6 Data Link Layer B-Channel

The agreements on ISO 7776 specified in section 2.6.2 shall apply here.

If the ISDN provides a circuit-mode service between two ISDN packet-mode devices, then the layer 2 address shall be assigned as follows:

1 For permanent ("non-switched") circuit-mode service, one terminal uses address A and the other terminal uses address B, as arranged by prior agreement, and

2 For demand ("switched") circuit-mode service, the terminal originating the circuit-mode call uses address A and the other terminal uses address B.

2.7.2.7 Packet Layer

The agreements on ISO 8208 specified in section 2.6.3 shall apply here. When ISO 8208 is used on the D-Channel, the maximum DATA packet size (i.e., actually the maximum size of the User Data Field in a DATA packet) shall be limited to 256 octets.

2.8 APPENDIX A

This appendix provides a cross-reference listing between those sections of the CCITT ISDN Recommendations given in section 2.7 of this document and the sections of the corresponding ANSI ISDN Standards. This appendix does not supersede section 2.7 but merely provides a pointer to those who wish to implement the ISDN procedures based on ANSI Standards.

2.8.1 Data Link Layer, D-Channel

CCITT Recommendation Q.921, which is referenced in section 2.7.2.4 of this document, is identical to ANSI Standard T1.602.

2.8.2 Signaling

The following table provides the cross-references between those sections of CCITT Recommendation Q.931 referenced in section 2.7.2.5 of this document and the corresponding ANSI ISDN Standards.

Table 2.1. ANSI-CCITT Cross-References

CCITT RECOMMENDATION Q.931	ANSI T1.608
Section 2.1	Section 4.1 (refers to sec. 2.1.1 of ANSI T1.607)
Section 2.2	Section 4.2
Section 3.1	Section 5.1 (refers to sec. 3 of ANSI T1.607)
Section 3.2	Section 5.2
Section 4.1	Section 6.1
Section 4.2	Section 6.2
Section 4.3	Section 6.3
Section 4.4	Section 6.4
Section 4.5	Section 6.5
Section 4.7	Section 6.5
Section 6	Section 7
Section 6.1.1	Section 7.1.1
Section 6.1.2.2	Section 7.1.2.2
Section 6.2.1	Section 7.2.1
Section 6.2.2.2	Section 7.2.2.3
Section 6.4.1	Section 7.4.1
Section 6.4.2	Section 7.4.2
Section 6.4.3	Section 7.4.3

2.9 Bibliography

Local Area Networks: Baseband Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Profiles and Link Layer Protocol, FIPS 107, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

3. NETWORK LAYER

Editor's Note: The term "Ongoing" used in this document refers to the Working Implementation Agreements dated September 1990.

3.1 INTRODUCTION

This chapter presents agreements for providing the OSI network service. Also contained here are agreements on network layer addressing and routing.

3.2 SCOPE AND FIELD OF APPLICATION

These agreements cover both connectionless-mode and connection-mode network services.

3.3 STATUS

This version of the agreements was completed in December 1989. Additional material in 3.8 has been included since then. Part of this material had been in 3.13 of the Working Document.

3.4 ERRATA

Editor's Note: This section may contain "Defect Report" and resolutions material, and the versions of implementor agreements to which this material aplplies.

The following defects are being progressed in ISO:

- ISO 9542, defects 1-12.

3.5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS)

3.5.1 ISO 8473

1. Subsets of the protocol:

   - o Implementations will not transmit PDUs encoded using the inactive subset. Received PDUs encoded using the inactive subset will be discarded.

   - o The non-segmenting subset will not be used. Implementations will not generate data PDUs without a segmentation part. However, implementations will

receive and correctly process PDUs which do not contain the segmentation part.

2. Mandatory Functions:

- The lifetime parameter shall be used as specified in Clause 6.4 of ISO 8473. The parameter shall have an initial value of at least three times the network span or three times the maximum transit delay (in units of 500 milliseconds), whichever is greater.

- The reassembly timer for an initial PDU at the reassembly point shall be no greater than the largest value of all lifetime parameters contained in all derived PDUs.

- The use/non-use of checksums shall be capable of being configured. The default setting shall be non-use.

- If the implementation supports the generation of an ER PDU, the system shall insert in the destination address field of the ER PDU the contents of the source address field of the PDU that generated the error.

3. Optional Functions:

- The Security parameter is not defined by these Agreements. Implementations shall not transmit the parameter except where defined by bilateral agreements.

- Partial and complete source routing will <u>not</u> be supported.[1]

- Partial record of route will be supported by Intermediate systems.

- ISO 8473 will be followed with respect to QOS.

---

[1] A defect exists with the Partial Source Routing option which can cause PDUs to loop in the network until their lifetime expires.

- For systems implementing the congestion notification function, the following applies.

  A Globally Unique QOS Maintenance parameter shall be included in all PDU originated by End Systems. As specified in ISO 8473, the initial value of the Congestion Experienced flag (CE flag) within the Globally Unique QOS Maintenance Parameter shall be set by the originating End System to zero. All other flags within the Globally Unique QOS Maintenance Parameter shall be set based on the specific local needs of the originating End System.

  Intermediate systems not implementing queue length averaging shall leave the CE flag in the same state as it was received. In particular, no intermediate system (IS) shall ever clear (set to zero) the CE flag. All intermediate systems shall monitor all incoming and outgoing queues and compute average queue lengths as shown by example in table 3.1. The averaging is done from the beginning of the previous cycle to the current time. A cycle begins at the instant of the first NSDU arrival after an idle period.

  An IS should set the CE flag in all NSDUs forwarded on a queue which has an average queue length greater than one.

  The queue length averaging algorithm computes the average queue length over two cycles, where the two cycles are:

  1) the "previous cycle," which is the interval from when the IS becomes busy, until it becomes idle and the idle ends (indicated by the instant the first packet arrives to the idle IS), and

  2) the "current cycle," which is the interval from the end of the idle interval to the current time instant when the average queue length is computed.

  An embodiment of the averaging algorithm is shown in table 3.1.

Table 3.1. Queue Length Averaging Algorithm

The algorithm makes use of the following variables:

t = Current time
t_i = time of i^{th} arrival or departure event
q_i = number of packets in the system after the event
T_0 = time at the beginning of the previous cycle
T_1 = time at the beginning of the current cycle

The algorithm consists of three components:

1. Queue Length Update: Beginning with $q_0 = 0$,
   If the i^{th} event is an arrival event, $q_i = q_{i-1}+1$
   If the i^{th} event is a departure event, $q_i = q_{i-1}-1$

2. Queue Area (integral) update:

$$\text{Area of the previous cycle} = \sum_{t_i \epsilon (T_0, T_1)} q_{i-1}(t_i - t_{i-1})$$

$$\text{Area of the current cycle} = \sum_{t_i \epsilon (T_1, t)} q_{i-1}(t_i - t_{i-1})$$

3. Average Queue Length Update:

$$\text{Average Queue length over the two cycles} = \frac{\text{Area of the two cycles}}{\text{Time of the two cycles}} = \frac{\text{Area of the two cycles}}{t - T_0}$$

- (Refer to the Ongoing Implementation Agreements document for additional optional functions)

3.5.2 Provision of CLNS over Local Area Networks (LANS)

When providing CLNS over a LAN subnetwork, the following shall apply:

1. The definition of CLNS shall be as specified in ISO 8348/Add. 1,

2. The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 3.5.1, and

3. The necessary subnetwork dependent convergence function

shall be as defined in ISO 8473 - Clause 8.5.1, SNDCF used with ISO 8802/2 sub-networks."

3.5.3 Provision of CLNS over X.25 Subnetworks

When providing CLNS over X.25 subnetworks, the following shall apply:

1. The definition of CLNS shall be as specified in ISO 8348/Add. 1,

2. The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 3.5.1, and

3. The necessary subnetwork dependent convergence function shall be as defined in ISO 8473 - Clause 8.5.2, "SNDCF used with ISO 8208 subnetworks for operation over X.25 subnetworks," The default throughput class shall be used if this facility is available, and

4. The X.25 PLP shall be as specified in section 2.6.3.

3.5.4 Provision of CLNS over ISDN

When providing CLNS over an ISDN, the following shall apply.

3.5.4.1 CLNP Utilizing X.25 Services

- The definition of CLNS shall be as specified in ISO 8348/Add. 1,

- The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in section 3.5.1, and

- The necessary Sub-network Dependent Convergence function shall be as defined in:

  - ISO 8473 for operation of CLNP over X.25 with agreements as specified in section 3.5.3, and

  - ISO 9574 for control of the B and D channels.

    Note: The stated scope of ISO 9574 does not explicitly cover the operation of CLNP over an ISDN. However, the procedure identified for operating X.25 in

conjunction with I.451 is still applicable. The procedures in ISO 9574 that correspond to 8878 are not utilized when providing CLNS.

- The X.25 PLP shall be as specified in section 2.6.3.
- The agreements for the ISDN-related protocols are specified in section 2.7.

3.5.5 PROVISION OF CLNS OVER POINT-TO-POINT LINKS

(Refer to the Ongoing Implementation Agreements document).

3.6 CONNECTION-MODE NETWORK SERVICE (CONS)

The following agreements concern provision of the connection-mode Network Service.

3.6.1 Mandatory Method of Providing CONS

3.6.1.1 General

Independent of the subnetwork type (of sec. 2), when providing the CONS using X.25-1984, the following shall apply as described below.

- The definition of the CONS is as specified in ISO 8348, Network Service Definition.
- The mapping of the elements of the CONS to the elements of the X.25 Packet Layer Protocol (PLP) is as specified in section 3.6.3.1.
- The general procedures and formats of the X.25 PLP are as specified in ISO 8208, X.25 Packet Layer Protocol for Data Terminal Equipment.

3.6.1.2 X.25 WAN

No provisions additional to those in section 3.6.1.1 apply in an X.25 WAN.

3.6.1.3 LANs

When providing the CONS in a Local Area Network, the following aspects of ISO 8881, in addition to the documents listed in section 3.6.1.1, shall apply:

- Clauses 1-6 and 9-11 for LLC Type 1 operation, including the additional nonstandard default packet size listed in Clause 6.3, Note 2

  Note: Operation of ISO 8208 in conjunction with LLC Type 2 requires agreement on LLC Type 2 procedures.

3.6.1.4 ISDN

When providing the CONS in an ISDN, the considerations for control of a B and D channel in ISO 9574, in addition to those provided in section 3.6.1.1, shall apply.

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

When providing CONS over an X.25 1980 subnetwork, the following shall apply:

- The definition of the CONS is as specified in ISO 8348, Network Service Definition, and

- The subnetwork dependent convergence protocol required to provide CONS shall be as specified in ISO 8878 Annex A, and referred to as the Alternative Procedures for Network Connection Establishment and Release, with agreements as defined in 3.6.3.2.

3.6.3 Agreements on Protocols

3.6.3.1 ISO 8878

ISO 8878 Clauses 1-11 shall apply with the following exception:

- Where the ISO 8208 diagnostic codes are not provided, all Cause/Diagnostic code combinations can be mapped to the Originator/Reason code of "Undefined."

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

- The Receipt Confirmation service will not be provided, so the corresponding protocol elements need not be implemented.

- The Expedited Data service will not be provided, so the corresponding protocol elements need not be implemented.

3.6.4 Interworking

Interworking between subnetworks whose End Systems use ISO 8208 to provide the CONS as specified in section 3.6.1 shall be performed as specified in ISO TR 10029. That is, an Intermediate System connecting two such subnetworks shall operate ISO 8208 on both subnetworks and shall relay information from one subnetwork to the other as described in ISO TR 10029.

3.7 ADDRESSING

NSAP address formats supported will conform to Addendum 2 of ISO 8348.

- NSAP address formats will have a hierarchical structure. This will reduce the size of routing tables.

- If used in the Domain Specific Part (DSP), an NSAP selector shall be the least significant component in the hierarchy, and shall be encoded as the last octet of the DSP. The NSAP selector shall not be used to perform routing; it is simply intended to identify the network service user at the destination end system. For those implementations using an NSAP selector, there shall be one and only one selector for each NSAP within the end system. All NSAP addresses identifying a given NSAP will use the same NSAP selector value.

3.8 ROUTING

The basic principles of Network Layer routing are defined in the OSI Routing Framework ISO TR 9575. These principles include:

- The global OSI environment will consist of a number of Administrative Domains. An Administrative Domain consists of a collection of End Systems (ES) and Intermediate Systems (ISs), and subnetworks operated by a single organization or Administrative Authority. The Administrative Authority is responsible for: the organization of ESs and ISs into Routing Domains; the assignments of NSAP and SNPA addresses; the policies

that govern resource usage; the policies that govern the information that is collected and disseminated both internally and externally to the Administrative Domain; and, the establishment of subdomains and the corresponding delegation of responsibilities.

- A Routing Domain is a set of ESs and ISs which operate according to the same routing procedures and which is wholly contained within a single Administrative Domain. An Administrative Authority may delegate to the entity responsible for a Routing Domain the responsibilities to further structure and assign NSAP and SNPA addresses. The hierarchical decomposition of Routing Domains into subdomains may greatly reduce the resources required in the maintenance, computation, and storage of routing information.

- The OSI routing problem, and consequently OSI routing protocols, has been decomposed into three distinct classes:

  - End System (ES) to Intermediate System (IS) routing within a single subnetwork.

  - IS to IS routing within a single routing domain (Intra-domain).

  - IS to IS routing between routing domains (Inter-domain).

~~The OSI routing problem has been decomposed into two distinct classes of routing:~~

~~- End Systems (ES) to Intermediate Systems (IS) routing;~~

~~- IS to IS routing;~~

3.8.1 End System to Intermediate System Routing

For use in conjunction with ISO 8473 over LANs (refer to sec. 2.5) and point-to-point links, ISO 9542 shall be used to provide the routing exchange protocol.

Additionally, a management mechanism capable of adding and deleting entries into the Routing Information Base (RIB) is recommended. When using the management mechanism to add an entry, there should be no holding timer, and the entry should be write protected from alteration by the ES-IS protocol. This mechanism enables routing table entries to be made which are static in nature.

The agreements below apply to the use of ISO 9542.

1. Implementors shall support any valid NSAP format. For the purposes of the protocol, NSAP addresses are treated simply as octet strings.

2. For LANs, implementors shall support both Configuration Information and Route Redirection Information; no subsets are permitted.

3. All timer values shall be configurable.

4. Use or non-use of checksums shall be configurable. It is recommended not to use ISO 9542 checksums when originating PDUs.

5. The QOS, Security and Priority parameters should not be used for routing. For conformance, intermediate systems must transmit these parameters in RD PDUs if they are present in the data PDU which generated the redirect. However, end systems must ignore them in received RD PDUs.

6. If the configuration notification function described in clause 6.7 of the protocol specification is implemented, a mechanism shall be provided to enable/disable this function on broadcast networks. If supported in end systems listening to both ISHs and ESHs, this function shall only be invoked upon receipt of an ISH.

   An alternative mechanism for achieving rapid configuration which is scaleable to large broadcast networks is described below. This mechanism makes use of the Suggested ES Configuration Timer. Implementation of this mechanism is optional.

   IS Actions
   ----

   a) IS Actions

   When an Intermediate system wants to quickly acquire the End system configuration (for example, when a broadcast circuit is enabled on the IS or the topology changes because of a failure of a bridge or repeater), it initiates a "poll" of the End system configuration by performing the following actions:

   1. Delay a random interval between 0 and PollESHelloRate seconds. (This is to avoid

synchronization with other ISs which have detected a change.)

2. In order to rapidly time out any End systems which are no longer present on the broadcast circuit (for example, after a LAN partition), reset the entryRemainingTime in the Routing Information Base for all End systems on this circuit to the value:

(ISHelloTimer + PollESHelloRate) * HoldingMultiplier

or the existing value whichever is lowest. Where ISHelloTimer is the Intermediate system's configuration timer, HoldingMultiplier is a predefined number (for example, 2) which multiplied by ISHelloTimer gives the value for the Holding Time field of IS Hellos.

3.. Then transmit HoldingMultiplier IS Hellos with a Suggested ES Configuration Timer value of PollESHelloRate seconds with an interval of ISHelloTimer seconds between each and setting the Holding Time field to ISHelloTimer * HoldingMultiplier.

4. Then start sending IS Hellos with a Suggested ES Configuration Timer of DefaultESHelloRate seconds (where DefaultESHelloRate is larger than PollESHelloRate).

b) ES Actions

An End system maintains for each circuit a list (CTList) which has HoldingMultiplier elements each of which stores a received value of the Suggested ES Configuration Timer. The function SaveCT(t) adds the value t as the first element of CTList and discards the last element. The function MinCT delivers the minimum value in CTList. When the circuit is enabled all the elements of CTList are initialized to PollESHelloRate.

An End system also maintains for each circuit the variables currentSuggestedHelloTimer and its associated lifetime currentSuggestedHelloTimerLifetime. These are both initialized to PollESHelloRate.

When the circuit is enabled the Configuration Timer is started by setting the entryRemainingTime to random (PollESHelloRate).

On Configuration Timer expiry the following actions are performed:

1. SaveCT(currentSuggestedHelloTimer).

2. Transmit an ES Hello with Holding Time field set to MinCT * HoldingMultiplier.

3. Set entryRemainingTime to MinCT - random(MinCT * 0.25). (The random element ensures that End systems do not become synchronized.)

When an End system receives an IS Hello which contains a Suggested ES Configuration Timer, it is processed as follows (where suggestedESCT is the value contained in the option):

1. If suggestedESCT is less than or equal to currentSuggestedHelloTimer then set curentSuggestedHelloTimerLifetime to the value of the Holding Time field of the IS Hello.

2. If suggestedESCT is less than currentSuggestedHelloTimer then set currentSuggestedHelloTimer to suggestedESCT and reset entryRemainingTime to the smaller of its current value and random(currentSuggestedHelloTimer * 0.75).

When the currentSuggestedHelloTimerLifetime expires, set the currentSuggestedHelloTimer to DefaultESHelloTimer.

7. For LANs, this protocol employs the same LSAP as ISO 8473.

8. The encoding of the BSNPA address follows the syntax rules for the data link being used. On a LAN, for example, it is a 48-bit MAC address.

9. The multicast addresses corresponding to "all intermediate systems on the network" (ALL_ISN) and "All End Systems on the Network" (ALL_ESN) shall default to the following on IEEE802.3 and IEEE802.4 subnetworks:

ALL_ESN = 0900 2B00 0004
ALL_ISN = 0900 2B00 0005

It is understood that the hexadecimal octets shown are transmitted onto the medium form left most octet to right most octet. Within each hexadecimal octet the least significant bit is transmitted first.

10. The multicast addresses corresponding to "All Intermediate Systems on the network" (ALL_ISN) and "All End systems on the Network" (ALL_ESN) shall default to the following two functional addresses on IEEE802.5 subnetworks:

    ALL_ESN = C000 0000 4000
    ALL_ISN - C000 0000 8000

    It is understood that the hexadecimal octets shown are transmitted onto the medium from left most octet to right most octet. Within each hexadecimal octet the most significant bit is transmitted first."

9. For LANs, the multicast addresses corresponding to "All Intermediate Systems on the network" (All_ISN) and "All End Systems on the network" (All_ESN) shall default to the following:

    All_ESN = 0900 2B00 0004
    All_ISN = 0900 2B00 0005

    Note: Version 1, Edition 1 of these agreements had the above addresses reversed (i.e., All_ISN was given as 0900 2B00 0004, etc.) because of an unintentional error in the original proposal. There is no intent to maintain compatibility with systems using Version 1, Edition 1 addresses.

    For IEEE 802.3 and 802.4 LANs, it is understood that the hexadecimal octets shown are transmitted onto the medium from left most octet to right most octet. Within each hexadecimal octet, the least significant bit is transmitted first.

11. The Error Report flag shall be set to zero (0) for NPDUs sent as a result of invoking the QUERY Configuration Function.

3.8.2 Intermediate Systems to Intermediate Systems Routing

Intermediate systems shall provide mechanisms to create and update the required Routing Information Base.

3.9 PROCEDURES FOR OSI NETWORK SERVICE/PROTOCOL IDENTIFICATION

3.9.1 General

The Protocol Identifiers specified in ISO DTR 9577 ("Protocol Identification in the OSI Network Layer") provide a basis from which OSI systems (both end systems and intermediate systems) may derive a set of procedures for indicating which OSI protocols are used in a particular instance of communication. As such, these procedures are only concerned with Initial Protocol Identifiers (IPIs) and Subsequent Protocol Identifiers (SPIs) that identify OSI protocols and pertain to the following types of systems:

A. systems providing/supporting only CONS (using ISO 8208/8878),
B. systems providing/supporting only CLNS (using ISO 8473), and
C. systems providing/supporting both CONS and CLNS.

From this set of definitions, the following possibilities for success (S) or failure (F) of an instance of communication can be defined, as shown in the table below:

Table 3.2. End Systems Communications

Originating End System Type	Destination End System Type A	B	C
A	S	F	S
B	F	S	S
C	S	S	S

3.9.2 Processing of Protocol Identifiers

The usage of Protocol Identifiers in Network Protocol Data Units (NPDUs) depends on several factors:

- the OSI Network Service to be provided,

- the protocol to be used in providing this service,
- the role the protocol is to be used in (per the Internal Organization of the Network Layer), and
- the type of subnetwork to which the system is connected.

3.9.2.1 Originating NPDUs

The use of a particular OSI Network Service depends on the capabilities of both the origination and destination end systems. It is not the intent of this section to provide guidelines on how to make this choice except for simple obvious criteria; rather, it is intended only to provide guidance on how to convey this choice to the destination system.

Where a priori knowledge exists in the originating end system about the capabilities (with respect to OSI Network Services available) of the destination end system, it should be used. This may result in no communication if the two end systems involved only provide Network Services of different types. A selection is required in cases where both end systems provide both types of network services; this selection is conveyed by the use of the IPI and SPI (but the selection process is an implementation matter). Alternatively, where a priori knowledge does not exist, then the selection of a service to use in an instance of communication depends solely on the capabilities of the originating end system as described below.

- If only CONS-related protocols (e.g., ISO 8208) are available, then this should be used and the Protocol Identifiers specified so as to reflect the chosen protocol(s) and service.

- If only CLNS-related protocols (e.g., 8473) are available, then this should be used and the Protocol Identifiers specified so as to reflect the chosen protocol(s) and service.

- If both services are available, then other criteria are used in deciding which to use in an instance of communication.

Note: The choice of OSI Network Service to be used in an instance of communication is reflected in the Network Service primitives issued by the Network Service user.

Once a selection of Network Service has been made, the use of particular protocols depend on, for example, the subnetwork to which the originating End System is attached. Some specific cases are given in Annex A of ISO DTR 9577. Another case involves use of the Protocol for Providing the Connectionless Network Service directly over the Data Link Service, as given in ISO 8473 (e.g., in a LAN). In this case, the IPI indicates ISO 8473.

3.9.2.2 Destination System Processing

A system receiving an NPDU must first be concerned with the protocol identified by the IPI. Valid values are given in table 2 of ISO DTR 9577. If the protocol is recognized as one supported by the system, further processing of the protocol is performed according to the rules of that protocol. If not, an error is recognized and may be conveyed to the originating peer entity. With respect to ISO 8208 and ISO 8473, the following would apply for such error conditions.

1. For ISO 8208, the condition is classified as an "invalid General Format Identifier," for which a DIAGNOSTIC packet may be returned. If DIAGNOSTIC packets are not used by the system, the NPDU is discarded without any further action.

2. For ISO 8473, the NPDU is discarded without any further action.

Given acceptance of the protocol identified by the IPI, the system must also determine the acceptability of the subsequent protocols and OSI Network Service being requested. Use of ISO 8473 implies CLNS; however, use of ISO 8208 can imply either CONS or CLNS, as identified by the SPI. In the case of ISO 8208, therefore, further processing is needed to determine the acceptability of the requested protocol/service. If these are not acceptable (e.g., not supported by the system), the call should be cleared with a diagnostic code of "Connection Rejection - unrecognizable protocol identifier in user data" (decimal 249).

Note: In ISO 8208, a call may be refused for reasons other than non-support of the requested OSI Network Service.

3.9.2.3 Further Processing in Originating End System

Further processing on receipt of an NPDU in response to an initial attempt to communicate may be necessary/useful to determine the success of such an attempt.

For ISO 8473, when used directly over the Data Link Service, the success or failure of an attempt to communicate may not be visible/obvious within the Network Layer. On the other hand, use of ISO 8473 over ISO 8208 may provide, via the diagnostic code in a received CLEAR INDICATION packet, an indication of failure to communicate (e.g., the remote system does not support CLNS).

When using ISO 8208 to provide the CONS, the diagnostic code in a received CLEAR INDICATION packet may provide the necessary indication of why a call was refused.In cases where an ISO 8208 call is refused with diagnostic #249, it would not be desirable to re-attempt such calls with the exact same set of parameters; however, how the originating system ensures this is a local matter.

In cases where an originating system is capable of supporting both OSI Network Services, it may wish to re-attempt communications using the other mode of Network Service than that initially attempted.

3.9.3 APPLICABLE PROTOCOL IDENTIFIERS

The protocol identifiers applicable to these agreements are given in table 3.1 and table 3.2.

Table 3.1. IPI Values

Bit Pattern 8	7	6	5	4	3	2	1	Protocol
0	0	0	0	1	0	0	0	CCITT I.451/Q.931
1	0	0	0	0	0	0	1	ISO 8473 (excluding the inactive subset)
1	0	0	0	0	0	1	0	ISO 9542
x	x	0	1	x	x	x		ISO 8208/CCITT X.25-Modulo 8
x	x	1	0	x	x	x	x	ISO 8208/CCITT X.25-Modulo 128
0	0	1	1	x	x	x	x	ISO 8208/CCITT X.25-GFI Extension

Table 3.2. SPI Values

Bit Pattern * 8	7	6	5	4	3	2	1	Protocol
0	0	0	0	0	0	0	0	ISO 8073 ADD1/CCITT X.224 See table 4.1
			thru					
0	0	1	1	1	1	1	1	
1	0	0	0	0	0	0	1	ISO 8473
1	0	0	0	0	1	0	0	ISO 8878/Annex A

* A null SPI value (e.g., no Call User Data Field in an ISO 8208/CCITT X.25 Call Request/Incoming Call packet) shall indicate ISO 8073/CCITT X.224.

When using ISO 8208, values other than one of those listed in table 3.2 are outside the scope of these agreements.

3.10 MIGRATION CONSIDERATIONS

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

3.10.1 X.25-1980

Until there is widespread availability of 1984 X.25 service, it will be necessary for X.400 systems to use those existing packet-switched public data networks which offer only pre-1984 X.25 service. While 1980 X.25 does not provide the CONS as defined by ISO 8348, there is no implication of non-conformance to these Agreements resulting there from for systems using 1980 X.25 to interchange data at the Network Layer, provided they conform in all other respects.

This is an exception to the Agreements for providing the OSI Network Service, granted temporarily for practical reasons. This exception will be removed when it is deemed to be no longer necessary, in the judgement of the Workshop. While this provision is in effect, it provides an alternative method of using 1980 X.25 to the provisions of 3.6.2.

3.11 USE OF PRIORITY

(Refer to the Ongoing Implementation Agreements document).

3.11.1 INTRODUCTION

(Refer to the Ongoing Implementation Agreements document).

3.11.2 OVERVIEW

(Refer to the Ongoing Implementation Agreements document).

3.12 CONFORMANCE

(Refer to the Ongoing Implementation Agreements document).

3.13 BIBLIOGRAPHY

1. CCITT Recommendation X.223 - 1988, Use of X.25 to Provide the OSI Connection-mode Network Service for CCITT Applications.

2. Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Operation with Packet-Switched Data Communications Networks, FIPS 100, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

3. Information Processing Systems - Data Communications - Protocol Combinations to Provide and Support the OSI Network Service - Part 1: General Principles, ISO/IEC 8880-1.

4. Information Processing Systems - Data Communications - Protocol Combinations to Provide and Support the OSI Network Service - Part 2: Provision and Support of the Connection-mode Network Service; ISO/IEC 8880-2.

5. Information Processing Systems - Data Communications - Protocol Combinations to Provide and Support the OSI Network Service - Part 3: Provision and Support of the Connectionless-mode Network Service; ISO/IEC 8880-3.

6. Information Technology - Telecommunications and Information Exchange Between Systems - OSI Routing Framework; ISO/IEC TR 9575.

4. TRANSPORT

Editor's Note: The term "Ongoing" used in this section refers to the Working Implementation Agreements document dated September 1990.

4.1 INTRODUCTION

These agreements support the integration of LANs, packet networks, and other WANs with the smallest possible set of mandatory protocol sets, in accordance with the other agreements already reached. Nothing here shall preclude vendors from implementing protocol suites in addition to the ones described in this document.

4.2 SCOPE AND FIELD OF APPLICATION

This chapter presents agreements for providing the OSI Transport layer services over both connection mode and connection-less mode services.

4.3 STATUS

Completed December 1989.

4.4 ERRATA

4.5 PROVISION OF CONNECTION MODE TRANSPORT SERVICE

Three connection mode protocol classes have been identified for implementation. Transport classes 0, 2 and 4 of X.224 (1988)[1] have been endorsed for use over CONS. Only Transport Class 4 of ISO 8073/Add. 2[2] has been endorsed for use over CLNS. The following class combinations are endorsed for CONS: (0), (0,2) or (0,2,4).

---

[1] Where a CR TPDU proposing Class 2 or 4 is initiated, Class 0 shall be explicitly indicated as an alternative class except if there is already one (or several) transport connection(s) assigned to the network connection (multiplexing being possible).

[2] In general, references to ISO 8073 in ISO 8073/Add. 2 should be interpreted as applying to X.224 (1988); however, the reference to Clause 14.6.a in Clause 14 of ISO 8073/Add. 2 should be interpreted as a reference to Clause 14.5.a of X.224(1988).

4.5.1 TRANSPORT CLASS 4

4.5.1.1 Transport Class 4 Overview

Transport Class 4 is mandatory for communication between systems using the OSI CLNS and may also be used for systems using the OSI CONS (e.g., a private MHS, etc.).

4.5.1.2 Protocol Agreements

A disconnect request shall be issued in response to a connect request when the maximum number of Transport connections is reached or exceeded.

4.5.1.2.1 General Rules

- All implementations shall request "use of extended formats" in the CR TPDU. Implementations shall accept the "use of extended formats" in the CC TPDU if it was proposed in the CR TPDU. Implementations shall accept "use of normal formats" if it was proposed in the CR TPDU.

- Negotiation of protection is outside the scope of these agreements. If negotiation of protection is not supported, receipt of the protection parameters in CR TPDU and CC TPDU shall be ignored.

- Implementations shall be capable of proposing and accepting the non-use of checksums.

- Use of the acknowledgment time parameter is optional. If an implementation is operating any policy which delays the transmission of AK TPDUs, the maximum amount of time by which a single AK TPDU may be delayed shall be indicated to the peer Transport service provider using the acknowledgment time parameter. The value transmitted should be expressed in units of milliseconds and rounded up to the nearest whole millisecond.

- QoS negotiation is outside the scope of these agreements. If QoS negotiation is not supported, receipt of the parameters "throughput," "residual

error rate," "priority," and "transit delay" in the CR and CC TPDUs shall be ignored.

- It is recommended that implementations not send user data in the CR TPDU or the CC TPDU. The disposition of any user data received in a CR TPDU or CC TPDU is implementation dependent.

- It is recommended that implementations not send user data in the **DR TPDU**. The disposition of any user data received in a **DR TPDU** is implementation dependent.

- An unknown parameter in any received CR TPDU shall be ignored.

- A Transport entity shall accept a DR TPDU and a corresponding DC TPDU with or without a checksum in response to a CR or CC TPDU.

- Transmitted DR TPDUs shall carry a disconnect reason code which pertains to the actual cause of the disconnect. A DR TPDU may carry a reason code of "0" (unspecified) if an appropriate reason code is not defined.

- Known parameters with valid lengths but with invalid values in a CR TPDU shall be handled as follows:

Parameter	Action
TSAP id	Send DR TPDU
TPDU size	ignore parameter, use default
Version	ignore parameter, use default
Checksum	discard CR TPDU
Alternate Protocol Classes	Protocol Error

- Unrecognized or not applicable bits of the Additional Options parameter shall be ignored.

4.5.1.2.2 TRANSPORT CLASS 4 SERVICE ACCESS POINTS OR SELECTORS

If present, the TSAP Id. field in the CR and CC TPDUs shall be encoded as a variable length field and will be interpreted as an octet string. The length of the string cannot exceed 32 octets.

4.5.1.2.3 Retransmission Timer

It is recommended that the value used for the retransmission timer be based upon the round-trip delay experienced on a transport connection. The implementation should maintain, and continually update, an estimate of the round-trip delay for the TC. From this estimate, a value for the retransmission timer is calculated each time it is started. An example technique for maintaining the estimate and calculating the retransmission timer is described below. The value of the retransmission timer may be calculated according to the following formula:

$$T1 \longleftarrow kE + AR$$

In this formula, E is the current estimate of the round-trip delay on the transport connection, AR is the value of the acknowledgement time parameter received from the remote transport service provider during connection establishment, and k is some locally administered factor.

A value for k should be chosen to keep the retransmission timer sufficiently small such that lost TPDUs will be detected quickly, but not so small that false alarms are generated causing unnecessary retransmission.

The value of E may be calculated using an exponentially weighted average based upon regular sampling of the interval between transmitting a TPDU and receiving the corresponding acknowledgment. Samples are taken by recording the time of day when a TPDU requiring acknowledgment is transmitted and calculating the difference between this and the time of day when the corresponding acknowledgment is received. New samples are incorporated with the existing average according to the following formula.

$$E \longleftarrow E + (1 - \alpha)(S - E)$$

In this formula, S is the new sample and α is a parameter which can be set to some value between 0 and 1. The value chosen for α determines the relative weighting placed upon the current estimate and the new sample. A large value of α weights the old estimate more heavily causing it to respond only slowly to variations in the round-trip delay.

A small value weights the new sample more heavily causing a quick response to variations. (Note that setting α to 1 will effectively disable the algorithm and result in a constant value for E, being that of the initial seed.)

If α is set to $1-2^{-n}$ for some value of n, the update can be reduced to a subtract and shift as shown below.

$$E \leftarrow\!\!- E + 2^{-n} (S - E)$$

When sampling, if an AK TPDU is received which acknowledges multiple DT TPDUs, only a single sample should be taken being the round-trip delay experienced by the most recently transmitted DT TPDU. This attempts to minimize in the sample any delay caused by the remote transport service provider withholding AK TPDUs.

4.5.1.2.4 Keep-Alive Function

The Class 4 protocol detects a failed Transport connection by use of an 'inactivity timer'. This timer is reset each time a TPDU is received on a connection. If the timer ever expires, the connection is terminated.

The Class 4 protocol maintains an idle connection by periodically transmitting an AK TPDU upon expiration of the 'window timer'. Thus, in a simple implementation, the interval of one transport entity's window timer must be less than that of its peer's inactivity timer, and vice versa. The following agreements permit communicating transport entities to maintain an idle connection without shared information about timer values.

- In accordance with ISO 8073/X.224, Clause 12.2.3.9.a, all implementations must respond to the receipt of a duplicate AK TPDU not containing FCC by transmitting an AK TPDU containing the 'flow control confirmation' parameter.

- Implementations must always transmit duplicate AK TPDUs without FCC on expiration of the local window timer (see ISO 8073/X.224, Clause 12.2.3.8.1). Receipt of this TPDU by the remote Transport entity will cause it to respond with an AK TPDU containing the 'flow control confirmation' parameter. When this is received by the local transport entity, it will reset its inactivity timer. See figure 4.1.

- It is a local matter for an implementation to set the intervals of its timers to appropriate relative values. Specifically:

- The window timer must be greater than the round-trip delay. See section 4.5.1.2.3.

- The inactivity timer must be greater than two times the window timer; and should normally be an even greater multiple if the Transport connection is to be resilient to the loss of an AK TPDU.

A duplicate AK TPDU (See fig. 4.1) is one which contains the same values for YR-TU-NR, credit, and subsequence number as the previous AK TPDU transmitted. A duplicate AK TPDU does not acknowledge any new data, nor does it change the credit window.

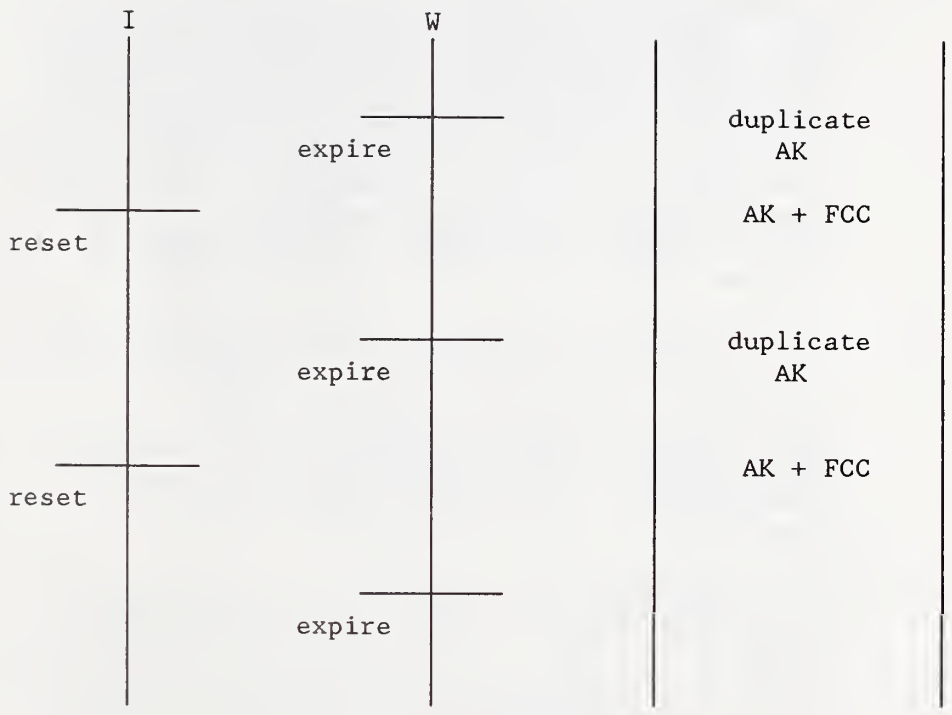


Figure 4.1. AK exchange on idleconnection.

4.5.1.2.5 Congestion Avoidance Policies

This section defines both mandatory and optional requirements relating to avoiding congestion in OSI networks and recovering from it when it is experienced. The mandatory requirements specify a minimum approach to congestion avoidance/recovery which can be tuned based upon the specific requirements of the network. The optional requirements specify a dynamic window sizing scheme which, if implemented, will contribute further to the avoidance of congestion in the network.

Mandatory Requirements

1 A maximum size for the "receive credit window," the value of which is locally configurable, should be provided. A "receive credit window" reflects the number of credits sent by a Transport entity for a Transport connection. The maximum size of the "receive credit window" shall be referred to as WR_1.

2 A maximum size for the "sending credit window," the value of which is locally configurable, shall be provided. A "sending credit window" reflects the number of data TPDUs that a Transport entity is willing to send on a Transport connection. The maximum size of the "sending credit window" shall be referred to as WS_1. As specified in ISO 8073, the "sending credit window" shall also be less than or equal to the remote "receive credit window" as conveyed in the last CDT field.

3 It is strongly recommended that an implementation use a retransmission timer per Transport connection. If, upon expiration of the retransmission timer, an implementation allows more than "1" TPDU to be transmitted a means to locally adjust the maximum number shall be provided.

4 All implementations shall have the capability of operating without delaying ACKs of data TDPUs received in-sequence (i.e., A_L essentially equals zero). If an implementation optionally chooses to explicitly delay ACKs, a means to locally adjust A_L shall be provided.

Optional Requirements

For systems implementing the dynamic window sizing scheme the following rules apply as described below.

RECEIVING TRANSPORT ENTITY (RTE) RULES:

Rule 1: Initialization of Window

The initial value of WR (known as WR_0) shall have a locally configurable upper bound. This window is sent to the sending transport entity (STE) in the next CDT field transmitted.

Rule 2: Required Sampling Period

All RTEs shall maintain a fixed value for WR until the next 2WR DT TPDU arrive since the last CDT field was transmitted by the RTE.

Rule 3: Required Counting of Received TPDUs in a Sampling Period

All RTEs shall maintain a count, N equal to the total number of TPDUs received and a count, NC equal to the total number of TPDUs received which had the CE Flag set. All types of TPDUs are included in the counts for N and NC, not just DT TPDUs.

Rule 4: Required Action upon the end of a Sampling Period

All RTEs shall take the following action at the end of each sampling period:

- If the count NC is less than 50 percent of the count N, the RTE shall increase WR by adding 1 up to a maximum, WR_1, (that is based on the local buffer management policy) otherwise, it shall decrease WR by multiplying by 0.875 (a minimum of 1).
- Reset N and NC to zero.
- Transmit the new window WR in the next CDT field sent to the sending transport entity.

SENDING TRANSPORT ENTITY (STE) RULES

Rule 1: Initialization of Window

All STEs shall maintain a sending window size (WS). Initially and also as long as there is no loss, WS is set equal to the receiving window value WR received from the remote RTE in the last CDT field.

Rule 2: Required Action on a Timeout

All STEs shall reset WS to one when the retransmissions timer expires and indicates a lost TPDU. WS now limits the number of DT TPDUs that may be transmitted or retransmitted without further acknowledgments.

Rule 3: Required Counting of Acknowledged TPDU

All STEs shall maintain a count, ACKRCVD of the number of DT TPDUs acknowledged, by the RTE, since WS was last adjusted. Therefore each time WS is adjusted, the count ACKRCVD shall be reset to zero.

Rule 4: Increase Window Policy

All STEs shall increase WS by one each time ACKRCVD is equal to or greater than the current value of WS, unless WS exceeds the window permitted by the remote RTE.

4.5.1.2.6 USE OF PRIORITY

(Refer to the Ongoing Implementation Agreements).

4.5.2 Transport Class 0

4.5.2.1 Transport Class 0 Overview

Transport Class 0 over X.25 is mandatory (see X.400) for use in communicating with public MHS systems operating in accordance with the CCITT X.400 series recommendations. The purpose of the agreements concerning Transport Class 0 is to allow connection to these public services. Transport Class 0 over X.25 can also be used in communicating between PRMDs (this choice is prevalent outside North America).

4.5.2.2 Protocol Agreements

Transport Class 0 agreements follow.

- The Error (ER) TPDU may be used at any time and upon receipt requires that the recipient disconnect the network connection, and by extension the transport connection.
- The allowed values for the maximum TPDU size are 128, 256, 512, 1024, and 2048.
- The Class 0 protocol does not support multiplexing. At any instant, one Transport corresponds to one Network connection.
- It is recommended that the optional timers TS1 and TS2, if implemented, be settable by local system management. Values in the order of minutes should be supported.
- An unlimited TSDU length must be supported.

4.5.2.2.1 Transport Class 0 Service Access Points

For communicating with public MHS systems, section 5 of X.410 specifies the use and format of TSAP identifiers.

4.5.2.3 Rules for Negotiation

The rules for class negotiation shall be used.

4.5.3 Transport Class 2

4.5.3.1 Transport Class 2 Overview

Transport Class 2 is applicable in OSI end systems which provide the Connection-mode Network Service.

4.5.3.2 Protocol Agreements

Transport Class 2 agreements follow:

- The values of the TS1 and TS2 timers shall be configurable. The recommended timer values are:

  TS1: 60 seconds
  TS2: 60 seconds

- If present, the TSAP-id field in the CR and CC TDPUs shall be encoded as a variable length field and will be interpreted as an octet string. The length of the string cannot exceed 32 octets

- The rules for class negotiation shall be used.

- QoS negotiation is outside the scope of these agreements. If QoS negotiation is not supported, receipt of the parameters "throughput," "residual error rate," "priority," and "transit delay" in the CR and CC TPDU shall be ignored.

  Note 1: If Class 0 is indicated in the Alternative Protocol Class field and QoS parameters are conveyed and the responding end system chooses Class 0, then the QoS parameters have been ignored by the responding system.

4.6 PROVISION OF CONNECTIONLESS TRANSPORT SERVICE

Document ISO 8072/Add. 2 is the Transport Service Definition covering Connectionless-mode Transmission. Document ISO 8602 is the Protocol for providing the Connectionless-Mode Transport Service.

4.6.1 Connectionless Transport Overview

When providing the connectionless Transport Service, the protocol shall be implemented as specified in ISO 8602.

4.6.2 Protocol Agreements

The connectionless Transport protocol is a relatively simple protocol providing little opportunity for conflicting interpretations. A few relevant agreements follow.

- The optional elements of procedure for use of CLTS over CONS (i.e., clause 6.3 of ISO 8602) will not be supported.
- A Unitdata TPDU that is received that contains a protocol error or an unknown destination TSAP ID shall be discarded.

4.6.2.1 Connectionless Transport Service Access Points or Selectors

The TSAP selector field in the UD TPDU shall be encoded as a variable length field and will be interpreted as an octet string. The length of the string cannot exceed 32 octets.

4.7 TRANSPORT PROTOCOL IDENTIFICATION

The absence of Call User Data (CUD) in an X.25/ISO 8208 Call Request/Incoming Call packet indicates the operation of ISO 8073/CCITT X.224.

Protocol Identification TPDU values applicable to these agreements are given in table 4.1. These TPDUs, when used, are conveyed as N-connect user data.

Table 4.1. Protocol Identification TPDU Values

TPDU Value	Protocol
03 01 01 00 *	ISO 8073/Add. 1
03 01 02 00 **	ISO 8602

Notes: * Corresponds to an ISO 8073/Add. 1 UN-TPDU and a X.224 Annex B PI-TPDU.

** Corresponds to an ISO 8073/Add. 1 UN-TPDU

The following agreements apply.

- Any additional TPDU, which follows (by concatenation) a Protocol Identification TPDU shall be ignored if ISO 8073/Add. 1 is not supported.

- When using ISO 8208, usage of a Protocol Identification TPDU not corresponding to those listed in table 4.1 is outside the scope of these agreements.

5. UPPER LAYERS

5.1 INTRODUCTION

In this portion of the Implementors' Agreements, the NIST Upper Layers SIG is primarily concerned with providing implementation agreements for ACSE, ROSE, RTSE, and the Presentation and Session layers, so that systems implemented according to these agreements can successfully interoperate.

5.1.1 References

All documents referenced in the Upper Layers section of these agreements can be found in the REFERENCES section of this NIST Implementation Agreements document.

5.2 SCOPE AND FIELD OF APPLICATION

The agreements in this section apply to all ASE agreements in this document. Each ASE SIG chooses which protocols, functional units, application contexts, and parameters it requires. These must be listed in the "Specific ASE Requirements" section of this chapter.

5.3 STATUS

This text is stable as of December, 1989.

5.4 ERRATA

5.4.1 ISO Defect Solutions

This section lists the defect solutions from ISO which are currently recognized to be valid for the purposes of NIST conformance:

ISO 8326 defect solutions:

023, 024

ISO 8327 defect solutions:

037, 038

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

The following approved defect solutions have been integrated into the current revisions of ISO 8326 and ISO 8327, but are not part of CCITT X.215 and X.225 (1984). The defect solutions must be incorporated into CCITT Session to insure conformance with ISO Session.

ISO 8326 defect solutions:

004, 006, 007, 009, 011, 012, 013, 014, 015, 016, 017, 020.

ISO 8327 defect solutions:

001, 003, 004, 005, 006, 007, 008, 009, 010, 012, 017, 018, 019, 026, 027, 030, 034, 035.

5.4.3 Errata approved at March, 1990 meeting

Errata to this chapter are marked with change bars; deleted text is left but with strikeouts. The following table indicates the section and type for each erratum.

SECTION	TYPE	COMMENT
5.4.3	editorial	added errata summary table
5.5.2	editorial	grammar
5.8.3.7	editorial	spelling
5.8.5.1	editorial	typographical
5.8.5.1	editorial	extraneous word
5.8.7	editorial	update to document reference
5.9.4	editorial	update to document reference
5.12.1	editorial	moved some abstract syntaxes so that all abstract syntaxes are listed under Presentation requirements; deleted some Associated transfer syntax entries so that there is only one reference to Associated transfer syntax per group of abstract syntaxes
5.12.1.1.1	technical	changes to the encoding of two FTAM abstract syntaxes: "NBS abstract syntax AS1" and "NBS file directory entry abstract syntax"
5.14.2	editorial	correction to abstract syntax definition
5.14.2	editorial	added explanatory note on nil application context

5.5 ASSOCIATION CONTROL SERVICE ELEMENT

5.5.1 Introduction

This section details the implementation requirements for the Association Control Service Element (ACSE) of the Application layer as defined in ISO 8649 and ISO 8650.

5.5.2 Services

All ACSE services are within the possible scope of a NIST-conformant system.

5.5.3 Protocol Agreements

5.5.3.1 Application Context

Values for and uses of Application Context names are determined by specific ASEs. Values used by NIST ASE SIGS are listed in the section entitled "Specific ASE Requirements."

5.5.3.2 AE Title

AE-titles shall be implemented as specified in Amendment 1 to ISO 8650 (ISO 8650/AM1).

5.5.4 ASN.1 Encoding Rules

When the ABRT APDU is used during the connection establishment phase, Presentation layer negotiation is considered to be complete, and the "direct-reference" component of EXTERNAL shall not be present.

5.5.5 Connectionless

The connectionless ACSE protocol shall be implemented as specified in ISO DIS 10035.

No further agreements beyond those specified elsewhere in this chapter have been made regarding this standard.

5.5.6 Result Parameter

Refer to Working Implementation Agreements, ~~March 1990~~ September 1990.

5.6 ROSE

ROSE shall be implemented as specified in ISO DIS 9072-1.2 and ISO DIS 9072-2.2.

No further agreements beyond those specified elsewhere in this chapter have been made regarding this standard.

5.7 RTSE

RTSE shall be implemented as specified in ISO 9066-1 and ISO 9066-2.

No further agreements beyond those specified elsewhere in this chapter have been made regarding this standard.

5.8 PRESENTATION

5.8.1 Introduction

This section details the implementation requirements for the Presentation layer as defined in the Presentation Service Definition, ISO 8822, and the Presentation Protocol Definition, ISO 8823.

The task of the Presentation layer is to carry out the negotiation of transfer syntaxes and to provide for the transformation to and from transfer syntaxes. The transformation to and from a particular transfer syntax is a local implementation issue and is not discussed within this section. This section is concerned with the protocol agreements, and thus is entirely devoted to the issues involved with the negotiation of transfer syntaxes and the responsibilities of the Presentation protocol.

5.8.2 Service

Only the Kernel functional unit need be supported. The Context Management and Context Restoration functional units are outside the scope of these agreements.

The requirement that the Presentation kernel functional unit be implemented does not imply that any of the Session functional units for expedited data, typed data, and capability data and the corresponding Presentation service primitives are required to be implemented.

5.8.3 Protocol Agreements

5.8.3.1 Transfer Syntaxes

- The following transfer syntax must be supported for all mandatory abstract syntaxes: the basic encoding rules for ASN.1. This syntax is derived by applying the basic encoding rules for ASN.1 to the abstract syntax (see the Basic Encoding Rules for ASN.1, ISO 8825).

- The number of transfer syntaxes proposed is dependent upon the recognized transfer syntaxes which are available to support the particular abstract syntaxes used by an Application Entity.

5.8.3.2 Presentation Context Identifier

- A conformant implementation shall encode Presentation context identifiers in the range 0 to 32,767.

- Implementations must be able to handle a minimum of two Presentation contexts per connection.

5.8.3.3 Default Context

If the Presentation expedited data service is required, the default context must be explicitly present in the P-CONNECT PPDU at Presentation connect time.

5.8.3.4 P-Selectors

Local P-selectors shall be a maximum of four octets. This applies only to P-selectors in PPDUs whose receipt by an NIST-conformant system normally results in either a P-CONNECT indication or a P-CONNECT confirmation being issued.

5.8.3.5 Provider Abort Parameters

No conformance requirements are implied by the use of either the Abort-reason or the Event-identifier component of the ARP-PPDU. The decision to include these parameters is left up to the implementation issuing the abort.

5.8.3.6 Provider Aborts and Session Version

The Presentation Provider Abort PPDU (ARP-PPDU) shall be present regardless of the Session version in effect for a given association. This precludes the use of indefinite length encoding of an ARP-PPDU when Session Version 1 is in effect.

5.8.3.7 CPC-Type

Implementations shall not use any CPC-type values in the SS-user data parameter of the S-CONNECT unless more than one transfer syntax is proposed for a single Presentation context of the Presentation data values. Each CPC-type represents a unique transfer syntax, so if more than one transfer syntax is proposed, CPC-type values may appear in that SS-user-data parameter.

For a Presentation context for which the Basic Encoding Rules are a proposed transfer syntax, all PDVs in the user data parameter of the CP PPDU must be encoded first using the Basic Encoding Rules and must be examined by the receiving Presentation protocol machine. Following CPC-type values may be examined or ignored at the receiver's option (see ISO 8823, sec. 6.2.5.3).

5.8.3.8 Presentation-context-definition-result-list

No semantics are implied by the absence of the optional Presentation-context-definition-result-list component of the CPR-PPDU. This component is required if the Provider-reason is absent in the CPR-PPDU. If the Provider-reason is present, then the Presentation-context-definition-result-list is optional.

5.8.3.9 RS-PPDU

The Presentation-context-identifier-list shall not be present when only the kernel functional unit is in effect.

5.8.4 Presentation ASN.1 Encoding Rules

5.8.4.1 Invalid Encoding

If a received PPDU contains any improperly encoded data values (including data values embedded within the User Data field of a PPDU) and an abort is issued, then either an ARU or an ARP shall be issued.

5.8.5 General

5.8.5.1 Presentation Data Value (PDV)

- A Presentation data value (PDV) is a value of a type in an abstract syntax, e.g., a value of an ASN.1 type.
- A PDV may contain embedded PDVs in different contexts. A change of context within a PDV is indicated by an EXTERNAL. EXTERNAL implies an embedded PDV.
- A PDV cannot be split across PDV-lists in fully-encoded user data.
- Fully-encoded-data that is a series of PDVs in the same Presentation context (e.g., grouped FTAM PDUs) shall be encoded either as a single PDV-list (using the octet-aligned choice) or as a series of PDV-lists, each encoding either a single PDV (using the single-ASN1-type choice) or multiple PDVs (using the octet-aligned choice). Note that receivers must accept any of the above encodings.

5.8.6 Connection Oriented

The Transfer-syntax-name component of a PDV-list value shall be present in a CP PPDU if and only if more than one transfer syntax name was proposed for the Presentation context of the Presentation data values. The Transfer-syntax-name component of a PDV-list value shall always be present in a CPC-type. If only the Kernel functional unit is negotiated, then the

Transfer-syntax-name component of a PDV-list value shall only appear in the CP PPDU and CPC-type.

5.8.7 Connectionless

The connectionless Presentation protocol shall be implemented as specified in ISO DIS 9576.

The Transfer-syntax-name component of a PDV-list value shall be present in a UD PPDU if and only if more than one transfer syntax name was proposed for the Presentation context of the Presentation data values. The Transfer-syntax-name component of a PDV-list value shall always be present in a UDC-type. The Transfer-syntax-name component of a PDV-list value shall only appear in the UD PPDU and UDC-type.

No further agreements beyond those specified elsewhere in this chapter have been made regarding this standard.

5.9 SESSION

5.9.1 Introduction

This section details the implementation requirements for the Session layer as defined in the Session Service Definition, ISO 8326 and the Session Protocol Definition, ISO 8327.

5.9.2 Services

The following functional units are within the scope of a NIST conformant system.

Kernel

Duplex

Expedited Data

Resynchronize

Exceptions

Activity Management

Half-duplex

Minor Synchronize

Major Synchronize

Typed Data

5.9.3 Protocol Agreements

5.9.3.1 Concatenation

When a category 0 SPDU is concatenated with a category 2 SPDU, the category 0 SPDU shall not contain User Data.

Extended concatenation is not required and can be refused using the normal negotiation mechanisms of the Session protocol.

5.9.3.2 Segmenting

Session segmenting is not required and can be refused using the normal negotiation mechanisms of the Session protocol. All conformant implementations shall be able to interwork without Session segmenting.

5.9.3.3 Reuse of Transport Connection

Reuse of a Transport connection is not required and can be refused.

5.9.3.4 Use of Transport Expedited Data

The Session use of Transport expedited service is supported. The meaning of "supported" is specified in section 3.1 of ISO DISP AFTnn-1.

Note: It is recommended to request and use this feature.

5.9.3.5 Use of Session Version Number

Session Versions 1 and 2 are recognized. Each relevant NIST SIG chooses the version or versions of Session which it requires for a particular implementation phase, and these choices are documented in section 5.12.

Session Version 2 specifies the use of unlimited user data during connection establishment as dictated by the AD 2 to ISO 8327 to Incorporate Unlimited User Data.

All Session Version 1 implementations must be able to negotiate Version 1 operation when responding to a CONNECT (CN) SPDU proposing both Version 1 and Version 2.

In addition, all Session Version 1 implementations, upon receipt of a CONNECT (CN) SPDU proposing only Version 2, should respond with a REFUSE (RF) SPDU containing a Reason Code indicating that the proposed version is not supported. Until pending defect reports are adopted, implementations may disconnect.

If Session Versions 1 and 2 are both proposed in the CONNECT (CN) SPDU, then the maximum length of the User Data parameter value in the CONNECT (CN) SPDU shall be 512 octets and a PGI field of 193 shall be associated with this parameter. This implies that an implementation supporting both Session Versions 1 and 2 can establish a connection with an implementation supporting only Version 1.

If only Session Version 2 is proposed in the CONNECT (CN) SPDU, then the maximum length of the Session User Data parameter value of the S-CONNECT service request shall be 10,240 octets. This restriction implies that the OVERFLOW ACCEPT (OA) SPDU and CONNECT DATA OVERFLOW (CDO) SPDU are not used. If the length of the User Data parameter value is no greater than 512 octets, then an associated PGI field of 193 shall be used, otherwise a PGI field of 194 shall be used.

When Session Version 2 is negotiated, then in all SPDUs the maximum length of the User Data parameter value with an associated PGI field of 193 shall be 10,240 octets. NIST-conformant Session Version 2 implementations need only support the maximum data lengths specified in the Specific ASE Requirements section.

5.9.3.6 Receipt of Invalid SPDUs

Upon receipt of an invalid SPDU, the SPM shall take any action in A.4.3 of the Session Protocol Definition ISO/IS 8327 except Action d.

5.9.3.7 Invalid SPM Intersections

If the conditions described in A.4.1.2 of the Session Protocol Definition ISO/IS 8327 are satisfied, the SPM shall always take the actions described by A.4.1.2 a.

Note: This implies that no S-P-EXCEPTION-REPORT indications will be generated nor EXCEPTION REPORT SPDUs sent due to invalid intersections of the Session state table resulting from received SPDUs.

5.9.3.8 S-Selectors

S-selectors shall be a maximum of 16 octets.

5.9.4 Connectionless

The connectionless Session protocol shall be implemented as specified in ISO IS 9548.

No further agreements beyond those specified elsewhere in this chapter have been made regarding this standard.

5.10 UNIVERSAL ASN.1 ENCODING RULES

5.10.1 TAGS

The maximum value of an ASN.1 basic encoding tag that need be handled by an NIST-conformant implementation shall be 16,383. This is the maximum unsigned number that can be represented in 14 bits, therefore, the maximum encoding of a tag occupies 3 octets.

5.10.2 Definite Length

The maximum value of an ASN.1 length octets component that need be handled by an NIST-conformant implementation shall be 4,294,967,295. This is the maximum unsigned integer that can be represented in 32 bits, therefore, the maximum encoding of a length octets component will occupy 5 octets. Also, note this restriction does not apply to indefinite length encoding.

5.10.3 EXTERNAL

It is assumed that "Presentation layer negotiation of encoding rules" is always in effect, and therefore clause 32.5 of the Specification of ASN.1, ISO 8824 never applies.

a. If a data value to be encapsulated in an EXTERNAL type is an instance of a single ASN.1 type encoded according to the Basic Encoding Rules for ASN.1, then the option "single-ASN.1-type" shall be chosen as its encoding.

b. If a data value to be encapsulated in an EXTERNAL type is encoded as an integral number of octets, and case a. does not apply, then the option "octet-aligned" shall be chosen as its encoding.

5.10.4 Integer

- Any incidence of an ASN.1 INTEGER type defined in an abstract syntax describing protocol control information must be encoded so that the length of its contents octets is no more than four octets, unless an explicit NIST agreement to the contrary is made for a specific INTEGER type.

5.10.5 String Types

- The contents octets for a constructed encoding of a BIT STRING, OCTET STRING, or character string value consists of the complete encoding of zero, one, or more data values, and the encoding of these data values must be primitive.

5.10.6 Bit String

- Unless otherwise specified in the abstract syntax definition, each bit named in a BIT STRING type used in that abstract syntax definition shall be explicitly encoded in the associated BIT STRING value, even if it is part of a string of trailing zero bits.

  Extra trailing bits beyond the exact number of bits which correspond to the complete list of the named bits specified shall never be encoded. This rule applies to all BIT STRING types unless stated otherwise in the standards.

5.11 CHARACTER SETS

See aligned section of Working Implementation Agreements dated December 1989.

5.12 CONFORMANCE

In order for an implementation to be in conformance with the NIST implementors' agreements, the rules below shall be followed.

- A conformant implementation must meet all of the requirements of this specification. All documents referenced in the Upper Layers section shall be used as the supporting documents for all implementations of ACSE, ROSE, RTSE, Presentation, or Session. The full references for these documents are in the REFERENCES section.

- NIST-conformant implementations shall be ISO conformant. PICS may contain limitations on length or value aspects of a protocol. PICS of NIST-conformant systems shall not contain restrictions more severe than those in these implementation agreements.

  Note: An implementation may abort a connection if the constraints specified in these agreements are violated.

5.12.1 Specific ASE Requirements

The following list for each ASE the corresponding NIST SIG's requirements of and restrictions on ACSE, ROSE, RTSE, Presentation, and Session.

All listed requirements and restrictions shall be included in an NIST-conformant system and shall be implemented in accordance with these NIST Implementor's agreements.

5.12.1.1 FTAM

5.12.1.1.1 Phase 2

ACSE Requirements:
all

Application Contexts:
- "ISO FTAM"

{ iso(1) standard(0) 8571
application-context iso-ftam(1) }
- implies the use of the ACSE and the FTAM ASE.

A value is defined for the AE Title only to satisfy the FTAM requirement for exchanging fields of this type. This value does not identify an Application Entity and carries no semantics.

If the AE title is used, AE-title-form2 shall be supported. Support of AE-title-form2 includes support of AP-title-form2 and AE-qualifier-form2.

The value for the AP title is { 1 3 9999 1 ftam-nil-ap-title (7) } at this time. Values for the AE qualifier are outside the scope of these agreements.

The use of AP invocation identifiers and AE invocation identifiers by FTAM is outside the scope of these agreements.

Presentation Requirements:

Presentation Functional Units:

- o kernel

Presentation Contexts:

- o At least 3 Presentation Contexts must be supported.

Abstract Syntaxes:

Abstract Syntaxes for conformant Implementations

- "ISO 8650-ACSE1"

{joint-iso-ccitt(2)
association-control(2)
abstract-syntax(1) apdus(0) version1(1)
}

o "FTAM-PCI"
{ iso(1) standard(0) 8571
abstract-syntax(2) ftam-pci(1) }

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

o "FTAM unstructured binary abstract
syntax"
{ iso(1) standard(0) 8571
abstract-syntax(2)
unstructured-binary(4) }

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

Editor's Note: In Definitions below, "NBS" designation will be preserved.

Abstract Syntaxes Depending on Implementation Profile

o "FTAM-FADU"
{ iso(1) standard(0) abstract-syntax(2)
ftam-fadu(2) }

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

o "FTAM unstructured text abstract syntax"
{ iso(1) standard(0) 8571
abstract-syntax(2) unstructured-text(3)
}

Associated-Transfer-Syntax:

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

- "NBS abstract syntax AS1"
  { iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) }

Associated-Transfer-Syntax:

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

- "NBS file directory entry abstract syntax"
  { iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2) }

Associated Transfer Syntax:

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

Editor's Note: The changes above involving "OIW(14)" were not explicitly mentioned at the March 1990 Plenary, but were implied from a correspondingly approved FTAM motion.

Session Requirements:

Session Functional Units:

- kernel
- duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options:

Session Functional Units:

- resynchronize
  only a Resynchronize Type value of "abandon"

- o minor synchronize

  Note: The minor synchronize functional unit is required whenever the resynchronize functional unit is available.

The default value for Minor Sync Point Sync type item shall always be used, i.e., explicit confirmation is required.

ASN.1 Encoding Requirements

Some INTEGER types of the FTAM PCI may exceed the maximum size specified in the UNIVERSAL ASN.1 ENCODING Rules. See the Range of values for INTEGER type Parameters of the FTAM chapter.

5.12.1.2 MHS

5.12.1.2.1 Phase 1 (1984 X.400)

Session Requirements:

Session Functional Units:

- o kernel
- o half-duplex
- o exceptions
- o activity management
- o minor synchronize

Version Number: 1

Maximum size of User Data parameter field: 512

Session Notes:

- o Restricted use is made by the RTS of the Session services implied by the functional units selected. Specifically,
  - . No use is made of S-TOKEN-GIVE, and
  - . S-PLEASE-TOKENS only asks for the data token.

- In the S-CONNECT SPDU, the Initial Serial Number should not be present.

- The format of the Connection Identifier in the S-CONNECT SPDU is described in Version 5 of the X.400-Series Implementors' Guide.

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

ROSE Requirements:

ROSE is not used.

RTSE Requirements:

- Monologue
- TWA - optional
- checkpointing
  .minimum checkpointsize = 1
  .minimum windowsize = 1
- no checkpointing

Notes:

- Monologue Association:
  - initiator keeps initial turn
  - APDUs are transferred from initiator to responder only
  - no turn passing
  - only the initiator effects the orderly release of an association

- Two way alternate Association
  - the initiator may keep or pass the initial turn, at binding
  - APDUs are transferred by the holder of the turn
  - only the initiator effects the orderly release of an association, when it possesses the turn

ACSE Requirements:

As per Phase 2, Protocol P7.

Application Contexts:

- "MTS-transfer-protocol-1984" - mandatory
- "MTS-transfer-protocol" - mandatory
- "MTS-transfer" - mandatory

Presentation Requirements:

Presentation Functional Units:
- kernel

Presentation Contexts:
- at least 3 must be supported

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

- "MTS-RTSE"

- "MTSE"

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

Session Requirements:
As per Phase 2, Protocol P7.

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

ROSE Requirements:
Operation and association classes are used as per the standard.

RTSE Requirements:
- TWA
- normal-mode
- checkpointing
  .minimum checkpointsize = 1
  .minimum windowsize = 1
- no checkpointing

Notes:
- Monologue Association:
  - initiator keeps initial turn
  - APDUs are transferred from initiator to responder only
  - no turn passing
  - only the initiator effects the orderly release of an association

- Two way alternate Association
  - the initiator may keep or pass the initial turn, at binding
  - APDUs are transferred by the holder of the turn
  - only the initiator effects the orderly release of an association, when it possesses the turn

ACSE Requirements:

all

The use of AP-TITLE, AE-QUALIFIER, AP-INVOCATION-ID, and AE-INVOCATION-ID is not recommended; however, a receiving entity must be capable of ignoring them (if present) without refusing the connection.

Application Contexts:

- "MS-access" - mandatory; normal mode
- "MS-reliable-access" - optional; normal mode

Abstract-Syntaxes:

- "ISO-8650-AGSE1"
  {-joint-iso-ccitt(2)
  association-control(2)
  abstract-syntax(1)-apdus(0)-version1(1)
  }

Associated-Transfer-Syntax:

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

Presentation Requirements:

Presentation Functional Units:

- kernel

Presentation Contexts:

- at least 5

Abstract Syntaxes:

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

- MSBind/MSUnbind (with or without RTSE)
- MSSE (Message Submission)
- MASE (Message Administration)
- MRSE (Message Retrieval)

Associated Transfer Syntax:

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

Session Requirements:

Session Functional Units:

- kernel
- half-duplex
- exceptions
- activity management
- minor synchronize

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Notes:

- MHS proposes both versions 1 and 2 for pass through mode (X.410 mode), but only version 2 for normal mode.
- Restricted use is made by the RTS of the Session services implied by the functional units selected. Specifically,
  - No use is made of S-TOKEN-GIVE, and
  - S-PLEASE-TOKENS only asks for the data token.
- In the S-CONNECT SPDU, the Initial Serial Number should not be present.
- The format of the Connection Identifier in the S-CONNECT SPDU is described in Version 5 of the X.400-Series Implementors' Guide.

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

ROSE Requirements:
As per Phase 2, P7.

RTSE Requirements:
As per Phase 2, P7.

ACSE Requirements:
As per Phase 2, P7.

Application Contexts:
- o "MTS-access" - mandatory
- o "MTS-reliable-access" - optional
- o "MTS-forced-access" - mandatory
- o "MTS-forced-reliable-access" - optional

Presentation Requirements:
As per Phase 2, P7.

Session Requirements:
As per Phase 2, P7.

5.12.1.3 DS

5.12.1.3.1 Phase 1

ACSE Requirements:
all

Application Contexts:
- o "id-ac-directoryAccessAC"
  (joint-iso-ccitt(2) ds(5) 3 1)
- o "id-ac-directorySystemAC"
  (joint-iso-ccitt(2) ds(5) 3 2)

Abstract-Syntaxes:
- o "ISO-8650-AGSE1"
  {-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)-}

- "id-as-directoryAccessAS"
  joint-iso-ccitt(2)-ds(5)-9-1-}

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

- "id-as-directorySystemAS"
  {-joint-iso-ccitt(2)-ds(5)-9-2-}

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

Presentation Requirements:

Presentation Functional Units:
- kernel

Presentation Contexts:
- At least 2 Presentation Contexts must be supported.

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

- "id-as-directoryAccessAS"
  joint-iso-ccitt(2) ds(5) 9 1 }

- "id-as-directorySystemAS"
  { joint-iso-ccitt(2) ds(5) 9 2 }

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

Session Requirements:

Session Functional Units:
- o kernel
- o duplex

Version Number: 2

Maximum size of User Data parameter field:
10,240

5.12.1.4 Virtual Terminal

5.12.1.4.1 Phase 1a

ACSE Requirements:
all

Application Contexts:
- o "ISO VT"
  { iso(1) standard(0) 9041
  application-context(1) }
  - implies the use of the ACSE and the VT ASE

Abstract-Syntaxes:
- o "ISO-8650-AGSE1"
  {-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)-}

Presentation Requirements:

Presentation Functional Units:
- o kernel

Presentation Contexts:
- o at least 2 must be supported

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

- o "VT Basic"
  { iso(1) standard(0) 9041 abstract-syntax(2)
  }

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

Session Requirements:

Session Functional Units:
- o kernel
- o duplex
- o expedited data
- o major synchronize
- o resynchronize
  - only a Resynchronize Type value of "restart"
- o typed data

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options:
- o expedited data

5.12.1.4.2 Phase 1b

ACSE Requirements:
all

Application Contexts:
- o "ISO VT" { iso(1) standard(0) 9041 application-context(1) }
  - implies the use of the ACSE and the VT ASE

Presentation Requirements:

Presentation Functional Units:
- o kernel

Presentation Contexts:
- o at least 2 must be supported

Abstract Syntaxes:
- o "ISO 8650-ACSE1" { joint-iso-ccitt(2) association-control(2)

abstract-syntax(1) apdus(0) version1(1)
}

o "VT Basic"
{ iso(1) standard(0) 9041
abstract-syntax(2) }

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

Session Requirements:

Session Functional Units:
- o kernel
- o duplex
- o half-duplex
- o expedited data
- o major synchronize
- o resynchronize
  - only a Resynchronize Type value of "restart"
- o typed data

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options:
- o expedited data

5.12.1.5 MMS

See Working Implementation Agreements Document, ~~March 1990~~ September 1990.

5.12.1.6 Transaction Processing

See Working Implementation Agreements Document, ~~March 1990~~ September 1990.

5.13 APPENDIX A: RECOMMENDED PRACTICES

The optional "Reflect Parameter Values" parameter in the Provider ABORT SPDU shall be encoded so as to represent the Session connection state, the incoming event and the first invalid SPDU field exactly at the moment a protocol error was detected.

The first octet encodes the Session state as a number relative to 0 as detailed in table 5.1.

The second octet encodes the incoming event as a number relative to 0 as detailed in table 5.2.

The third octet contains the SI, PGI, or PI Code of any SI field, PGI unit or PI unit in error.

Note: The remaining 6 octets are undefined herein.

Table 5.1. Session States

State	Rel.#	Description
1	0	Idle, no transport connection
1B	1	Wait for T-connect confirm
1C	2	Idle, transport connected
2A	3	Wait for the ACCEPT SPDU
3	4	Wait for the DISCONNECT SPDU
8	5	Wait for the S-CONNECT response
9	6	Wait for the S-RELEASE response
16	7	Wait for the T-DISCONNECT indication
713	8	Data Transfer state
1A	9	Wait for the ABORT ACCEPT SPDU
4A	10	Wait for the MAJOR SYNC ACK SPDU or PREPARE SPDU
4B	11	Wait for the ACTIVITY END ACK SPDU or PREPARE SPDU
5A	12	Wait for the RESYNCHRONIZE ACK SPDU or PREPARE SPDU
5B	13	Wait for the ACTIVITY INTERRUPT SPDU or PREPARE SPDU
5C	14	Wait for the ACTIVITY DISCARD ACK SPDU or PREPARE SPDU
6	15	Wait for the RESYNCHRONIZE SPDU or PREPARE SPDU
10A	16	Wait for the S-SYNC-MAJOR response
10B	17	Wait for the S-ACTIVITY-END response
11A	18	Wait for the S-RESYNCHRONIZE response
11B	19	Wait for the S-ACTIVITY-INTERRUPT response
11C	20	Wait for the S-ACTIVITY-DISCARD response
15A	21	After PREPARE, wait for the MAJOR SYNC ACK SPDU or the ACTIVITY END ACK
15B	22	After PREPARE, wait for the RESYNCHRONIZE SPDU or the ACTIVITY DISCARD SPDU
15C	23	After PREPARE, wait for the RESYNCHRONIZE ACK SPDU, or the ACTIVITY INTERRUPT ACK SPDU or the ACTIVITY DISCARD ACK SPDU
18	24	Wait for GIVE TOKENS ACK SPDU
19	25	Wait for a recovery request or SPDU
20	26	Wait for a recovery SPDU or request
21	27	Wait for the CAPABILITY DATA ACK SPDU
22	28	Wait for the S-CAPABILITY-DATA response
1D	29	Wait for the CONNECT DATA OVERFLOW SPDU
2B	30	Wait for the OVERFLOW ACCEPT SPDU
15D	31	After PREPARE, wait for the ABORT SPDU

Table 5.2. Incoming Events

Event	Rel.#	Description
SCONreq	0	S-CONNECT request
SCONrsp+	1	S-CONNECT accept response
SCONrsp-	2	S-CONNECT reject response
SDTreq	3	S-DATA request
SRELreq	4	S-RELEASE request
SRELrsp+	5	S-RELEASE accept response
SUABreq	6	S-U-ABORT request
TCONcnf	7	T-CONNECT confirmation
TCONind	8	T-CONNECT indication
TDISind	9	T-DISCONNECT indication
TIM	10	Time out
AA	11	ABORT ACCEPT
AB-nr	12	ABORT - no reuse
AC	13	ACCEPT
CN	14	CONNECT
DN	15	DISCONNECT
DT	16	DATA TRANSFER
FN-nr	17	FINISH - no reuse
RF-nr	18	REFUSE - no reuse
SACTDreq	19	S-ACTIVITY-DISCARD request
SACTDrsp	20	S-ACTIVITY-DISCARD response
SACTEreq	21	S-ACTIVITY-END request
SACTErsp	22	S-ACTIVITY-END response
SACTIreq	23	S-ACTIVITY-INTERRUPT request
SACTIrsp	24	S-ACTIVITY-INTERRUPT response
SACTRreq	25	S-ACTIVITY-RESUME request
SACTSreq	26	S-ACTIVITY-START request
SCDreq	27	S-CAPABILITY-DATA request
SCDrsp	28	S-CAPABILITY-DATA response
SCGreq	29	S-CONTROL-GIVE request
SEXreq	30	S-EXPEDITED-DATA request
SGTreq	31	S-TOKEN-GIVE request
SPTreq	32	S-TOKEN-PLEASE request
SRELrsp-	33	S-RELEASE response reject
SRSYNreq(a)	34	S-RESYNCHRONIZE request abandon
SRSYNreq(r)	35	S-RESYNCHRONIZE request restart
SRSYNreq(s)	36	S-RESYNCHRONIZE request set
SRSYNrsp	37	S-RESYNCHRONIZE response
SSYNMreq	38	S-SYNC-MAJOR request
SSYNMrsp	39	S-SYNC-MAJOR response
SSYNmreq	40	S-SYNC-MINOR request
SSYNmrsp	41	S-SYNC-MINOR response
STDreq	42	S-TYPED-DATA request
SUERreq	43	S-U-EXCEPTION-REPORT request

Table 5.2. Incoming Events Continued

Event	Rel.#	Description
AB-r	44	ABORT - reuse SPDU
AD	45	ACTIVITY DISCARD SPDU
ADA	46	ACTIVITY DISCARD ACK SPDU
AE	47	ACTIVITY END SPDU
AEA	48	ACTIVITY END ACK SPDU
AI	49	ACTIVITY INTERRUPT SPDU
AIA	50	ACTIVITY INTERRUPT ACK SPDU
AR	51	ACTIVITY RESUME SPDU
AS	52	ACTIVITY START SPDU
CD	53	CAPABILITY DATA SPDU
CDA	54	CAPABILITY DATA ACK SPDU
ED	55	EXCEPTION DATA SPDU
ER	56	EXCEPTION REPORT SPDU
EX	57	EXPEDITED DATA SPDU
FN-r	58	FINISH - reuse SPDU
GT	59	GIVE TOKENS SPDU
GTA	60	GIVE TOKENS ACK SPDU
GTC	61	GIVE TOKENS CONFIRM SPDU
MAA	62	MAJOR SYNC ACK SPDU
MAP	63	MAJOR SYNC POINT SPDU
MIA	64	MAJOR SYNC ACK SPDU
MIP	65	MINOR SYNC POINT SPDU
NF	66	NOT FINISHED SPDU
PR-MAA	67	PREPARE (MAJOR SYNC ACK) SPDU
PR-RA	68	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	69	PREPARE (RESYNCHRONIZE) SPDU
PT	70	PLEASE TOKENS SPDU with Token Item Parameter
RA	71	RESYNCHRONIZE ACK SPDU
RF-r	72	REFUSE - reuse SPDU
RS-a	73	RESYNCHRONIZE - abandon SPDU
RS-r	74	RESYNCHRONIZE - restart SPDU
RS-s	75	RESYNCHRONIZE - set SPDU
TD	76	TYPED DATA SPDU
CDO	77	CONNECT DATA OVERFLOW SPDU
OA	78	OVERFLOW ACCEPT SPDU
PR-AB	79	PREPARE (ABORT) SPDU

5.14 APPENDIX B: OBJECT IDENTIFIER REGISTER

5.14.1 Register Index

Each entry in the index contains an object identifier value and a reference to the section describing the object identifier's use.

1. { iso(1) identified-organization(3) oiw(14) ulsig(8) application-context(1) nil(1) } is defined in section 5.14.2.

2. { iso(1) identified-organization(3) oiw(14) ulsig(8) abstract-syntax(2) octet-string(1) } is defined in section 5.14.2.

5.14.2 Object Identifier Descriptions

1. { iso(1) identified-organization(3) oiw(14) ulsig(8) application-context(1) nil(1) }

   This application context may be used by applications having a prior agreement regarding the application context.

   Note: This value is intended to be used by private applications that have an a priori agreement concerning the set of ASEs, related options, and any other information necessary for the interworking of AEs on an application association. This value does not identify any specific application context and cannot be used to identify the intended communications environment for the application association. Therefore, it is strongly recommended that private applications define and register an object identifier for their application context.

2. { iso(1) identified-organization(3) oiw(14) ulsig(8) abstract-syntax(2) octet-string(1) }

```
NIST-OIW-ULSIG-AS-::=BEGIN
octet-string---single-octet-string::=OCTET-STRING-
DEFINITIONS          END

NIST-OIW-ULSIG-AS-octet-string
DEFINITIONS ::= BEGIN
   Single-octet-string ::= OCTET STRING
END
```

This abstract syntax may be used by applications having a prior agreement regarding the content of the octet string.

6. REGISTRATION AUTHORITY PROCEDURES FOR THE OSI IMPLEMENTORS WORKSHOP (OIW)

Editor's Note: Previous material in this section has been deleted and is no longer applicable.

This chapter establishes the policies and procedures for the registration of technical objects defined by the OSI Implementors Workshop. Procedures for registering operational and administrative objects, such as the MHS ADMD and PRMD names and addresses, are outside the scope of this chapter.

6.1 INTRODUCTION AND SCOPE

6.1.1 What is Registration?

In order to communicate, it is necessary to identify the objects involved in communication. These objects have names and addresses. A name identifies an object within the domain of a registration authority. An address is a name that is used to specify the physical or logical location of an object.

OSI names and addresses consist of attributes which are hierarchical in nature and which combine to identify or locate an OSI object unambiguously. Since the relationship between the components of a name or address is hierarchical, it follows that the registration authority for names and addresses should also be hierarchical. A governing organization does not always have sufficient knowledge of organizations lower in the hierarchy to assign values within those organizations. Thus, an approach frequently taken is to delegate registration authority to the lower organizations.

Hierarchy implies an inverted tree-like structure where the number of objects increases from the root of the tree to the leaves of the tree. At the root of the tree, there is one designator that has the greatest scope of authority (largest domain). This designator assigns identifier values to objects under its authority. Each of these objects has a smaller scope of authority than the objects immediately above and may create zero, one, or many subauthorities at the next-lower level. The number of levels in such a tree-like structure is arbitrary.

6.1.2 Scope

This chapter defines registration procedures for OSI Implementors Workshop (OIW) information objects and identifies additional registration requirements. These procedures shall be used by the Special Interest Groups (SIGs) of the Workshop to register information objects used in OSI communications according to the OIW Agreements Document.

In this chapter, the OIW and the SIGs themselves are assigned arcs in the object identifier tree. These arcs are for OIW-specified objects. The SIGs should note that, as national and international registration authorities are established, objects of interest beyond the Workshop are more appropriately registered by a higher level in the hierarchy. This will allow more widespread acceptance of the registered objects.

This chapter is structured as follows: section 6.2 describes the information objects that need to be registered. Section 6.3 describes a registration procedures for OIW object identifiers. Appendix A lists the object identifier component values assigned to the OIW and the SIGs. Appendix B discusses object identifiers used in the 1987 and 1988 Stable Implementation Agreements. The appendices are integral parts of this specification.

6.2 REGISTERED INFORMATION OBJECTS

If networks are to interoperate as envisioned in the OSI model, there must be a universal open and agreed upon naming schema. There are many information objects that fall under this requirement.

Some of the following objects are registered in the standards, some are registered by OIW and others by other registration authorities. An example list of objects to be registered is:

- Application-process-titles
- Application-entity-titles
- Abstract syntaxes
- Transfer syntaxes
- Application-contexts
- MHS
  - ADMD names
  - PRMD names
  - Organization names
  - Encoded information types
  - Extended body part types
  - Extensions
  - etc.

- Object Identifier values
- ASN.1 modules
- Directory
  - Relative distinguished names
  - Attribute types
  - Attribute syntaxes
  - Object classes
  - Encryption algorithms
  - etc.
- VT
  - Profiles
  - Reference information objects
  - etc.
- Network management objects
- Network layer addresses
- System titles
- FTAM
  - Document types
  - Constraint sets
  - etc.
- etc.

The OIW Registration Authority shall only administer information objects created by the OIW Agreements Document that are identified by the ASN.1 type OBJECT IDENTIFIER. Figure 6.1 illustrates the structure of the object identifier component value for OIW.

```
{ iso  identified-organization  oiw(14) }

  iso(1)

       identified-organization(3)

                                   oiw(14)
```

Figure 6.1. Structure of Object Identifier for OIW.

As an example figure 6.2 shows the object identifier component value for an example object.

{ iso identified-organization oiw(14) rasig(13) example(0)}

iso(1)

identified-organization(3)

oiw(14)

rasig(13)

example(0)

Figure 6.2. Structure of an Object Identifier for an Example Object for the Registration Authority SIG of OIW.

The ISO 6523 Registration Authority has assigned an International Code Designator (ICD) value of 14 to OIW, and OIW has assigned a unique object identifier component value to each SIG. The assigned object ID values for the OIW and for each SIG are in Appendix A. The assignment of values below each SIG in the object identifier tree is the responsibility of that SIG.

6.3 REGISTRATION PROCEDURES FOR OBJECT IDENTIFIERS

This section specifies the responsibilities of each SIG and the procedures to be followed for the registration of information objects, and submission to the OIW Plenary.

When an OIW SIG defines an information object the SIG shall register the object identifier. The registered value shall be incorporated into the appropriate OIW Agreements Document as a result of a positive ballot response of the OIW Plenary.

6.3.1 SIG Registration Authorization

An OIW SIG is authorized by its charter and the scope of its work to submit a registration request to the OIW Plenary.

6.3.2 SIG Registration Authority Function and Duties

The SIG Chair is responsible for the assignment, recording and maintenance of the SIG's registered objects. The SIG Chair may appoint a specific person to carry out the SIG duties and responsibilities.

6.3.3 Requirements for Information Object Registration

6.3.3.1 Assignment of Object Identifier Component Values

Each SIG shall register an object identifier component value for each object's technical definition. The NameAndNumberForm of the ObjIdComponent specified in ISO 8824/CCITT X.208 is used exclusively. This form comprises an ASN.1 identifier and, significantly, a NumberForm.

It is suggested that the SIG assign a monotonically increasing integer to the NumberForm at any given level. To the significant root the SIG shall add a assigned object identifier component value that shall be unique. An example of an object identifier created by the RASIG is shown as follows:

{iso(1)identified-organization(3) oiw(14) rasig(13) example(0)}

Technical-Definition:

Example-PrintableString::=-

"This-is-an-example-"

Here rasig is the SIG identifier and 13 is the NumberForm assigned by the OIW Registration Authority (see Appendix A); example is the identifier and 0 is the NumberForm assigned by the RASIG.

6.3.3.2 Proposal of Object and Identifier to Plenary

Registration of an object identifier and its definition is proposed by inclusion of the object identifier and its definition in the OIW "Working Implementation Agreements" document.

6.3.3.3 Completion of Registration Procedure

Registration of an object identifier and its definition is completed upon Plenary vote to move "Working Implementation Agreements" text which contains the object identifier and its definition of to the "Stable Implementation Agreements" document.

6.3.3.4 Changes and Revisions to the Information Object Registration

Neither the technical definition nor the object identifier shall be changed or modified after registration i.e., after the definitions and their identifiers have been voted into the "Stable Implementation Agreements" document.

6.3.4 Register Index

Each SIG shall maintain an index of object identifiers that point to the technical definitions of the respective objects in the OIW Agreements Document. The index shall appear in the appropriate chapter annexes of the OIW Agreements Document.

Index entry example:

Object Identifier	Reference
iso identified-organization oiw(14) rasig(13) example(0)	Chapter 6.3.3.1

6.4 APPENDIX A: ASSIGNMENTS TO WORKSHOP ORGANIZATIONS

Identifier	Value	
oiw	14	(Assigned to OIW by ISO 6523 RA)
llsig	1	(Assigned to SIG by OIW)
nmsig	2	"
secsig	3	"
tpsig	4	"
ftamsig	5	"
mhsig	6	"
dssig	7	"
ulsig	8	"
rdasig	9	"
mmssig	10	"
odasig	11	"
vtsig	12	"
rasig	13	"

6.5 APPENDIX B: STATUS OF 1987 AND 1988 AD-HOC OBJECT IDENTIFIERS

In the 1987 and 1988 versions of the Stable Implementation Agreements, a number of OIW-specified information objects are assigned object identifiers.

OSI requires names and addresses, e.g., object identifiers, be globally unambiguous. This chapter specifies object identifier component values which are globally unambiguous. Other chapters in this document specify the correct object identifiers to be used when referencing OIW-specified information objects.

The use of the 1987 and 1988 OIW-specified object identifiers is deprecated. Newly defined objects shall use the new OIW Identifier.

7. CCITT 1984 X.400 BASED MESSAGE HANDLING SYSTEM

Note: The classification schema used in the chapter (see table 7.7) pre-dated TR 10,000 and was the basis of extensive harmonization, as such: No attempt will be made to align this chapter with TR 10,000.

7.1 INTRODUCTION

This is an implementation agreement developed by the Implementor's Workshop sponsored by the U.S. National Institute of Standards and Technology to promote the useful exchange of data between devices manufactured by different vendors. This agreement is based on, and employs protocols developed in accord with, the OSI Reference Model. While this agreement introduces no new protocols, it eliminates ambiguities in interpretations.

This is an implementation agreement for a Message Handling System (MHS) based on the X.400-series of Recommendations (1984) and Version 5 of the X.400 Series Implementor's Guide from the CCITT. It is recommended that product vendors consult later versions of this guide. Figure 7.1 displays the layered structure of this agreement.

This agreement can be used over any Transport protocol class. In particular, this MHS agreement can be used over the Transport protocol class 0 used over CCITT X.25, described in section 5.2 of this document. In addition, this MHS agreement can be used over the Transport profiles used in support of MAP (Manufacturing Automation Protocol) or TOP (Technical and Office Protocols). Note that the MAP or TOP environment must support the reduced Basic Activity Subset (BAS) as defined in X.410.

The UAs and MTAs require access to directory and routing services. A Directory Service is to be provided for each (vendor-specific) domain. Except insofar as they must be capable of providing addressing and routing described hereunder, these services and associated protocols are not described by this agreement.

User Agent Layer	CCITT X.420
Message Transfer Agent Layer	CCITT X.411
Reliable Transfer Service Layer	CCITT X.410
Presentation Layer	CCITT X.410 Sec. 4.2
Session Layer	See section 5.9

Figure 7.1. The layered structure of this implementation agreement.

7.2 SCOPE

This agreement applies to Private Management Domains (PRMDs) and Administration Management Domains (ADMDs). Four boundary interfaces are specified:

(A) PRMD to PRMD,
(B) PRMD to ADMD,
(C) ADMD to ADMD, and
(D) MTA to MTA (within a PRMD, e.g., for MTAs from different vendors.)

In case A, the PRMDs do not make use of MHS services provided by an ADMD. In cases B and C, UAs associated with an ADMD can be the source or destination for messages. Furthermore, in cases A and B, a PRMD can serve as a relay between MDs, and in cases B and C an ADMD can serve as a relay between MDs. Figure 7.2 illustrates the interfaces to which the agreement applies.

X.400 protocols other than the Message Transfer Protocol (P1) and the Interpersonal Messaging Protocol (P2) are beyond the scope of this agreement. Issues arising from the use of other protocols or relating to P1 components in support of other protocols are outside the scope of this document. This agreement describes the minimum level of services provided at each interface shown in figure 7.2. Provision for the use of the remaining services defined in the X.400 Series of Recommendations is outside the scope of this document.

With the exception of intra domain connections, this agreement does not cover message exchange between communicating entities within a

domain even if these entities communicate via P1 or P2. Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this agreement requires the ability to exchange messages without use of bilateral agreements.

PRMD = Private Management Domain
ADMD = Administration Management Domain

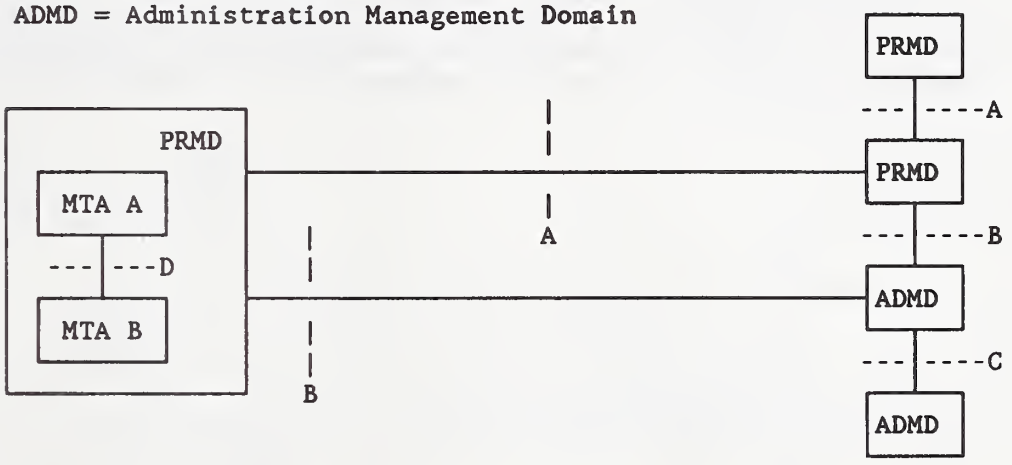


Figure 7.2. This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD; and (D) MTA and MTA.

7.3 STATUS

This version of the X.400 based Message Handling System implementation agreements was completed on December 16, 1988. No further enhancements will be made to this version. See the next section--Errata.

7.4 ERRATA

Defect report material may be in this section, including implementor agreement versions to which it applies.

7.5 PRMD to PRMD

7.5.1 Introduction

This section is limited in scope to issues arising from the direct connection (interface A in fig. 7.2) of two PRMDs. "Direct" means that no ADMD or relaying PRMD provides MHS services to facilitate message interchange. "Direct" does not exclude those instances for which Administrations happen to be ADMDs but are not providing X.400

services, that is, they are used only to provide lower layer services such as X.25. Figure 7.3 schematically represents the scope of this section.

These issues relate to the use of the UAL (User Agent Layer) and MTL (Message Transfer Layer) services, protocol elements, recommended practices and constraints. In particular, this section addresses the P1 and P2 protocols and their related services in a direct connection environment. This section describes the minimum level of services provided by a PRMD. Provision for the use of the remaining services defined in the X.400 Series of Recommendations is beyond the scope of this section.

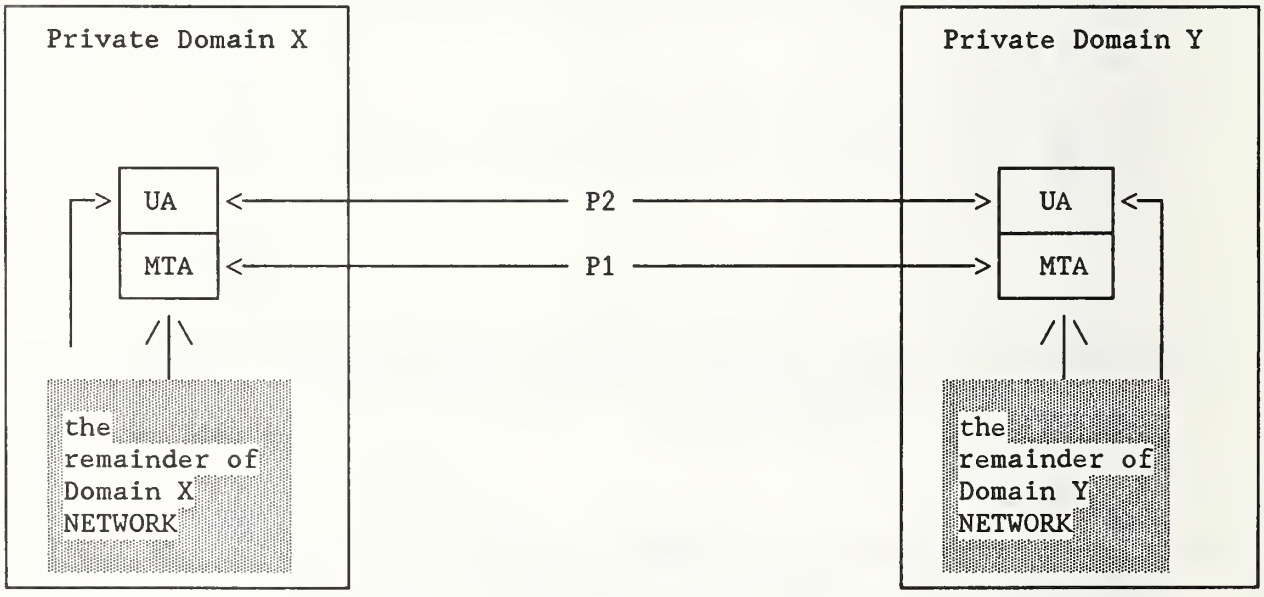


Figure 7.3. Interconnection of private domains.

7.5.2 Service Elements and Optional User Facilities

This section identifies those service elements and optional user facilities that must be provided in support of P1 and P2.

7.5.2.1 Classification of Support for Services

The classification of UA and MT-Service elements is used to define characteristics of equipment. Equipment can claim SUPPORT or NON-SUPPORT of a Service; in the case of

UA-service elements, a separate classification is given for Origination and Reception.

The service provider is defined as the entity providing the service, in this case, the MTL or the UAL. The service user is either the MHS user or the UAL. The classification of provider and user relates to the sublayer for which the service element is defined.

7.5.2.1.1 Support (S)

a) Support means:

- The service provider makes the service element available to the service user, and
  - The service user gives adequate support to the MHS to invoke the service element or makes information associated with the service element available.

b) Support for Origination means that:

- The service provider makes the service element available to the service user for invocation, and
- The service user gives adequate support to the end user of the MHS to invoke the service element.

c) Support for Reception means that:

- The service provider makes information associated with the service element available to the service user.

Note: A UA- or MT-service element can carry information from originator to recipient only if:

- the service element is available to the originator,
- the service element is available to the recipient, and
- all intermediate steps carry the information.

7.5.2.1.2 Non Support (N)

This means that the service provider is not required to make the service element available to the service user. However, the service provider should not regard the occurrence of the corresponding protocol elements as an error and should be able to relay such elements. Implementations making a profile available should indicate deviations (additions or deletions) with respect to the requirement in the profile.

7.5.2.1.3 Not Used (N/U)

This means that although the Recommendation allows this service element, this profile does not use it.

7.5.2.1.4 Not Applicable (N/A)

This means that this service element does not apply in this particular case (for originator or recipient).

7.5.2.2 Summary of Supported Services

a) Within a PRMD, a User Agent must support all P2 BASIC IPM Services (X.400) and all P2 ESSENTIAL IPM Optional user facilities (X.401) subject to the qualifiers listed in Appendix 7A.

b) Within a PRMD, a MTA must support all BASIC MT Services (X.400) and all ESSENTIAL MT optional user facilities (X.401) subject to the qualifiers listed in Appendix 7A.

c) No support is required of the additional optional user facilities of X.401.

7.5.2.3 MT Service Elements and Optional User Facilities

Tables 10.1 through 10.3 show the message transfer (MT) service elements and optional user facilities.

Table 7.1. Basic MT service elements

Service Elements	Support (S) or Non-support (N)
Access Management	N/U[1]
Content Type Indication	S
Converted Indication	S
Delivery Time Stamp Indication	S
Message Identification	S
Non-delivery Notification	S
Original Encoded Information Types Indication	S
Registered Encoded Information Types	N/U[1]
Submission Time Stamp Indication	S

---

1 Not applicable to co-resident UA and MTA.

Table 7.2. MT optional user facilities provided to the UA-selectable on a per-message basis

MT Optional User Facilities	Categorization	Support (S) or Non-support (N)
Alternate Recipient Allowed	E	S
Conversion Prohibition	E	S
Deferred Delivery	E	N[2]
Deferred Delivery Cancellation	E	N[2]
Delivery Notification	E	S
Disclosure of Other Recipients	E	N[3]
Explicit Conversion	A	N
Grade of Delivery Selection	E	S
Multi-destination Delivery	E	S
Prevention of Non-delivery Notification	A	N
Probe	E	N[4]
Return of Contents	A	N

Table 7.3. MT optional user facilities provided to the UA agreed for any contractual period of time

MT Optional User Facilities	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N/U
Implicit Conversion	A	N

E: Essential optional user facility.
A: Additional optional user facility.
2 A local facility subject to qualifiers in Appendix 7A.
3 Support not required for an originating MT user; support must be provided for recipient MT users.
4 Subject to qualifiers in Appendix 7A.

7.5.2.4 IPM Service Elements and Optional User Facilities

Tables 7.4 through 10.5 show the IPM service elements and optional user facilities.

Table 7.4. Basic IPM service elements

Service Elements	Origination by UAs	Reception by UAs
Access Management	N/U[5]	N/U[5]
Content Type Indication	S	S
Converted Indication	N/A	S
Delivery Time Stamp Indication	N/A	S
Message Identification	S	S
Non-delivery Notification	S	N/A
Original Encoded Information Types Indication	S	S
Registered Encoded Information Types	N/A	N/A[5]
Submission Time Stamp Indication	S	S
IP-message Identification	S	S
Typed Body	S	S

5 Does not apply to co-resident UA and MTA.

Table 7.5. IPM optional facilities agreed for a contractual period of time

Service Elements	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N
Implicit Conversion	A	N

Table 7.6. IPM optional user facilities selectable on a per-message basis

IPM Optional User Facilities	Origination by UAs	Reception by UAs
Alternate Recipient Allowed	A (N)	A (N)
Authorizing Users Indication	A (N)	E (S)
Auto-forwarded Indication	A (N)	E (S)
Blind Copy Recipient Indication	A (N)	E (S)
Body Part Encryption Indication	A (N)	E (S)
Conversion Prohibition	E (S)	E (S)
Cross-referencing Indication	A (N)	E (S)
Deferred Delivery	E (N)[6]	N/A
Deferred Delivery Cancellation	A (N/U)[6]	N/A
Delivery Notification	E (S)	N/A
Disclosure of Other Recipients	A (N)	E (S)
Expiry Date Indication	A (N)	E (S)
Explicit Conversion	A (N)	N/A
Forwarded IP-message Indication	A (N)	E (S)
Grade of Delivery Selection	E (S)	E (S)
Importance Indication	A (N)	E (S)
Multi-destination Delivery	E (S)	N/A
Multi-part Body	A (N)	E (S)
Non-receipt Notification	A (N)	A (N)
Obsoleting Indication	A (N)	E (S)
Originator Indication	E (S)	E (S)
Prevention of Non-delivery Notification	A (N)	N/A
Primary and Copy Recipients Indication	E (S)	E (S)
Probe	A (N)	N/A
Receipt Notification	A (N)	A (N)
Reply Request Indication	A (N)	E (S)
Replying IP-message Indication	E (S)	E (S)
Return of Contents	A (N)	N/A
Sensitivity Indication	A (N)	E (S)
Subject Indication	E (S)	E (S)

6 A local facility subject to qualifiers in Appendix 7A.

7.5.3 X.400 Protocol Definitions

This section reflects the agreements of the NIST/OSI Workshop regarding P1 and P2 protocol elements.

7.5.3.1 Protocol Classification

The protocol classifications are defined below in table 7.7:

Table 7.7. Protocol Classifications

1) UNSUPPORTED = X
These elements may be generated, but no specific processing should be expected in a relaying or delivering domain. A relaying domain must at least relay the semantics of the element. The absence of these elements should not be assumed, in a relaying or delivering domain, to convey any significance.

2) SUPPORTED = H
These elements may be generated. However, implementations are not required to be able to generate these elements. Appropriate actions shall be taken in a relaying or delivering domain.

3) GENERATABLE = G
Implementations must be able to generate and handle these protocol elements, although they are not necessarily present in all messages generated by implementations of this profile. Appropriate actions shall be taken in a relaying or delivering domain.

4) REQUIRED = R
Implementations of this profile must always generate this protocol element. However, its absence cannot be regarded as a protocol violation as other MHS implementations may not require this protocol element. Appropriate actions shall be taken in a relaying or delivering domain.

5) MANDATORY = M
This must occur in each message as per X.411 or X.420 as appropriate; absence is a protocol violation. Appropriate actions shall be taken in a relaying or delivering domain.

7.5.3.2 General Statements on Pragmatic Constraints

a) Where a protocol element is defined as a choice of Numeric String and Printable String (i.e., Administration Domain Name and Private Domain Identifier), then a numeric value encoded as a printable string is equivalent to the same value encoded as a numeric string. This does not apply to the Country Name protocol element.

b) The maximum number of recipients in a single MPDU is 32K - 1 (that is, 32767). However, no individual limits on the number of occurrences (recipients) are placed on the following protocol elements: Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, Obsoletes and Cross References. Additionally, there is no limit on the number of Reply to Users. This is a local matter for the originating system.

c) Use of strings. A Printable String is defined in terms of the number of characters, which is the same number of octets. For T.61 strings the number of octets is twice the number of characters specified.

d) The ability to generate maximum size elements is not required, with the exception of the component fields in the Standard Attribute List, in which case it is required.

7.5.3.3 MPDU Size

The following agreements govern the size of MPDUs:

- All MTAEs must support at least one MPDU of at least two megabyte, and
- The size of the largest MPDU supported by a UAE is a local matter.

7.5.3.4 P1 Protocol Elements

7.5.3.4.1 P1 Envelope Protocol Elements

Table 7.8 contains Protocol Elements and their classes.

Table 7.8. P1 protocol elements

Element	Class	Restrictions and Comments
MPDU		
UserMPDU	G	
DeliveryReportMPDU	G	
ProbeMPDU	H	
UserMDPU		
UMPDUEnvelope	M	
UMPDUContent	M	
UMPDUEnvelope		
MPDUIdentifier	M	
originator ORname	M	
originalEncodedInformationTypes	G	If this field is absent, then the Encoded Information Type is "unspecified."
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
Priority	G	
PerMessageFlag	G	Maximum length = 2 octets.
deferredDelivery	X	
PerDomainBilateralInfo	X	No limit on number of occurrences.
RecipientInfo	M	Maximum number = 32K - 1 occurrences. More severe limitations are by bilateral agreement.
TraceInformation	M	
UMPDUContent	M	
MPDUIdentifier		
GlobalDomainIdentifier	M	
IA5String	M	Maximum length = 32 characters, graphical subset only. Refer to T.50 for clarification of graphical subset.
PerMessageFlag		
discloseRecipients	H	
conversionProhibited	G	
alternateRecipientAllowed	H	
contentReturnRequest	X	

(Continued on next page.)

Table 7.8. P1 protocol elements, Continued

Element	Class	Restrictions and Comments
PerDomainBilateralInfo		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName	M	Maximum length = 16 characters.
BilateralInfo	M	Maximum depth = 8; maximum length = 1024 octets (including encoding).
RecipientInfo		
recipient	M	
ExtensionIdentifier	M	Maximum value = 32K - 1 (32767).
perRecipientFlag	M	Maximum length = 2 octets.
ExplicitConversion	X	
perRecipientFlag		
ResponsibilityFlag	M	
ReportRequest	M	
UserReportRequest	M	
TraceInformation		Reference should be made to Version 5 of the X.400 Implementor's Guide for information related to Trace sequencing.
GlobalDomainIdentifier	M	
DomainSuppliedInfo	M	
DomainSuppliedInfo		
arrival	M	
deferred	X	
action	M	
0=relayed (value)	G	
1=rerouted (value)	H	Rerouting is not required.
converted	H	
previous	H	
ORName		See section 7.5.3.5
EncodedInformationTypes		
bit string	M	Delivery can only occur if match is made with Registered Encoded Information Types. Individual vendors may impose limits. Maximum length = 4 octets.
G3NonBasicParameters	X	
TeletexNonBasicParameters	X	
PresentationCapabilities	X	

(Continued on next page.)

Table 7.8. P1 protocol elements, Continued

Element	Class	Restrictions and Comments
DeliveryReportMPDU		
DeliveryReportEnvelope	M	
DeliveryReportContent	M	
DeliveryReportEnvelope		
report	M	
originator	M	
TraceInformation	M	
DeliveryReportContent		
original	M	
intermediate	G	See comment at end of table.
UAContentID	G	
ReportedRecipientInfo	M	Maximum number = 32K - 1 occurrences.
returned	H	Can only be issued if specifically requested in the originating message.
billingInformation	X	Maximum depth = 8; maximum length = 1024 octets (including encoding).
ReportedRecipientInfo		
recipient	M	
ExtensionsIdentifier	M	
PerRecipientFlag	M	
LastTraceInformation	M	
intendedRecipient	H	
SupplementaryInformation	X	Maximum length = 64 characters. Value is pending verification by the CCITT SG VIII or XI.
LastTraceInformation		
arrival	M	
converted	G	
Report	M	

(Continued on next page.

Table 7.8. P1 protocol elements, continued

Element	Class	Restrictions and Comments
Report		
DeliveredInfo	G	Generated if delivery is reported.
NondeliveredInfo	G	Generated if failure to deliver is reported.
DeliveredInfo		
delivery	M	
typeofUA	R	This element must be generated with a PRIVATE value by PRMDs.
NonDeliveredInfo		
ReasonCode	M	
DiagnosticCode	H	Whenever possible, use a meaningful diagnostic code.
ProbeEnvelope		
probe	M	
originator	M	
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
original	G	If this field is absent, then the Encoded Information Type is "unspecified."
TraceInformation	M	
PerMessageFlag	G	
contentLength	H	
PerDomainBilateralInfo	X	
RecipientInfo	M	Maximum number = 32K - 1 occurrences.
GlobalDomainIdentifier		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName (4)	M	Maximum length = 16 characters or digits.
PrivateDomainIdentifier	R	Maximum length = 16 characters or digits. This element must be generated by PRMDs.
End of Definitions		

Notes on table 7.8

Comment on intermediate TraceInformation in the DeliveryReportContent in table 7.8: Audit and confirmed reports should not be requested by other than the originating domain for two reasons. First, the return path of the report may be different from the path taken by the original message, and it may exclude the domain that added the request for audit and confirmed to the message. Second, if the return path is different from the path of the original message, the originating domain would receive intermediate trace information that it did not request.

7.5.3.5 ORName Protocol Elements

Only form 1 variant 1 O/R names are supported.

Table 7.9 contains ORName protocol elements.

These agreements interpret 1984 X.400 section 3.4 to mean that the attributes in the ORName in the MPDU must identify exactly one UA, and that all the values for the attributes specified in the ORName must be identical to the values of the corresponding attributes associated with the recipient UA. Underspecified names that are unique are deliverable.

Overspecified ORNames, ORNames with mismatching fields, and ambiguous names are to be non-delivered or sent to the alternate recipient as appropriate.

Table 7.9. ORName protocol elements

Element	Class	Restrictions and Comments
ORName		
StandardAttributeList	M	
DomainDefinedAttributeList	X	
StandardAttributeList (1)		
CountryName	R	As defined in X.411, Maximum length = 3 characters.
AdministrationDomainName (4)	R	Maximum length = 16 characters or digits.
X121Address	X	Maximum length = 15 digits.
TerminalID	X	Maximum length = 24 characters.
PrivateDomainName (2)	G	Maximum length = 16 characters.
OrganizationName (2)	G	Maximum length = 64 characters.
UniqueUAIdentifier	X	Maximum length = 32 digits.
PersonalName	G	Maximum length of values of sub-elements = 64 characters. Note: The possibility that this value may be reduced to 40 characters is for further study by the CCITT.
OrganizationalUnit (3)	G	Maximum length = 32 characters per occurrence. A maximum of four occurrences are allowed.
DomainDefinedAttributeList (5)		Maximum = 4 occurrences.
type	M	Maximum length = 8 characters.
value	M	Maximum length = 128 characters.
PersonalName		
surName	M	Maximum length = 40 characters.
givenName	G	Maximum length = 16 characters.
initials	G	Maximum length = 5 characters; excluding surname initial and punctuation and spaces.
generationQualifier	G	Maximum length = 3 characters.

(Continued on next page.)

Table 7.9. ORName Protocol Elements, Continued

Notes:

1. The following apply for comparison of the Standard Attributes of an O/R Name:

   a. Lower case is interpreted as upper case (for IA5).

   b. Multiple spaces may be interpreted as a single space. Originating domains shall only transmit single significant spaces. If multiple spaces are transmitted, non-delivery may occur.

2. At least one of these must be supplied.

3. These should be sent in descending sequence, from the most significant <Organizational Unit> (highest in organization hierarchy) to the least significant. Only those specified should be sent. (That is, an unspecified <Organizational Unit> should not be sent along as a field of [null] content, nor zero length, etc.)

4. This attribute shall contain one space in all ORNames of messages originated in a PRMD that is not connected to an ADMD, and in ORNames of recipients reachable only through a PRMD; otherwise, this attribute shall contain an appropriate ADMD name.

5. Some existing systems which will be accessed via an X.400 service (whether directly connected using X.400 protocols or otherwise) may require the provision of addressing attributes which are not adequately supported by Standard Attributes as defined in these Agreements. In such cases, Domain Defined Attributes are an acceptable means of carrying additional addressing information. Failure to support the specification and relaying of DDAs may prevent successful interworking with such existing systems until such time as all systems are capable of relaying and delivery using only the Standard Attribute list. Specific recommendations on the use of DDAs for particular applications are in the Recommended Practices section 7.12, Appendix B.

7.5.3.6 P2 Protocol Profile (Based on [X.420])

Tables 7.10 and 7.11 classify the support for the P2 protocol elements required by this profile. The tables give restrictions and comments in addition to X.420.

Restriction on length is one of the types of restrictions.

The reaction of implementations to a violation of this restriction is not defined by this Profile.

7.5.3.6.1 P2 Protocol - Heading

Table 7.10 below specifies the support for protocol elements in P2 Headings.

Table 7.10. P2 heading protocol elements

Element	Class	Restrictions and Comments
UAPDU		
IM-UAPDU	G	
SR-UAPDU	X	
IM-UAPDU		
Heading	M	
Body	M	
Heading		
IPMessageId	M	
originator	R	
authorizingUsers	H	
primaryRecipients	G	At least one of primaryRecipients,
copyRecipients	G	copyRecipients, or blindCopyRecipients must be
blindCopyRecipients	H	present.
inReplyTo	G	
obsoletes	H	
crossReferences	H	
subject	G	Maximum length = 128 T.61 characters (256 octets);the ability to generate the maximum size subject is not required.
expiryDate	H	
replyBy	H	
replyToUsers	H	
importance	H	Appropriate action is for further study.
sensitivity	H	Appropriate action is for further study.
autoforwarded	H	

(Continued on next page.)

Table 7.10. P2 heading protocol elements, continued

Element	Class	Restrictions and Comments
IPmessageId		
ORName	H	
PrintableString	M	Maximum length = 64 characters.
ORDescriptor		
ORName	H	Specify the ORName whenever it is possible. See Appendix 7B.
freeformName	H	Maximum length = 64 characters, graphical subset only (128 octets.)
telephoneNumber	H	Maximum length = 32 characters. This allows for punctuation. It does not take into account possible future use by ISDN.
Recipient		
ORDescriptor	M	
reportRequest	X	
replyRequest	H	
Body		No limit on number of BodyParts.
BodyPart	G	No limit on length of any BodyPart or the depth of ForwardedIPMessage BodyParts nested. Classification is subject to pending CCITT resolution
SR-UAPDU		
nonReceipt	H	
receipt	H	
reported	M	
actualRecipient	R	
intendedRecipient	H	
converted	X	
NonReceiptInformation		
reason	M	
nonReceiptQualifier	H	
comments	H	Maximum length = 256 characters.
returned	H	May only be issued if specifically requested by originator.

(Continued on next page.)

Table 7.10. P2 heading protocol elements, continued

Element	Class	Restrictions and Comments
ReceiptInformation		
receipt	M	
typeOfReceipt	H	
SupplementaryInformation	X	Maximum length = 64 characters. Note: This value is pending verification by the CCITT SG VIII or IX.
End of Definitions		

7.5.3.6.2 P2 Protocol - BodyParts

a) All BodyParts with identifiers in the range 0 up to and including 16K -1 are legal and should be relayed. BodyPart identifiers corresponding to X.121 Country Codes should be interpreted as described in Note 2 of figure 7.4.

- o Implementations are required to generate and image IA5Text.
- o Implementations should specify the other BodyPart types supported.
- o If an implementation supports a particular BodyPart type for reception, it should also be able to support that BodyPart type for reception if it is part of a ForwardedIPMessage.
- o For the BodyPart types currently considered, support for the protocol elements is as indicated in table 7.11.

b) Privately Defined BodyParts

This section describes an interim means for identifying privately defined BodyParts. This section shall be replaced in a future version taking into account CCITT recommendations with equivalent functionality.

```
BodyPart  :: =  CHOICE {

  [0]IMPLICIT IA5Text,
           [1]IMPLICIT TLX,
                .
                .
                .
           [234]IMPLICIT UKBodyParts,
                .
                .
                .
           [310]IMPLICIT USABodyParts,
                .
                .
                .                          }
          Where UKBodyParts and USABodyParts are defined as:

            SEQUENCE {BodyPartNumber, ANY}

            BodyPartNumber ::= INTEGER
```

Note 1: In the EncodedInformationTypes of the P1 Envelope, the undefined bit must be set when a message contains a privately defined BodyPart. Each UA that expects such BodyParts should include undefined in the set of deliverable EncodedInformationTypes it registers with the MTA.

Note 2: All BodyPartNumbers assigned must be interpreted relative to the BodyPart in which they are used, which is that tagged with the value [310] for those defined within the United States. The NIST assigns unique message BodyPartNumbers for privately defined formats within the United States.

Note 3: Refer to section 7.12.6 for recommendations regarding the implementaion of USABodyParts.

Figure 7.4. X.409 Definition of Privately Defined BodyParts.

7.5.3.6.3 P2 BodyPart Protocol Elements

Table 7.11. P2 BodyParts

Elements	Class	Restrictions and Comments
BodyPart		
IA5Text	G	
TLX	X	
Voice	X	
G3Fax	X	
TIFO	X	
TTX	X	
Videotex	X	
NationallyDefined	X	
Encrypted	X	
ForwardedIPMessage	H	
SFD	X	
TIF1	X	
unidentified	X	
IA5Text		
repertoire	H	
IA5String	M	For rendition of IA5Text see Appendix 7C.
TLX		For further study by CCITT.
Voice		
Set		For further study by CCITT.
BitString	M	
G3Fax		
numberOfPages	X	
P1.G3NonBasicParameters	X	
SEQUENCE (OF BIT STRING)	M	
BIT STRING	H	See Note.
P1.G3NonBasicParameters		Support for individual elements is for further study.
TIFO		
T.73Document	M	
T.73ProtocolElement	H	See Note.

(Continued on next page.)

Table 7.11. P2 BodyParts, continued

Elements	Class	Restrictions and Comments
TTX		
numberOfPages	X	
telexCompatible	X	
P1.TeletexNonBasicParams	X	
SEQUENCE	M	
T61String	H	See Note.
P1.TeletexNonBasicParams		
graphicCharacterSets	X	
controlCharacterSets	X	
pageFormats	X	
miscTerminalCapabilities	X	
privateUse	X	
Videotex		
SET		For further study by CCITT.
VideotexString	M	
NationallyDefined		
ANY	M	
Encrypted		
SET		For further study by CCITT.
BIT STRING	M	
ForwardedIPMessage		
delivery	H	
DeliveryInformation	H	
IM-UAPDU	M	
DeliveryInformation		
P1.ContentType	M	
originator	M	
original	M	
P1.Priority	G	
DeliveryFlags	M	
otherRecipients	H	
thisRecipient	M	
intendedRecipient	H	
converted	X	
submission	M	

(Continued on next page.)

Table 7.11. P2 BodyParts, continued

Elements	Class	Restrictions and Comments
SFD		
SFD.Document	M	
TIF1		
T73 Document	M	
T73.ProtocolElement	H	See note.

Note: This element is not an addition to the definition of the BodyPart. It is described here to show that the SEQUENCE may contain zero elements. A Problem Report has been submitted to the CCITT to clarify whether this is permissible. The NIST/OSI Workshop will adopt the CCITT decision.

7.5.4 Reliable Transfer Server (RTS)

7.5.4.1 Implementation Strategy

Based on X.410 Clause 3 and X.411 Clause 3.5.

7.5.4.2 RTS option selection

a) The maximum number of simultaneous associations is not limited in this profile; if the capacity of a system is exceeded, it should not initiate or accept additional associations.

b) Associations are established by the MTA which has messages to transfer.

c) Associations are released when they are not needed. Associations may also be ended prematurely due to internal problems of the RTS.

d) For both monologue and two way alternate associations, the initiator keeps the initial turn.

e) When establishing an RTS association, the following rules apply to the use of parameters in addition to those in X.410 Clause 3.2.1:

Dialogue mode: Monologue must be supported for this profile; two-way alternate is used only if both partners agree.
Initial turn: Kept by the initiator of the association.

f) The 'priority-mechanism' and the 'transfer-time limit' are regarded as local matters.

7.5.4.3 RTS Protocol Options and Clarifications

Realization of the RTS protocol is subject to the following rules in addition to those specified in X.410 Clause 4:

a) One RTS association corresponds to one or more consecutive session connections (not concurrent ones). The first is opened with ConnectionData of type OPEN, and subsequent ones are opened with type RECOVER.

b) Recovery of a Session connection is only by RTS initiator.

c) Checkpoint size:

- Checkpointing and No Checkpointing should be supported. Whenever possible, checkpointing should be used.
- The minimum checkpointSize is 1 (that is, 1024 octets).

d) Window size:

- Minimal value of 1 (if checkpointing is supported).
- WindowSize = 1 means: After an S-SYNCH-MINOR request is sent, wait until the confirmation is received before issuing an S-DATA, S-SYNCH-MINOR, or S-ACTIVITY-END request.

e) APDUs should not be blocked into one activity.

f) Only one SSDU shall be transferred:

- Between two adjacent minor synch points.
- Between minor synch points and adjacent S-ACTIVITY-START and S-ACTIVITY-END requests.

- o Between S-ACTIVITY-START and S-ACTIVITY-END without checkpoints.

g) A monologue association is defined as follows:

- o The RTS user responsible for establishing the association is called the initiator.
- o The initiator keeps the initial turn.
- o APDUs are transferred in the direction of the initiator to the recipient only.
- o There shall be no token passing.
- o Only the initiator can effect an orderly release of the association.

h) A two-way alternate association is as described in X.410.

i) In the UserData parameter of the S-U-ABORT, the ReflectedParameter will not be used in the AbortInformation element.

j) When the S-ACTIVITY-RESUME is used to resume an activity in the same session connection as the one in which it started, this must happen immediately after the activity has been interrupted (i.e., no intervening activity can occur). Otherwise, X.410 Clause 4.3 paragraph 1 may be violated.

k) When S-ACTIVITY-RESUME is used to resume an activity started in another session connection, the following conditions must be met:

- o The current session connection is of type "recover."
- o The value of OldSessionConnectionIdentifier in S-ACTIVITY-RESUME must match the value of the SessionConnectionIdentifier parameter used in the S-CONNECT of the prior session connection. This value is also identical to the SessionConnectionIdentifier in the ConnectionData (in PConnect, in SS-UserData) for the current session connection.
- o This must occur as the first activity of the next session connection for the same RTS-association. It

must be the first, otherwise X.410 Clause 4.5.1 point 1 is violated.

Note: It is in the same RTS-ASSOCIATION because the use of S-ACTIVITY-RESUME only makes sense within the scope of one RTS association.

l) If the transfer of an APDU is interrupted before the confirmation of the first checkpoint, the value of the SynchronizationPointSerialNumber in S-ACTIVITY-RESUME should be zero, and the S-ACTIVITY-RESUME must be immediately followed by an S-ACTIVITY-DISCARD.

m) In S-TOKEN-PLEASE, the UserData parameter shall contain an integer conforming to X.409 which conveys the priority.

n) The receiving RTS can use the value of the Reason parameter in the S-U-EXCEPTION-REPORT to suggest to the sending RTS that it should either interrupt or discard the current activity. S-U-Exception Reports are handled as stated in Version 5 of the Implementors Guide pages 12-13, paragraph E-33.

o) In the case of S-P-ABORT, the current activity (if any) is regarded as interrupted, rather than discarded.

p) Table 7.12 illustrates the legal negotiation possibilities allowed by X.410 Clause 4.2.1 regarding checkpoint size and window size.

q) These agreements include the provisions of Version 6 of the Implementors Guide identical in all respects to Version 5, except that the following points have been added to section 3.5:

o for section 4.4.1 of X.410;
"If the receiving RTS receives an S-ACTIVITY-DISCARD indication primitive and has already issued a TRANSFER indication primitive, it aborts the connection (S-U-ABORT request) with the "transfer completed" version code."

o for section 4.6.2 of X.410
"The "transfer completed (7)" abort reason is used to indicate to the sending RTS that the receiving RTS could not discard the activity because it has already completed the transfer (issued a TRANSFER indication primitive)." Transfer completed (7) is also added to the table of abort reasons in this section.

Table 7.12. Checkpoint window size of IP

		acceptor answer		
		CS = 0 (or unspecified) WS unspecified	CS = m WS = j (or unspecified)	CS = n WS = j (or unspecified)
initiator proposal	CS = 0 (or unspecified) WS = i (or unspecified)	legal	legal	legal
	CS = k WS = i (or unspecified)	legal	legal	not allowed

Legend:

- CS means CheckpointSize
- WS means WindowSize
- i, j, k, m, and n are integer values with the following relations:

$0 \leq m \leq k < n$ (values assigned to CS)
$0 < j \leq i$ (values assigned to WS)

- For unspecified parameters, the default applies. In this case,the numeric relations apply, that is, the default values substitutefor the unspecified integer.

7.5.4.4 RTS Protocol Limitations

The RTS Protocol Limitations for this profile are listed in table 7.13.

Table 7.13. RTS protocol elements

Element	Class	Restriction
PConnect	M	
DataTransferSyntax	M	Value = 0.
pUserData	M	
checkpointSize	H	
windowSize	H	
dialogueMode	H	
ConnectionData	M	
applicationProtocol	G	Value = 1.
	H	Value = 8883.
ConnectionData		
open	G	
recover	G	
open		
RTS user data	G	
recover		
SessionConnectionIdentifier	G	
RTS user data		
mTAName	G	Maximum length 32 characters graphic subset of IA5 only.
password	G	Maximum length 64 octets graphic subset of IA5 only.
< null RTS User Data >	G	Generated if other validation methods are used.
SessionConnectionIdentifier		
CallingSSUserReference	M	Maximum length 64 octets including encoding = 62 octets of T.61.
CommonReference	M	
AdditionalReferenceInformation	H	Maximum length 4 octets including encoding = 2 octets of T.61.
PAccept	G	
DataTransferSyntax	M	Value = 0.
pUserData	M	
checkpointSize	H	
windowSize	H	
ConnectionData	M	

(Continued on next page.)

Table 7.13. RTS protocol elements, continued

Element	Class	Restriction
PRefuse	G	
RefuseReason	M	
SS User Data (in S-TOKEN-PLEASE)	G	See Note
AbortInformation (in S-U-ABORT)	G	
AbortReason	H	
reflectedParameter	X	Restricted to 8 bits.
End of Definitions		

Note: Generated if supplied by the RTS-user. The RTS use may specify a priority in the TURN-PLEASE primitive, and if so, it is carried as the SS-User-Data in S-TOKEN-PLEASE.

7.5.5 Use of Session Services

The session requirements and use of session are covered in section 5 of this document.

7.5.6 Data Transfer Syntax

This section defines Presentation Transfer Syntax and notation rules applicable to these agreements. Implementations must conform EXACTLY as specified in X.409 with no further restrictions. Appendix 7C defines rendition of IA5 Text and T61 characters.

7.6 PRMD to ADMD and ADMD to ADMD

7.6.1 Introduction

This section defines the implementation agreements that apply to the interface between two management domains when at least one is an ADMD. A message arriving at an ADMD has either no recipient within that domain or one or more recipients within that domain. In the former case, the ADMD serves as a relay between two or

more domains and the actions required of that ADMD are independent of the nature (PRMD or ADMD) of the domains. In the latter case, the ADMD is responsible for delivering messages to the proper recipient(s) within its jurisdiction, and may also be responsible for relaying the message.

Given the two roles for an ADMD, this section describes two distinct sets of functional requirements for an ADMD. The first is the relaying requirement that is needed to provide PRMD and other ADMD interworking. The second is analogous to the PRMD's support to its customers through the integrated UAs. These are distinct functional differences. The services provided to the UAs of an ADMD are independent of the requirement that an ADMD provide a function for interworking with any type of Management Domain (MD). Figure 7.5 illustrates the two roles played by an ADMD.

This section is presented in the form of deviations from the agreements applicable to PRMD-to-PRMD (sec. 7.5). Unless explicitly noted in the remainder of this section, all of the specifications for PRMD to PRMD apply to PRMD to ADMD and ADMD to ADMD.

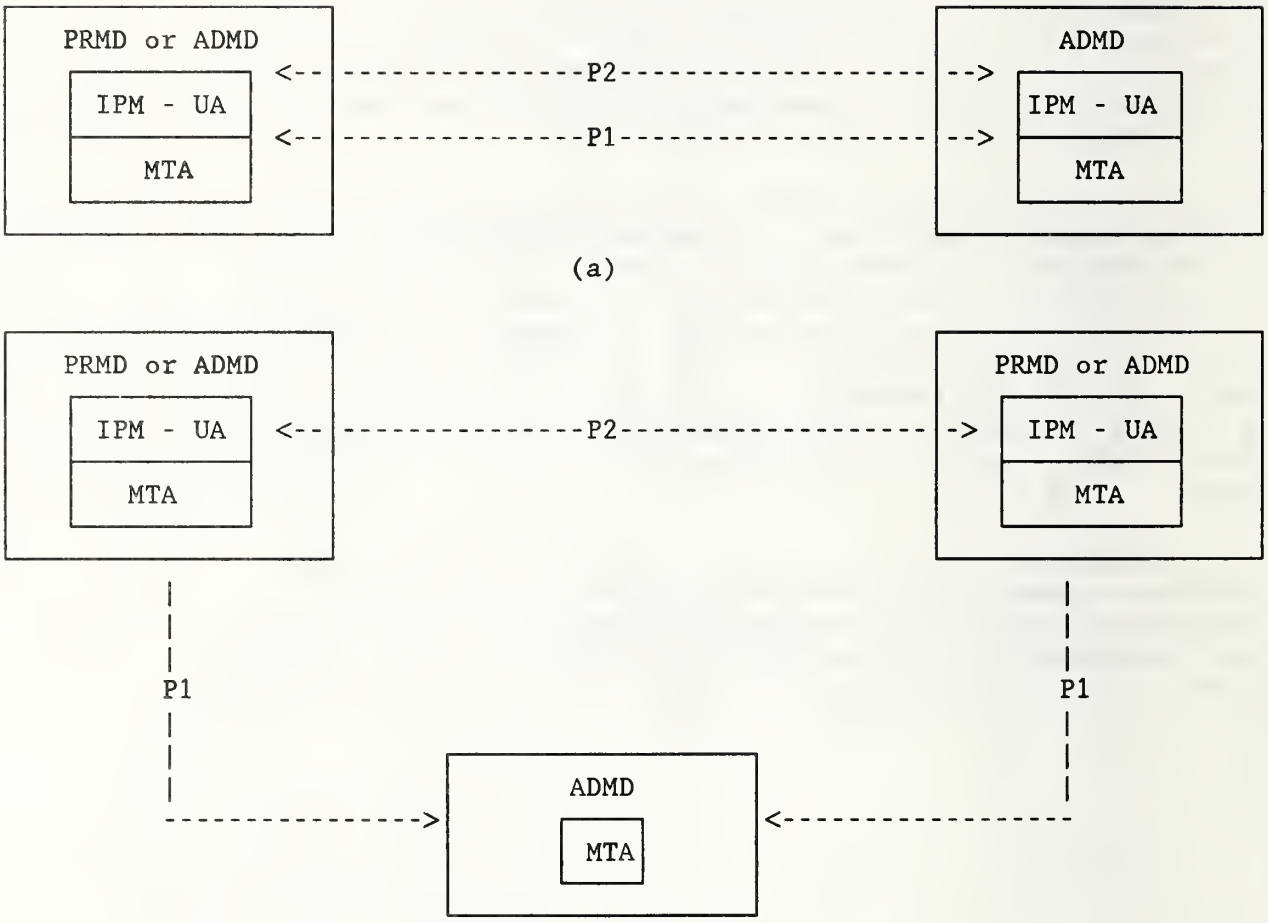


Figure 7.5. An ADMD may (b) or may not (a) serve as a relay.

7.6.2 Additional ADMD Functionality

The following defines the additional ADMD specific functionality required over and above that specified in the PRMD section.

7.6.2.1 Relay Responsibilities of an ADMD

ADMDs will relay all content types (not just P2) unchanged in the absence of a request for conversion.

7.6.2.2 P1 Protocol Classification Changes

Table 7.14 describes the changes to the PRMD P1 Protocol classifications required for a delivering Administration domain (with respect to the original message; this means the domain which originates the delivery reports).

Table 7.14. P1 Protocol Classification Changes for a Delivering ADMD

Protocol Elements	Class	
DeliveredInfo		
typeOfUA	H	
ReportedRecipientInfo		
SupplementaryInformation	H	See Note 1.
GlobalDomainIdentifier		
PrivateDomainIdentifier	H	

For relaying Administration domains, the classifications are all "X"

For originating Administration domains, these are all "NOT APPLICABLE."

Note 1: Domains providing access to TELEX/TELETEX recipients, whether directly or indirectly as a result of bilateral agreements between domains, must ensure that this information, when present, is accessible by the recipient of the delivery report.

7.6.2.3 O/R Names

O/R Names shall consist of:

- CountryName,
- AdministrationDomainName.

as well as one of the following:

- PrivateDomainName,

- o PersonalName,
- o OrganizationName,
- o OrganizationalUnit,
- o UniqueUAIdentifier,
- o X121Address.

and permits the optional inclusion of a

- o DomainDefinedAttributeList.

Note: The destination PrivateDomainName or OrganizationName must be present if destined for a PRMD. The ADMD relaying the message to that destination PRMD requires this element.

7.6.2.4 P1 ADMD Name

Management Domains (MDs) must specify in the ADMD name field of the O/R Name StandardAttributeList in P1, the name of the Administration domain:

- o to which the message is being sent (in recipient names)
- o from which the message originated (in the originator name).

7.6.3 Interworking with Integrated UAs

If the message originates at a UA owned by an ADMD, or is delivered to such a UA, the O/R Name follows the same Form 1 Variant 1 constraints as the base specifications; except that the ADMD name is the name of the ADMD that owns the UA and instead of supplying a PRMD Name, one (or more) of the following must be provided:

- o OrganizationName,
- o OrganizationalUnit,
- o PersonalName.

and may optionally include a

- o DomainDefinedAttributeList.

7.6.4 Differences with Other Profiles

7.6.4.1 TTC Profile

There are no outstanding issues regarding interworking between TTC-conformant systems and NIST-conformant systems with the exception of the number of recipients and the supported MPDU sizes. The ExtensionIdentifier field may contain a maximum value of 32K-1; however, according to the current TTC profile, if a message with more than 256 recipients is received, some TTC-conformant domain may generate a nondelivery notification. This also applies to the ReportedRecipientInfo in a delivery report. Further, a TTC MTA is required to handle an MPDU size of at least 32KB. The NIST required MPDU size is 2MB as covered in section 7.5.3.3. Other differences are shown in Appendix E. TTC is currently based on Version 4 of the Implementor's Guide.

7.6.4.2 CEPT Profile

See Appendix 7E.

7.6.5 Connection of PRMDs to Multiple ADMDs

Given that Management Domain names (both PRMD and ADMD) shall be unique within the U.S., then when an ADMD is presented a message for transfer from a PRMD, it will accept O/R Names (both originator and recipient) which have an AdministrationDomainName field value different than the Administration's name. "Accept" implies the attempt to route/deliver the message shall be made, as appropriate, based upon the knowledge that MD names are unique.

Whether this functionality is required by an Administration for conformance to this agreement is for further study.

If a PRMD is connected to two or more ADMDs which are not effectively connected (either directly or via a third ADMD), full X.400 functionality shall not be available. Problems occur especially in the areas of:

- Naming,
- Routing,
- Replying.

It should be noted that a single PRMD that is connected to more than one ADMD can be represented by more than one combination of

country-name, ADMD-name, and PRMD name. For example, it may occur that the PRMD-name for a particular PRMD may take different values, depending on the ADMD-name. Implementors should be aware of the consequences of these possibilities on routing.

7.6.6 Connection of an ADMD to a Routing PRMD

It is possible for a collection of interconnected private domains to establish one domain as the "gateway" to an ADMD, and hence to the world.

If an ADMD is connected to such a gateway PRMD, the individual private domains shall be registered with the Administration. Administrations need not support such connections.

Note also that upon receipt by the ADMD of a message originating somewhere within the PRMD collection, that the TraceInformation may contain more than one element.

The X.400 Recommendations specify that an ADMD should not attempt to relay a message destined for another ADMD through a PRMD, thus an ADMD should ensure that messages destined for another ADMD are not relayed through a PRMD. It should be noted, however, that a relaying PRMD will relay any such message it receives.

7.6.7 Management Domain Names

- All Management Domain Names (both Private and Administration) shall be unique within the U.S.

- A central naming authority shall be established to register domain names.

7.6.8 Envelope Validation Errors

For validation errors, a non-delivery notice shall be generated (if possible) with reason code of 'unableToTransfer' and diagnostic code of 'invalidParameters' (unless specified otherwise).

ADMDs will validate P1 Envelopes in the following areas:

a) The X.409 syntax of all elements should be checked.

b) The pragmatic constraint limits (lengths of fields and number of occurrences of fields) should be checked.

c) Semantic validation of the following elements should be done:

- o originator O/R Name,
- o recipient O/R Name in the RecipientInfo,
- o Priority.

d) Only recipient Names with the responsibility flag set should be validated. The validation of O/R names is defined in 7.8.3.3; the validation of priority is defined in 7.8.3.7.1.

MPDU Identifier Validation

- o Validation of the GlobalDomainIdentifier component of the MPDU Identifier is performed upon reception of a message (i.e., as a result of a TRANSFER.Indication).

- o The country name should be known to the validating domain, and depending on the country name, validation of the ADMD name may also be possible.

- o Additional validation of the GlobalDomainIdentifier is performed against the corresponding first entry in the TraceInformation. If inconsistencies are found during the comparison, a non-delivery notice with the above defined reason and diagnostic codes is generated.

- o A request will be generated to the CCITT for a more meaningful diagnostic code (such as 'InconsistentMPDUIdentifier').

7.6.9 Quality of Service

7.6.9.1 Domain Availability

7.6.9.1.1 ADMD Availability

The goal is to provide 24 hour per day availability. Note that there will be periods of time when an ADMD may be unavailable due to maintenance windows in its supporting network or in an MTA within the domain.

7.6.9.1.2 PRMD Availability

Although the goal of PRMD availability is also 24 hours per day, business reasons are likely to dictate some different level of availability. ADMDs shall require a profile from the PRMD that indicates its schedule of regular availability to the ADMD.

7.6.9.2 Delivery Times

In the absence of standardized quality of service parameters, the following are agreed to. When standardized parameters from CCITT Study Group I become available, they shall be adopted.

a) In table 7.15 the following delivery time targets are established:

Table 7.15. Delivery Time Targets

Delivery Class	95% Delivered Before
Urgent	3/4 hour
Normal	4 hours
Non-Urgent	24 hours

b) The interval(s) between retries and the number of retry attempts that an ADMD uses in attempting delivery to a PRMD or integrated UA, will be locally determined domain parameters. However, the total elapsed times after which delivery attempts will be stopped are shown in table 7.16. This implies that, after these times, a Non-Delivery Notice will be generated.

An Administration shall continue to attempt delivery until the forced nondelivery time, even if the recipient domain has scheduled an unavailability window.

Table 7.16. Forced Nondelivery Times

Delivery Class	NonDelivery Forced After
Urgent	4 hours
Normal	24 hours
Non-Urgent	36 hours

Note: Both tables apply to the period between acceptance by the originating MTA in the originating Administration domain to the time of delivery in the destination Administration domain. Transit time within PRMDs is NOT included in the above times.

7.6.10 Billing Information

a) All aspects relating to billing, charging, tariffs, settlement, and in particular to the use of the billingInformation field in the delivery report, is subject

to bilateral agreement, and shall not be addressed in these implementation agreements.

b) No ADMD shall require a PRMD to supply or process billing information.

7.6.11 Transparency

a) No P1 extensions, other than the MOTIS extensions are to be allowed (Reference A/3211). Should an ADMD receive a message containing P1 extensions, it shall generate a non-delivery notice (if possible) with reason code of unableToTransfer and diagnostic code of invalidParameters.

If MOTIS elements are present, a relaying MTA can optionally:

- Relay the message. If the MTA does relay, it must not drop any of the protocol elements.
- Non-Deliver the message.

A receiving MTA can optionally:

- Deliver the message
- Non-Deliver the message.

b) The CCITT has been requested to establish a more meaningful diagnostic code (such as protocolError) for this occurrence. Such a code has now been provided in the Implementors Guide.

c) P2 extensions shall be relayed transparently by ADMDs.

7.6.12 RTS Password Management

RTS password management shall be a local matter. This includes:

- password length
- frequency of changes
- exchange of passwords with communicating partners
- loading passwords (i.e., the timing of password changes with respect to active associations).

7.6.13 For Further Study

Issues requiring further study are:

- Intra-Domain Routing
- Multi-Vendor Domains

7.7 INTER and INTRA PRMD CONNECTIONS

7.7.1 Introduction

This section is limited in scope to issues arising from the indirect connection of a PRMD to another PRMD or to an ADMD, and to the interconnection of MTAs to form inter-PRMD connections. Indirect means that the connection is made via a relaying PRMD. The X.400 Recommendations describe the way that a PRMD connects to a ADMD and the way that an ADMD connects to another ADMD. The Recommendations do not, however, describe the way that a PRMD connects indirectly to an ADMD or another PRMD, nor do they describe the way that MTAs are connected within a PRMD. These configurations (shown in figs. 7.6 and 7.7) are useful, for example, in connecting equipment from different vendors at a single customer site.

The P1 protocol and its related services for both inter and intra PRMD connections are addressed in this section. In addition, a method for routing within a PRMD is given. It is recognized that uniform methods for Administration, maintenance, and quality of service should be developed for such configurations, and this work is for further study.

This section describes the minimum that must be provided in order to implement a relaying PRMD and a MTA within a PRMD.

This section is presented in the form of deviations from agreements applicable to PRMD to PRMD connection (sec. 7.5). That is, unless specifically noted in the remainder of this section, the agreements in section 7.5 apply to both relaying PRMDs and MTAs within a PRMD.

It should be noted that the comments regarding ORNames in section 7.6.5 also apply to this section.

7.7.2 The Relaying PRMD

A PRMD that has the capability of relaying messages to another PRMD is called a relaying PRMD. A PRMD implementation need not claim to be a relaying PRMD. A PRMD implementation which does claim to be a relaying PRMD must follow the implementation agreements in this section.

7.7.2.1 Relay Responsibilities of a PRMD

The responsibilities of a relaying PRMD are the same as those of an ADMD (as specified in secs. 7.6.8 and 7.6.2.1). In addition, the PRMD will simply deliver messages routed to it from an ADMD, even if this results in routing a message from the ADMD to the PRMD to another ADMD.

7.7.2.2 Interaction with an ADMD

In order for an ADMD to route a message to ADMD A via ADMD B, it must know that A is reachable through B. Similarly, in order for any MD to route a message to PRMD A via a relaying PRMD B, it must know that A is reachable through B (see fig. 7.8).

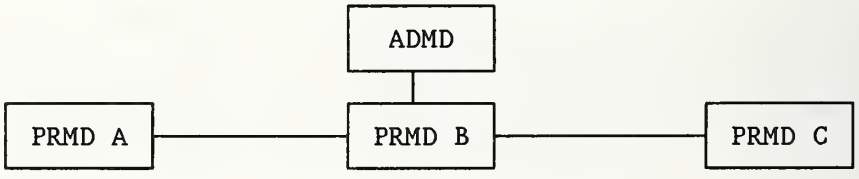


Figure 7.6. Relaying PRMD.

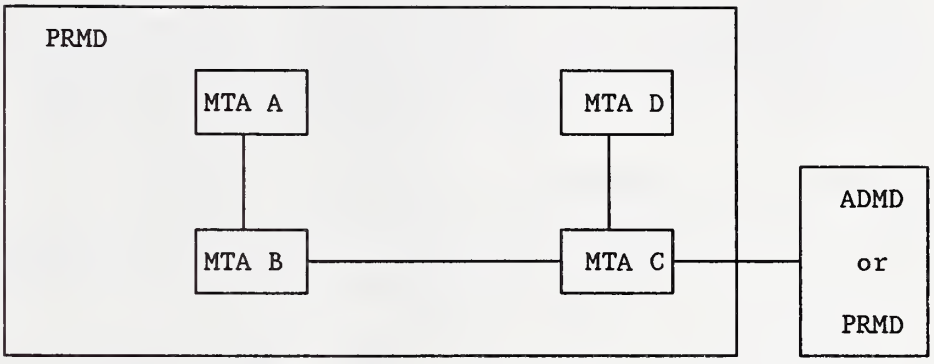


Figure 7.7. Intra PRMD connections.

Note 1: Section 7.6.6 specifies that ADMDs are not required to connect to a relaying PRMD, but they are not precluded from doing so.

Note 2: TraceInformation may have more than one sequence on submission of a message by a relaying PRMD to an ADMD.

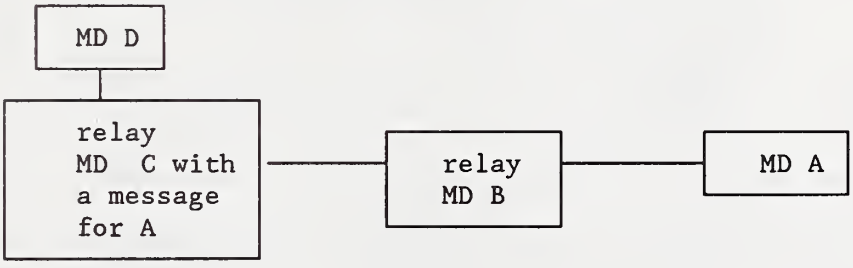


Figure 7.8. MD C must know of A to route the message.

7.7.3 Intra PRMD Connections

A PRMD is composed of MTAs which cooperate to perform the functions expected of a domain. An MTA implementation need not claim to follow the implementation agreements of this section.

7.7.3.1 Relay Responsibilities of an MTA

The relaying responsibilities of an MTA are the same as those of an ADMD (as specified in secs. 7.6.8 and 7.6.2.1) with one exception: loop suppression within the domain is done using the MOTIS InternalTraceInfo protocol element. The MTA must validate the InternalTraceInfo (see

sec. 7.8.3.5 for details on validation). In addition, the PRMD will simply deliver messages routed to it from an ADMD, even if this results in routing a message from the ADMD to the PRMD to another ADMD (please see sec. 7.6.6).

7.7.3.2 Loop Suppression within a PRMD

a) The only mechanism defined in the X.400 Recommendations for suppressing loops is TraceInformation, which is added on a per domain basis to detect and suppress loops among domains. Loops among MTAs within a domain need to be detected and suppressed. This implies that each MTA must add trace information that is meaningful within the domain. The MOTIS solution of adding InternalTraceInfo to the P1 Envelope of a message was adopted. The definition of InternalTraceInfo is given in figure 7.9. The InternalTraceInfo is added by each MTA within a PRMD to handle a message, and it is examined in the same way as TraceInformation to detect and suppress loops.

```
InternalTraceInfo ::= [APPLICATION 30]
   IMPLICIT SEQUENCE OF
   SEQUENCE {
      MTAName,
      MTASuppliedInfo }

MTAName ::= PrintableString
```

Figure 7.9. Definition of InternalTraceInfo.

If the MTAName and password of X.411 are used for validation, then it is recommended that the MTAName used for validation also be used in the InternalTraceInfo. However, there is a complication: in X.411, MTAName is an IA5String, and the MTAname defined by MOTIS is a PrintableString. Efforts will be made to change the MOTIS definition from PrintableString to IA5String.

b) Three actions are defined in MTASuppliedInfo: relayed, rerouted, and recipientReassignment as shown in figure 7.10. The recipientReassignment action is not supported in these agreements. The ability to generate it is not required, and if it is present on an incoming message, the action field will be ignored.

```
MTASuppliedInfo ::= SET {
  arrival [0] IMPLICIT Time,
  deferred [1] IMPLICIT Time OPTIONAL,
  action [2] IMPLICIT INTEGER
     { relayed(0), rerouted(1), recipientReassignment(2) }
  previous MTAName OPTIONAL }
```

Figure 7.10. Defined Actions in MTASuppliedInfo.

7.7.3.3 Routing Within a PRMD

a) Routing within a PRMD is complicated by the lack of a directory standard. In particular, it constrains intra-domain routing decisions to be based on some combination of the intra-domain attributes of the O/R Name, Organization Name Organizational Units, and Personal Name. In order to enhance interworking and to reduce the difficulty of configuring intra-domain connections, it is useful to restrict the ways in which these may be used in making routing decisions.

b) However, it is recognized that vendors may wish to provide MTAs with varying degrees of routing capability within a PRMD as a temporary expedient until appropriate standards for automated construction of directories and routing tables are available. This section assigns class numbers to certain levels of routing capability and discusses the consequences of using MTAs which fall into each class. The classification scheme will allow some diversity in allocating O/R Name space and in configuring intra-domain routes.

c) When other methods are recommended by standards bodies, the classification scheme described here will become obsolete. Large-scale, multi-vendor PRMDs may not be practical in the absence of standardized methods.

7.7.3.3.1 Class Designations

When it is clear that a message is to be delivered within a domain, the Country Name, ADMD Name, and PRMD Name have already served their purpose in determining the next MTA in the route to the recipient. The remaining fields that might be used on making routing decisions within the PRMD are the Organization Name, Organizational Units, and Personal Name.

MTAs are classified by their ability to discriminate between O/R names when making routing decisions within a PRMD. Conformant MTAs will be classified as shown in table 7.17.

Table 7.17. Conformant MTA Classifications

	Class 1	Class 2	Class 3
Organization Name	H	H	H
SEQUENCE OF Organizational Unit	X	H	H
Personal Name	X	X	H

a) An 'H' means that the MTA must be able to base its intra-domain routing decisions on the given component of the O/R Name. In particular, both Class 2 and Class 3 MTAs must be able to discriminate on all the members in a supplied sequence of OrganizationalUnits. A Class 3 MTA must be able to discriminate on all of the elements in a PersonalName.

An 'X' means that the MTA need not have the ability to discriminate on the given component.

b) There is a hierarchy in support of components. The ability to discriminate on a given component does not imply the requirement to do so: e.g., a Class 3 MTA is not required to have tables for every PersonalName in the domain. Equally, an MTA which can discriminate on OrganizationalUnits to make routing decisions need not always use the full sequence in an O/R Name if a partial sequence provides enough information.

c) The above classifications only apply to routing decisions in selecting a next hop within a domain. All MTAs are entitled to examine the full O/R Name when identifying their own directly served UAs.

d) The routing table of a Class 1 MTA will be relatively small, because intra-domain routing decisions are based solely on OrganizationName. The routing table of a Class 2 MTA may be substantially larger and more complex because of its ability to discriminate on OrganizationalUnits as well as OrganizationName to make routing decisions. The routing table of a Class 3 MTA may be larger still,

because its use of the components of PersonalName in addition to the other information.

7.7.3.3.2 Specification of MTA Classes

If an MTA implementation claims to follow the implementation agreements, it must be either a Class 1, Class 2, or a Class 3 MTA. The class of an MTA implementation should be specified so that PRMD administrators can choose equipment properly.

7.7.3.3.3 Consequences of Using Certain Classes of MTAs

Definition: An MTA which accepts submission requests and furnishes delivery indications to a UA is said to "directly serve" the UA.

a) The presence in a domain of an MTA acting as a Class 1 or Class 2 MTA imposes administrative restrictions on the assignment of O/R Names to UAs and in the configuration of routes within that domain.

   - o A Class 1 MTA may directly serve UAs from several OrganizationNames. However, if a Class 1 MTA directly serves a UA with a given OrganizationName, no other MTA in the domain may directly serve a user with the same OrganizationName. This means that if all MTAs in a domain are Class 1, then all UAs with a given OrganizationName must be assigned to the same MTA.

   - o A Class 2 MTA may directly serve UAs from any combination of OrganizationName and sequence of OrganizationalUnits. However, if a Class 2 MTA directly serves a UA with a given combination, no other MTA in the domain may directly serve a user with the same combination. This means that if all MTAs in a domain are Class 2, then all UAs with a given OrganizationName and sequence of OrganizationalUnits must be assigned to the same MTA.

   - o A domain consisting entirely of Class 3 MTAs is free of all the above restrictions.

b) If Class 1 or Class 2 MTAs are used to perform relaying within a PRMD containing MTAs of other classes, care must be exercised in determining the topology of the domain to avoid leaving certain UAs inacessible from certain MTAs within the domain. The example below shows one of the configurations that should be avoided. The example is intended to stimulate careful examination of the relationship between network and organizational topologies.

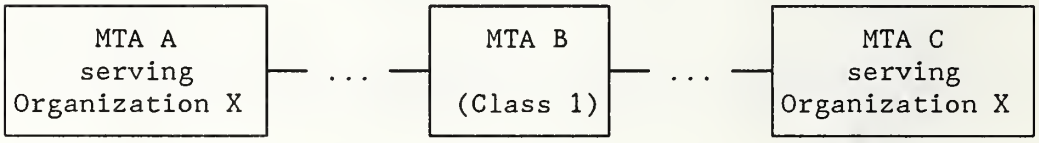


Figure 7.11. Example of a configuration to be avoided.

In figure 7.11, B will route all messages for Organization X to either A or C because B is a Class 1 MTA. The administrator who created this configuration probably wanted B to route some messages for Organization X to A, and some to C. However, B does not have the capability for this because it is only a Class 1 MTA. The configuration in figure 7.11 can be corrected by replacing B with a Class 2 or Class 3 MTA.

7.7.3.4 Uniqueness of MPDUidentifiers Within a PRMD

When generating an IA5String in an MPDUIdentifier, each MTA in a domain must ensure that the string is unique within the domain. This shall be done by providing an MTA designator with a length of 12 octets which is unique within the domain, to be concatenated to a per message string with maximum length of 20 octets.

Two pieces of information, the MTA name and MTA designator, need to be registered within a PRMD to guarantee uniqueness. This registration facility need not be automated. If the MTA name is less than or equal to 12 characters, it is recommended that it also be used as the MTA designator.

7.7.4 Service Elements and Optional User Facilities

A PRMD made up of MTAs which support varying sets of service elements in addition to those required in these agreements may appear to provide inconsistent service for these elements. For example, if one MTA supports deferred delivery and another MTA does not, then deferred delivery can be used by some, but not all, users in the PRMD. Similarly, if one MTA supports return of contents and another does not, then a user outside of the PRMD will receive returned contents for messages sent to one user, but not for messages sent to another user. Note that this same inconsistency occurs when sending to two PRMDs which support different additional optional elements.

7.7.5 X.400 Protocol Definitions

This section describes additions and modifications to section 7.5.3 which are required for implementation of a relaying PRMD or an MTA within a PRMD.

7.7.5.1 Protocol Classification

a) The classification scheme given in section 7.5.3.1 applies to elements passing from one PRMD to another. For both relaying PRMDs, and MTAs in a PRMD, the same classification scheme will be used, but within a PRMD the classification applies to elements passing from one MTA to another.

b) In addition to the classifications given in section 7.5.3.1, a classification of Prohibited has been used.

PROHIBITED = P

This element shall not be used. Presence of this element is a protocol violation.

7.7.5.2 P1 Protocol Elements

Table 7.18 contains protocol elements and their classes. An * signifies that the classification of the protocol element has not changed from table 7.8.

Table 7.18. P1 Protocol Elements

Element	Class	Restrictions and Comments
UMPDUEnvelope		
MPDUIdentifier	M*	This field needs to be unique within a PRMD. See section 7.7.3.4 for the method of ensuring uniqueness.
originator	M*	It is recommended that all components of the originator's ORName be included to help ensure that reports can be returned.
TraceInformation	M*	The first MTA in the domain to receive the message adds the TraceInformation. Subsequent MTAs can update the TraceInformation in the event of conversion or deferred delivery. When a message is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This element is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. Elements are always added to the end of the sequence. (See Note 1)
InternalTraceInfo	M	MTANames within a PRMD must be unique. See section 7.7.3.4 for the method of assuring uniqueness Maximum length = 32 characters.
MTAName		
MTASuppliedInfo	M	

(Continued on next page.)

Table 7.18. P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
MTASuppliedInfo		
arrival	M	
deferred	X	This field must be generated by MTAs which perform deferred delivery.
action	M	See section 7.7.3.2 for restrictions on values of this field.
previous	X	This field must be generated by MTAs which perform rerouting.
DeliveryReportEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the report adds the TraceInformation. When a report is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. (See Note 1)
DeliveryReportContent		
intermediate InternalTraceInfo	P	If it were possible to include this field in the delivery report content, an audit and confirmed report could be provided to detect problems within a PRMD. Efforts are being made to add this field to the MOTIS definition.
DeliveredInfo		
typeOFUA	R*	It is the responsibility of the MTA generating the report to generate this element.

(Continued on next page.)

Table 7.18. P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
ProbeEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the probe adds the TraceInformation. When a probe is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. (See Note 1)

Note 1: The M classification is only applicable if an implementation is claiming conformance according to section 7.10.2, point (g) 4th bullet.

7.7.5.3 Reliable Transfer Server (RTS)

In the pUserData of PConnect, the value of applicationProtocol should be 1. This value was chosen because the agreements on intra-domain connections are not strictly P1, nor are they MOTIS. Philosophically, it would be good to choose a new application protocol identifier for these agreements, but this introduces too many practical problems. Since these agreements are closer to P1 than to MOTIS, the value of 1 will be used. This will not cause interworking problems between domains, because the only deviation from P1 is the InternalTraceInfo, which will not be present in messages transferred outside of a domain.

7.8 ERROR HANDLING

This section describes appropriate actions to be taken upon receipt of protocol elements which are not supported in this profile, malformed MPDUs, unrecognized O/R Name forms, content errors, errors in reports, and unexpected values for protocol elements.

7.8.1 MPDU Encoding

The MPDU should have a context-specific tag of 0, 1, or 2. If it does not have one of these tags, it is not possible to figure out who originated the message. Therefore, the way this error is reported is a local matter.

7.8.2 Contents

Once delivery to the UA has occurred, it is not possible to report errors in P2 information to the originator. In addition, it seems unreasonable to insist that the MTA that delivers a message ensure that the P2 content of the message is acceptable. As a result, the handling of content errors is a local matter.

7.8.3 Envelope

This section describes the handling of errors in message envelopes. Some of the error conditions described below may be detected in a recipient's O/R Name. This may limit the reporting MTA's ability to generate a nondelivery notification that accurately reflects the erroneous O/R Name in the ReportedRecipientInfo. This handling of this situation is a local matter.

7.8.3.1 Pragmatic Constraint Violations

In all cases of pragmatic constraint violation, a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of pragmaticConstraintViolation.

7.8.3.2 Protocol Violations

a) If all required protocol elements are not present, a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of protocolViolation should be generated.

b) If a protocol element is expected to be of one type, but is encoded as another, then a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters should be generated.

7.8.3.3 O/R Names

a) The domain that has responsibility for delivering a message should also have the responsibility to send the nondelivery notification if the message cannot be delivered. Therefore, each MTA should only validate the O/R Names of recipients with responsibility flags set to TRUE. In addition, a nondelivery notification can only be sent if the originator's O/R Name is valid.

b) If any element in the O/R Name is unrecognized or if the CountryName, AdministrationDomainName, and one ofPrivateDomainName and OrganizationName (and, for ADMDs, PersonalName and OrganizationalUnit) are not all present, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of unrecognizedORName. If the message can be delivered even though the ORName is invalid, delivery is a local matter. Note, however, that if the message is delivered, the invalid ORName might be propagated through the X.400 system (e.g., by forwarding).

c) If the O/R Name has all of the appropriate protocol elements and the message still cannot be delivered to the recipient, the following DiagnosticCodes may appear in the nondelivery report: unrecognizedORName, ambiguousORName, and uaUnavailable.

7.8.3.4 TraceInformation

a) Since non-relaying domains need not do loop suppression, domains with responsibility for delivering the message need not be concerned about the semantics of the TraceInformation, that is, arrival time and converted EncodedInformationTypes can be provided to the UA without inspection by the MTAs of the domain as long as the TraceInformation is properly encoded according to X.409.

b) When a message is accepted for relay, the relaying domain must check that a TraceInformation SEQUENCE has been added by the domain that last handled the message. If the appropriate TraceInformation was not added, this should be treated as a protocolViolation (sec. 7.8.3.2).

c) In addition, the relaying domain must check that the information was added in the sequence defined by the

rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid, then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow a more meaningful description of this problem. A request for this new diagnostic code will be submitted.

7.8.3.5 InternalTraceInfo

This section applies only to MTAs which follow the agreements of section 7.7.

a) When a message is accepted for relay from another MTA in the domain, the relaying MTA must check that an InternalTraceInfo SEQUENCE has been added by the MTA that last handled the message. If the appropriate InternalTraceInfo was not added, this should be treated as a protocolViolation (sec. 7.8.3.2).

b) In addition, the relaying MTA must check that the information was added in the sequence defined by the rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid, then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow for a more meaningful description of this problem. A request for this new diagnostic code will be submitted.

7.8.3.6 Unsupported X.400 Protocol Elements

The protocol elements defined in X.400 but unsupported by this profile are: the deferredDelivery and PerDomainBilateralInfo parameters of the UMPDUEnvelope, the ExplicitConversion parameter of RecipientInfo, and the alternateRecipientAllowed and contentReturnRequest bits of the PerMessageFlag. Appropriate actions are described below for domains that do not support the protocol elements.

7.8.3.6.1 deferredDelivery

The delivering domain shall do one of the following:

- deliver at once,
- hold for deferred delivery,
- return a nondelivery notification with a ReasonCode of unableToTransfer and a DiagnosticCode of noBilateralAgreement.

7.8.3.6.2 PerDomainBilateralInfo

If a delivering domain receives this element, the element can be ignored.

7.8.3.6.3 ExplicitConversion

If ExplicitConversion is requested the message should be delivered if possible. That is, if the UA is registered to accept the EncodedInformationTypes of the message, then the message should be delivered even though the requested conversion could not be performed along the route. If delivery is not possible, then a nondelivery report should be generated with a ReasonCode of conversionNotPerformed with no DiagnosticCode.

7.8.3.6.4 alternateRecipientAllowed

If a delivering domain receives this element the element can be ignored.

7.8.3.6.5 contentReturnRequest

If a delivering domain receives this element, the element can be ignored.

7.8.3.7 Unexpected Values for INTEGER Protocol Elements

There are three INTEGERs in the P1 Envelope. Appropriate actions are described below for domains receiving unexpected values for Priority, ExplicitConversion, and ContentType.

7.8.3.7.1 Priority

Additional values for Priority have been suggested by at least one group of implementors as upward compatible changes to the X.400 Recommendations. Therefore, if a PRMD receives an unexpected value for Priority, and this value is greater than one byte in length, a nondelivery report should be generated with a ReasonCode of unableToTransfer and DiagnosticCode of invalidParameters. If the value is less than or equal to one byte, the PRMD can either generate a nondelivery report as previously specified or interpret the Priority as normal and deliver or relay the message.

7.8.3.7.2 ExplicitConversion

When an unexpected value is received for ExplicitConversion, it should be handled as in section 7.8.3.6.3.

7.8.3.7.3 ContentType

If the ContentType is not supported by the delivering MTA, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of contentTypeNotSupported.

7.8.3.8 Additional Elements

In the absence of multilateral agreements to the contrary, receipt of privately tagged elements and protocol elements in addition to those defined in X.400 will result in a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters.

The exceptions to this are the MOTIS elements. The treatment of MPDU's containing these MOTIS extensions is described in section 7.6.11.

7.8.4 Reports

There is no mechanism for returning a delivery or status report due to errors in the report itself. Therefore the handling of errors in reports is a local matter.

7.9 MHS USE OF DIRECTORY SERVICES

7.9.1 Directory Service Elements

a) Recommendation X.400 recognizes the need of MHS users for a number of directory service elements. Directory service elements are intended to assist users and their UAs in obtaining information to be used in submitting messages for delivery by the MTS. The MTS may also use directory service elements to obtain information to be used in routing messages. Some functional requirements of directories have been identified and are listed below.

- o Verify the existence of an O/R name.
- o Return the O/R address that corresponds to the O/R name presented.
- o Determine whether the O/R name presented denotes a user or a distribution list.
- o Return a list of the members of a distribution list.
- o When given a partial name, return a list of O/R name possibilities.
- o Allow users to scan directory entries.
- o Allow users to scan directory entries selectively.
- o Return the capabilities of the entity referred to by the O/R name.
- o Provide maintenance functions to keep the directory up-to-date.

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

c) Currently, these aspects of directory service elements and procedures are under study by both the CCITT and the ISO. Both organizations are committed to the development of a single Directory Service specification for use by MHS and all other OSI based applications.

Given the incomplete nature of the ongoing activities within the CCITT and the ISO, no implementation details will be provided now for MHS use of Directory Services. Implementation agreements for MHS Use of Directory Services will be issued when current activities within the CCITT and the ISO are stable.

7.9.2 Use of Names and Addresses

a) It is recognized that these agreements enable a wide variety of naming and addressing attributes (see sec. 7.5.3.5 ORName Protocol Elements) wherein each PRMD may adopt particular routing schemes within its domain.

b) With the exception of the intra-domain connection agreements:
These agreements make no attempt to recommend a standard practice for electronic mail addressing.

c) Inter-PRMD addressing may be secured according to practices outside the scope of these agreements, such as:

- o manual directories
- o on-line directories
- o ORName address specifications
- o ORName address translation.

d) Further, each PRMD may adopt naming and addressing schemes wherein the user view may take a form entirely different from the attributes reflected in table 7.9. And, each PRMD may have one user view for the originator form and another for the recipient form, and perhaps other forms of user addressing. In some cases (e.g., receipt notification) these user forms must be preserved within the constraints of these implementation agreements. However, mapping between one PRMD user form to another PRMD user form, via the X.400 ORName attributes of these agreements, is outside the scope of these agreements.

7.10 CONFORMANCE

7.10.1 Introduction

In order to ensure that products conform to these implementation agreements, it is necessary to define the types and degrees of conformance testing that products must pass before they may be classified as conformant. This section defines the conformance requirements and provides guidelines for the interpretation of the results from this type of testing.

This section is incomplete and will be enhanced in future versions of this Agreement. Later versions will reflect the problems of conformance testing and will outline specific practices and recommendations to aid the development of conformance tests and procedures.

7.10.2 Definition of Conformance

For this section, the term conformance is defined by the following:

a) The tests indicated for this section are intended to establish a high degree of confidence in a statement that the implementation under test (IUT) conforms (or does not conform) to the agreements of this section.

b) Conformance to a service element means that the information associated with the service element is made accessible to the user (person or process) whenever this agreement says that this information should be available.

   Accessible means that information must be provided describing how a user (person or process):

   - causes appropriate information to be displayed, or
   - causes appropriate information to be obtained.

c) Conformance to P1, P2, and RTS as part of an X.400 OSI application requires that only the external behavior of that OSI system adheres to the relevant protocol standards.

   In order to achieve conformance to this section, it is not required that the inter-layer interfaces be available for testing purposes.

d) Conformance to the protocols requires:

   - that MPDUs correspond to instances of syntactically correct data units,

- o MPDUs in which the data present in the fields and the presence (or absence) of those fields is valid in type and semantics as defined in X.400, as qualified by this profile,

- o correct sequences of protocol data units in responses (resulting from protocol procedures).

e) Statements regarding the conformance of any one implementation to this profile are not complete unless a Protocol Implementation Conformance Statement (PICS) is supplied.

f) The term "Implementation Under Test" (IUT) is interchangeable with the term "system" in the definition of conformance, and may refer to:

- o a domain, which may be one or more MTA's with co-located or remote UA's,

- o a single instance of an MTA and co-located UA with X.400 (P1, P2, RTS and session) software,

- o a relaying product with P1, RTS and session software,

- o a gateway product.

g) Claiming Implementation Conformance

- o An implementation which claims to be conformant as an ADMD must adhere to the agreements in sections 7.5 and 7.6.

- o An implementation which claims to be conformant as a PRMD must adhere to the agreements in section 7.5.

- o An implementation which claims to be conformant as a relaying PRMD must adhere to the agreements in section 7.5 and the appropriate sections of 7.7.

- o An implementation which claims to be conformant to the intra-domain connection agreements must adhere to the agreements in section 7.5 and the appropriate sections of 7.7.

7.10.3 Conformance Requirements

7.10.3.1 Introduction

Conformance to this specification requires that all the services listed as supported in sections 7.5, 7.6, and if appropriate, 7.7 of these agreements are supported in the manner defined, in either the CCITT X.400 Recommendations or these agreements. It is not necessary to implement the recommended practices of section 7.12, Appendix B, in order to conform to these agreements.

It is the intention to adopt, where and when appropriate the testing methodology and/or the abstract test scenarios currently being defined by the CCITT X.400 Conformance Group. However, it is recognized that formal CCITT Recommendations relating to X.400 Conformance Testing will not be available until 1988. It is also recognized that aspects of these agreements are outside the scope of the CCITT, and that other organizations will have to provide conformance tests in these cases.

7.10.3.2 Initial Conformance

This section is intended to provide guidelines to vendors who envisage having X.400 products available prior to any formal mechanism, or "Conformance Test Center" being made accessible that would allow for conformance to this product specification to be tested.

It is feasible that vendors and carriers will want to enter bilateral test agreements that will allow for initial trials to be carried out for the purposes of testing initial interworking capabilities. It is equally feasible that for the purposes of testing interoperability, only a subset of this specification will initially be tested.

> **Note:** By claiming conformance to this subset of information the vendor or carrier CANNOT claim conformance to this entire specification.

There are two aspects to the requirements, interworking and service, as described in the following sections.

7.10.3.2.1 Interworking

The interworking requirements for conformance implies that tests be done to check for the syntax and semantics of protocol data elements for a system as defined by the classification scheme of sections 7.5.2.1.1 and 7.7.5.2. For a relay system, the correct protocol elements should be relayed as appropriate. For a recipient system, a message with correct protocol elements must not be rejected where appropriate.

7.10.3.2.2 Service

For information available to the recipients via the IPMessage Heading and Body, the following should be made accessible:

- IPMessage ID - only the PrintableString portion of the IPMessageId needs to be accessible.
- subject,
- primaryRecipients,
- copyRecipients,
- blindcopyRecipients,
- authorizingUsers,
- originator,
- inReplyTo,
- replyToUsers,
- importance,
- sensitivity,
- IA5Text Bodypart.

7.11 APPENDIX A: INTERPRETATION OF X.400 SERVICE ELEMENTS

The work on service element definitions is limited to those that are defined as 'supported' in section 7.5 of this specification. Furthermore it is not the intent of this section to define how information should be made available or presented to a MHS user, nor is it intended to define how individual vendors should design their products. In addition, statements on conformance to a specific service element and the allocation of error codes that are generated as a result of violations of the service should be defined in the sections on conformance and errors as part of the main product specification. The main objective is to provide clarification, where required, on the functions of a service element, and in particular what the original intent of the Recommendations were.

SERVICE ELEMENTS

The following Service Elements defined in X.400 have been examined and require further text to be added to their definitions to represent the proposed implementation of these service elements by the X.400 SIG.

The service element clarifications are to be taken in the context of this profile.

Service elements not referenced in this section are as defined in X.400.

PROBE

A PRMD need not generate probes.

If a probe is addressed to and received by a PRMD, the PRMD must respond with a Delivery Report as appropriate at the time the probe was processed.

DEFERRED DELIVERY

In the absence of bilateral agreements to the contrary, Deferred Delivery and Deferred Delivery Cancellation are local matters (i.e., confined to the originating domain) and need not be provided.

The extension of Deferred Delivery beyond the boundaries of the initiating domain is via bilateral agreement as specified in section 3.4.2.1 of X.411.

Content Type Indication

It is required that both an originating and recipient domain be able to support P2 content type. The ability for domains to be able to exchange content types other than P2 will depend on the existence of bilateral or multi-lateral agreements.

Original Encoded Information Types Indication

It is required that both an originating and recipient domain be able to support IA5 text. Support for other encoded information types, for the purposes of message transfer between domains, will depend on the existence of bilateral or multi-lateral agreements.

The use of the 'unspecified' form of encoded information type should only be used when the UMPDU content represents an SR-UAPDU or contains an auto-forwarded IM-UAPDU.

The original encoded information type of a message is not meaningful unless a message is converted en route to the recipient. These agreements support only IA5 text, which should not undergo conversion. The original encoded information types should be made accessible to the recipient for upward compatibility with the use of non-IA5 text message body parts.

Registered Encoded Information Types

A UMPDU with an 'unspecified' value for Original Encoded Information Type shall be delivered to the UA.

Delivery Notification

The UAContentID may be used by the recipient of the delivery notification for correlation purposes.

Disclosure of Other Recipients

This service is not made available by originating MTAE's to UAE's, but must be supported by relaying and recipient MTAE's.

By supporting the disclosure of other recipients the message recipient can be informed of the O/R names of the other recipient(s) of the message, as defined in the P1 envelope, in addition to the O/R Descriptors within the P2 header.

These agreements do not support initiation of disclosure of other recipients, but the information associated with it should be made accessible to the recipient for upward compatibility with support for the initiation of this service element.

Typed Body

As defined in X.400 with the addition of the Private Body Types that are to be supported. At present there is no mechanism provided within X.420 that would allow you to respond to reception of an unsupported body type.

Action taken in this situation is a local matter.

Blind Copy Recipient Indication

It should be considered that the recipient's UA acts on behalf of the recipient, and therefore may choose to disclose all BCC recipients to each other. Therefore it is the responsibility of the originating domain to submit two or more messages, depending on whether or not each BCC should be disclosed to each other BCC.

Auto Forwarded Indication

A UA may choose not to forward a message that was previously auto-forwarded. In addition there is no requirement for an IPM UA that does not support non-receipt or receipt notification to respond with a non-receipt notification when a message is auto-forwarded.

Primary and Copy Recipients Indication

It is required that at least one primary recipient be specified; however, for a forwarded message this need not be present. The recipient UA should be prepared to accept no primary and copy recipients to enable future interworking with Teletex, Fax, etc.

Sensitivity Indication

A message originator should make no assumptions as to the semantic interpretation by the recipients UA regarding classifications of sensitivity. For example, a personal message may be printed on a shared printer.

Reply Request Indication

In requesting this service an originator may additionally supply a date by which the reply should be sent and a list of the intended recipients of the reply. If no such list is provided then the initiator of the reply sends the reply to the originator of the message and any recipients the reply initiator wishes to include. The replytoUsers and the replyBy date may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request. Note that for an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Body Part Encryption

The original encoded information type indication includes the encoded information type(s) of message body parts prior to encryption by the originating domain. The ability for the recipient domain to decode an encrypted body part is a local matter. Successful use of this facility can only be guaranteed if there exists bilateral agreements to support the exchange of encrypted body parts.

Forwarded IP message Indication

The following use of the original encoded information type in the context of forwarded messages is clarified:

- If forwarding a private message body part the originator of the forwarded message shall set the original encoded information types in the P1 envelope to undefined for that body part.
- The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.
- See Appendix 7B on recommended practices for the use of the delivery information as part of Forwarded IP-message.

Multipart Body

It is the intent of multipart bodies to allow for the useful and meaningful structuring of a message that is constructed using differing body part types. For example, it is not recommended that a message made up of only IA5 text should be represented as a number of IA5 body parts, each one representing a paragraph of text.

7.12 APPENDIX B: RECOMMENDED X.400 PRACTICES

It is not necessary to follow the recommended practices when claiming conformance to these agreements.

7.12.1 RECOMMENDED PRACTICES IN P2

1. ORDescriptor

   Vendors following the NIST/OSI Workshop guidelines shall, whenever possible, generate the ORName portion of an ORDescriptor in ALL IPM heading fields.

2. ForwardedIPMessage BodyParts

   ForwardedIPMessage BodyParts should be nested no deeper than eight. There is no restriction on the number of ForwardedIPMessage BodyParts at any given depth.

3. DeliveryInformation

   It is strongly recommended that DeliveryInformation be supplied in both forwarded and autoforwarded message body parts. DeliveryInformation is useful when a message has multiple forwarded message body parts because without it, the EncodedInformationType(s) of the component forwarded messages cannot be deduced easily. DeliveryInformation is useful for autoforwarded messages because the EncodedInformationType of an autoforwarded message is "unspecified" and the EncodedInformationType(s) of the message cannot be determined easily without it. Absence of the EncodedInformationType(s) makes it difficult for a UA to easily determine whether the message can be rendered.

7.12.2 RECOMMENDED PRACTICES IN RTS

1. In the case where S-U-ABORT indicates a temporaryProblem, reestablishment of the session should not be attempted for a "sensible" time period (typically not less than 5 minutes).

   In instances where this delay is not required or necessary, report a localSystemProblem.

2. S-U-EXCEPTION-REPORT reason codes can be interpreted as follows:

   - receiving ability jeopardized (value 1)

Possible meaning: The receiving RTS knows of an impending system shutdown.

- o local ss-User error (value 5)
  Possible meaning: The receiving RTS needs to resynchronize the session dialogue.

- o irrecoverable procedure error (value 6)
  Possible meaning: The receiving RTS has had to delete a partially received APDU, even though some minor synchronization points have been confirmed.

- o non specific error (value 0)
  Possible meaning: The receiving RTS cannot handle the APDU (for example, because it was too large) and wishes to inform the sending RTS not to try again.

- o sequence error (value 3):
  Possible meaning: The S-ACTIVITY-RESUME request specified a minor synchronization point serial number which does not match the checkpoint data.

3. For purposes of identifying an MTA during an RTS Open, OSI addressing information should be used. This addressing information is conveyed by lower layer protocols and is reflected by the calling and called SSAP parameters of the S-CONNECT primitives.

   MTA validation and identification are related, but separate, functions. The mTAName and password protocol elements of the RTS user data should be used for validation, rather that identification, of an MTA. The RTS initiator and responder may independently require each other to supply mTAName and password.

   The CallingSSUserReference parameter of the S-CONNECT primitives should only have meaning to the entity that encoded it and should not be used to identify an MTA.

7.12.3 RECOMMENDED PRACTICES FOR ORName

Table 7.9 stipulates that the StandardAttributeList must contain either PrivateDomainName or OrganizationName. It is recommended that, for both originator and recipients in a private domain, the PrivateDomainName field be used.

It is recommended that there should be a DomainDefinedAttribute to be used in addressing UAs in existing mail systems, in order to curtail the proliferation of different types of

DomainDefinedAttributes used for the same purpose. The syntax of this DomainDefinedAttribute conforms to the CCITT Pragmatic Constraints, and thus has a maximum value length of 128 octets and a type length of 8 octets, each of type Printable String. Only one occurrence is allowed.

This DomainDefinedAttribute has the type name "ID" (in uppercase). It contains the unique identifier of the UA used in addressing within the domain. This DomainDefinedAttribute is to be exclusively used for routing within the destination domain (i.e., once routed to that domain via the mandatory components of the StandardAttributeList); any other components of the StandardAttributeList may be provided. If they conflict delivery is not made.

The contents of this parameter need not be validated in the originating domain or any relaying domain, but simply transferred intact to the next MTA or domain.

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

There are existing mail systems that include a small set of non-Printable String characters in their identifiers. For these systems to communicate with X.400 messaging systems, either for pass-through service or delivery to X.400 users, gateways will be employed to encode these special characters into a sequence of Printable String characters. This conversion should be performed by the gateway according to a common scheme and before insertion in the ID DDA, which is intended to carry electronic mail identifiers. X.400 User Agents may also wish to perform such conversions.

It is recommended that the following symmetrical encoding and decoding algorithm for non-Printable String characters be employed by gateways. The encoding algorithm maps an ID from an ASCII representation to a PrintableString representation. Any non-printable string characters not specified in the table are covered by the category "other" in the table below.

The principal conversion table for the mapping is as follows:

Table 7B.1. Printable string to ASCII mapping

ASCII Character	Printable String Character
% (percent)	(p)
@ (at sign)	(a)
! (exclamation)	(b)
" (quote mark)	(q)
_ (underline)	(u)
((left paren.)	(l)
) (right paren.)	(r)
other	(3DIGIT)

where 3DIGIT has the range 000 to 377 and is interpreted as the octal encoding of an ASCII character.

To encode an ASCII representation to a PrintableString, the table and the following algorithm should be used:

```
IF current character is in the encoding set THEN
     encode the character according to the table above
ELSE
   write the current character;
continue reading;
```

To decode a PrintableString representation to an ASCII representation, the table and the following algorithm should be used:

```
IF current character is not "(" THEN
   write character
ELSE
   {
   look ahead appropriate characters;
   IF composite characters are in the above table THEN
      decode per above table
   ELSE
      write current character;
   }
continue reading;
```

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of

OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

7.12.4 POSTAL ADDRESSING

For domains wishing to support postal (or physical) delivery options, the following interim set of "nationally-defined" domain defined attributes are recommended. The CCITT will define Standard Attributes in support of physical delivery in its 1988 Recommendations; this is only an interim solution.

CCITT will also be addressing the services associated with physical delivery. This interim solution does not address the end-to-end service aspects of physical delivery; in particular, the following IPM service elements do not currently extend outside of the X.400 environment:

- o alternate Recipient Assignment
- o PROBE
- o Receipt Notification / Non-Receipt Notifications
- o Grade of delivery

"Delivery" means passing a message from the MTS to the physical delivery system (PDS), and not to the user (or user agent).

The following three DDAs are recommended to be used to specify a postal (or physical) address:

CNTRPC encodes the country and postal code for postal delivery. The DDA value is of the form "Country?Postalcode" (for example, "USA?22096"). The country field is optional, the postal code is optional; the separator ("?") is not. If both country and postal code are missing, this DDA should not be specified.

PDA 1 The country and postal code fields are free-form text.

PDA 2 These two DDA (signifying Postal Delivery Address strings 1 and 2) form a 256 character free-form postal address. Fields are separated by a question mark ("?"). There is no implied separator between PDA1 and PDA2. The meaning of the fields are defined by each domain supporting the physical delivery interface. PDA1 contains the first 128 characters, PDA2 the next 128 characters. If the PDA string is less than 128 characters, PDA2 is not used.

For example, if the domain interprets the PDA fields as lines, the address

Mr. John Smith
Conway Steel
123 Main Street
Reston VA 22096

would be encoded as follows:

type = "PDA1" value = "Mr. John Smith?Conway Steel?123 Main Street?Reston VA"
CNTRPC = "?22096"

7.12.5 EDI use of X.400

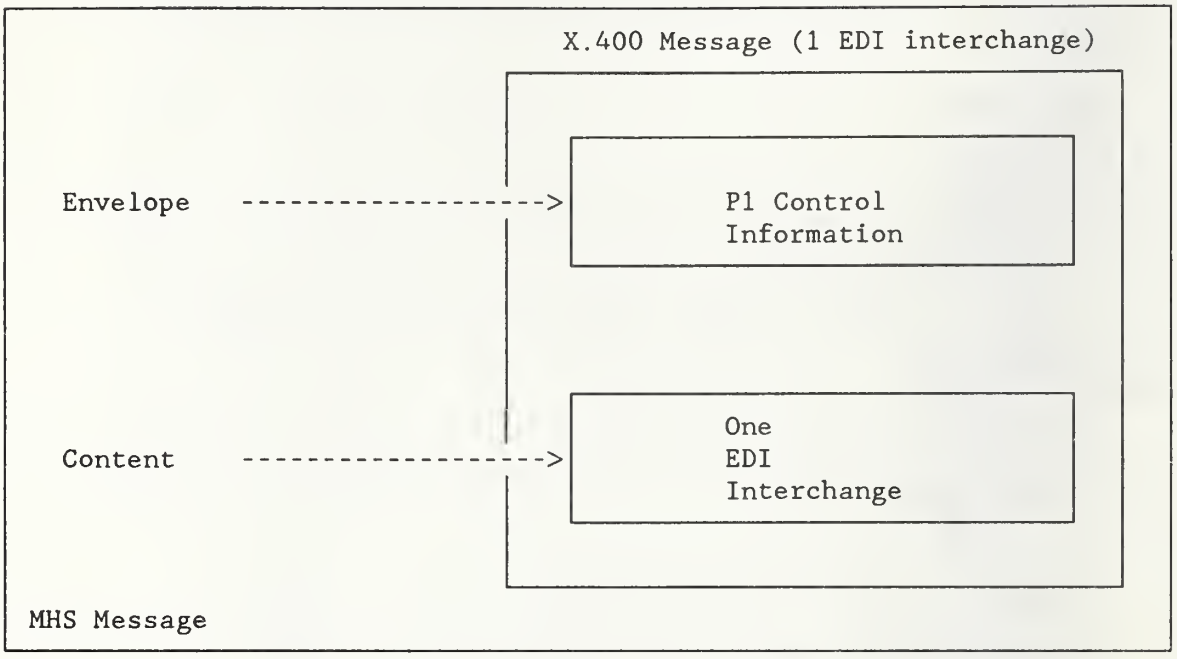
7.12.5.1 Introduction and Scope

This is a guideline for EDI data transfer in an X.400 environment conforming to the NIST agreements. These recommended practices outline procedures for use in transferring EDI transactions between trading partner applications in an attempt to facilitate actual X.400 implementation by EDI users.

The scope of this guideline is to describe specific recommendations for adopting X.400 as the data transfer mechanism between EDI applications.

7.12.5.2 Model

The MHS recommendations can accommodate EDI through the approach illustrated below. Many Message Transfer (MT) service elements defined in the X.400 recommendations are particularly useful to the EDI application.



This diagram depicts an EDI content (1 EDI interchange) enveloped by the P1 MHS envelope. All the MT Services defined in the X.400 Recommendations may be used for EDI. However, it is not required to support optional or non-essential services to exchange EDI data between EDI users. When an EDI user submits an EDI Trade Document to the EDI User Agent, the EDI UA will submit the EDI content plus P1 envelope to the Message Transfer System (MTS).

The EDI UA must support the essential MT Services as defined in these Agreements; for example, as a minimum, to provide default values for services not elected by the EDI user, such as Grade of Delivery.

Note: MT Services are not necessarily made available by the EDI UA to the EDI user.

7.12.5.3 Protocol Elements Supported for EDI

The following P1 protocol elements will be used to support EDI applications:

Content Type
For EDI applications, the content type will be 0 (undefined content).

Original Encoded Information Types
Any EIT defined in the X.400 Recommendations may be used to specify the encoding of EDI content. However, for ANSI X12 EDI applications in particular, it is expected that the "undefined" and "Ia5Text" EIT's will normally be used, with "undefined" used to signify the EBCDIC character set.

7.12.5.4 Addressing and Routing

It is anticipated that connection of some existing systems to an X.400 service for EDI purposes will be by other than X.400 protocols, at least in the short term.

EDI messages entering the X.400 environment will therefore need to have X.400 O/R Names added to identify the origination and recipient trading partners, typically by means of local directory services in the origination domain which will map EDI identifiers/addresses into O/R Names. Such O/R Names will contain Standard Attributes as defined in table 7.9 and for recipient trading partners will at least identify the destination domain.

In the case of trading partners outside the X.400 environment, it is expected, however, that there will be cases where message delivery will require the provision of addressing information beyond that which can be carried in Standard Attributes. In such cases, Domain Defined Attributes are recommended to be used.

The syntax of this DDA is as defined in table 7.9, with a single occurrence having the type name "EDI" (uppercase) and a value containing the identifier/address of the trading partner. For ASC X12 purposes, specifically, this value will comprise the 2 digit interchange ID qualifier followed by the interchange ID (max 15 characters). Routing on this DDA shall only occur, if at all, in the destination domain.

7.12.6 USA Body Parts

It is recommended that UAs can generate any USA Body Part, as defined in Section 7.5.3.6.2, and that they can receive such body parts as well. reception of USA Body Parts does not imply further processing by the UA, but merely that the body part is made available, with a indication of its registered body part identifier, to another process or deposition in a file. Generation implies the reverse of this process.

7.12.7 Recommended Practices for Binary Data Transfer

The capability to transfer binary data, such as those generated by word/document processing, spreadsheets, or graphics applications among X.400 system is a useful and desirable feature. Many messaging systems provide such capability today.

It is recommended that transfer of binary data through 1984-based systems be achieved using the Unidentified BodyPart in P2 with the ASN1 definition recaptured as follows:

```
BodyPart      ::= CHOICE {
                    [0] IMPLICIT IA5Text
                    ...
                    [14] IMPLICIT Unidentified
                    ...}

Unidentified  ::= OCTET STRING
```

Note: the Unidentified BodyPart is included in 1984 X.400 Implementor's Guide, Version 6, and is renamed as BilaterallyDefinedBodyPart in 1988 X.400 Series with the same tag and definition.

Additionally the binary data can be identified by a text string in the subject heading or in an IA5Text body part preceding the Unidentified BodyPart.

When the Unidentified BodyPart is present in a P2 message, the undefined(0) bit of the P1 EncodedInformationTypes will be set. If the IA5Text bodypart is also present, the IA5text(2) bit will also be set.

The binary data is the raw data as generated by user applications. Besides encapsulating it for transfer purposes, X.400 systems do not encode or interprete the binary data in any way further. How the data is encoded or decoded is defined by the cooperating user

applications. How the data is injected into X.400 systems or transferred out of X.400 systems to the user applications, or how the user applications are invoked to process the data is a local implementation issue and not defined.

7.12.8 Recommended Practice for Office Document Architecture (ODA) Transfer

It is recommended that the conveyance of ODA documents through 1984-based X.400 systems be achieved using the following schemes:

(a) In P2:
An ODA document will be transferred as a single body part with tag 12, recaptured as follows:

```
BodyPart  ::=  CHOICE  {
                [0]  IMPLICIT IA5Text
                ...
    oda         [12] IMPLICIT OCTET STRING
                ... }
```

The content of the Octet String will contain a value of type OdaBodyPart as follows:

```
OdaBodyPart  ::=  SEQUENCE  {
                    OdaBodyPartParameters,
                    OdaData  }
```

The Parameters and Data components are defined in Annex E of CCITT Recommendation T.411 (1988) (ISO 8613-1).

(b) In P1:
Both the undefined bit (bit 0) and the ODA bit (bit 10) of the EncodedInformationType will be set when an ODA document is present in P2.

7.13 APPENDIX C: RENDITION OF IA5Text AND T61String CHARACTERS

7.13.1 GENERATING AND IMAGING IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations:

CR LF	to start a new line
CR FF	to start a new page (and line)
LF .. LF	to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect. The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition. Note that X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

7.13.2 GENERATING AND IMAGING T61String

For further study.

7.14 APPENDIX D: DIFFERENCES IN INTERPRETATION DISCOVERED THROUGH TESTING OF THE MHS FOR THE CeBit 87 DEMONSTRATION

Several interworking problems were discovered through multi-vendor testing. These problems, and recommendations for solutions to them are discussed in this appendix.

7.14.1 ENCODING OF RTS USER DATA

The password is defined as an ANY in the X.400 Recommendations, and implementor's groups have decided to use an IA5String for this field. There was some confusion about what the X.409 encoding for this IA5String would be, and the correct encoding is:

```
class:    context specific
form:     constructor
id code:  1
length:   length of contents
contents:     (primitive encoding)
     IA5String:  16
     length:     length of contents
     contents:       the password string
class:    context specific
form:     constructor
id code:  1
length:   length of contents
contents:     (constructor encoding) left as an exercise for the
              reader
```

Implementations should be prepared to receive any X.409 type for the password because of its definition as an ANY.

7.14.2 EXTRA SESSION FUNCTIONAL UNITS

One vendor proposed more than the required set of functional units on opening the session connection, and the receiver rejected the connection. All debate aside about whether the initiator should have proposed units outside of the required set, or whether the receiver should have rejected the connection, the set of functional units can be negotiated in a straightforward way. The following is recommended.

If the initiator proposes using more than the required set of functional units, the responder should specify the set of functional units that it would like to use (which should include the required set) in the open response. The session

implementations will automatically use the intersection of the units proposed by both sides.

If the initiator proposes using less than the required set of functional units, the responder should reject the connection. Unfortunately, there is not an appropriate RefuseReason for rejecting the connection, so instead of refusing the connection in the response to the S-CONNECT, the receiver should issue an S-U-ABORT with an AbortReason of protocolError. Note that it is valid to issue an S-U-ABORT instead of responding to the S-CONNECT. A problem report has been submitted to the CCITT requesting the addition of a RefuseReason for this situation.

If the responder proposes using less than the required set of functional units, the session connection is established before the initiator can check for this. If too few functional units have been proposed, the initiator should abort the connection using S-U-ABORT, with an abort reason of protocolError.

7.14.3 MIXED CASE IN THE MTA NAME

The MTA name is frequently exchanged over the telephone when two systems are being configured to communicate with one another. In one such telephone exchange, the casing of the MTA name was not specified, the MTA name consisted of both upper and lower case letters, and one of the implementations compared MTA names for equality in a case sensitive manner. Consequently, connections failed until the problem was detected and repaired. It is recommended that the MTA name be compared for equality in a case insensitive manner, and that the password be compared for equality in a case sensitive manner.

7.14.4 X.410 ACTIVITY IDENTIFIER

The X.400 Implementor's Guide recommends that the activity identifier be X.409 encoded, but this is only a recommendation and not a requirement. Consequently, receiving systems cannot assume that the activity identifier will be X.409 encoded.

7.14.5 ENCODING OF PER RECIPIENT FLAG AND PER MESSAGE FLAG

In the definition of the PerRecipientFlag in X.411, there is a statement that the last three bits are reserved, and should be set to zero. It is unclear whether those bits are unused in the X.409 encoding. Receivers should accept encodings with either zero or three unused bits. A problem report has been submitted to the CCITT asking for clarification.

Though there is not any statement in X.411 about the last four bits of the PerMessageFlag, some vendors have encoded this with zero unused bits, and some have encoded it with four unused bits. The PerMessageFlag should be encoded with at least four unused bits.

7.14.6 ENCODING OF EMPTY BITSTRINGS

There are three valid encodings for an empty bitstring: a constructor of length zero, a constructor of indefinite length followed by the end-of-contents terminator, and a primitive of length one with a zero octet as the value.

7.14.7 ADDITIONAL OCTETS FOR BITSTRINGS

Nothing in X.409 constrains an implementation from sending two, three, four, or even more octets for a bitstring that fits into one octet, with the undefined bits set to zero. Note that the number of excess octets is bounded by the pragmatic constraints guidelines of the CCITT X.400 Implementor's Guide for all of the bitstrings in P1.

7.14.8 APPLICATION PROTOCOL IDENTIFIER

If a value other that 1 is received in the applicationProtocol of the pUserData in the PConnect, NIST implementations will reject the connection. If CEN/CENELEC implementations receive a value other than 8883 for this field, they will reject the connection. This is an unfortunate state of affairs, because if NIST implementations accept the value of 8883 without supporting the MOTIS service elements, they would be misrepresenting themselves. To make matters worse, CEPT uses a value of 1, but relays MOTIS elements, which means that MOTIS elements will be relayed to implementations using a value of 1 to demonstrate that they do not support MOTIS. Work is continuing to try to find a solution that will allow European implementations to interwork with U.S implementations.

7.14.9 INITIAL SERIAL NUMBER IN S-CONNECT

This should be implemented in accordance with section 3.5.1 E4 of the Implementors' Guide.

7.14.10 CONNECTION DATA ON RTS RECOVERY

It is clarified that the ConnectionData is identical in both the S-CONNECT.request and the S-CONNECT.response. The value of the ConnectionData is the old Session Connection Identifier.

7.14.11 ACTIVITY RESUME

If an activity is being resumed on a new session connection, it is not clear from X.410 and X.225 whether all four of the called-ss-user reference, the calling-ss-user reference, the common reference, and the additional reference information should be specified in the S-ACTIVITY-RESUME, or whether one of the ss-user-references should be absent. It is also unclear whether the called-ss-user reference should be identical to the calling-ss-user reference if both are present. Consequently, receivers should be tolerant of this situation. Appropriate problem reports will be submitted to the CCITT asking for clarification.

7.14.12 OLD ACTIVITY IDENTIFIER

The Old Activity Identifier in S-ACTIVITY-RESUME refers to the original activity identifier.

7.14.13 NEGOTIATION DOWN TO TRANSPORT CLASS 0

For European implementations, X.410 specifies that class 0 transport must be supported. However, it is permissible for an initiator to propose a higher class as the preferred class, provided that class 0 appears as the alternate class in the T-Connect PDU. A responding implementation can choose to use either the preferred or alternate class, but again, must be able to use class 0. In other words, for private to private connections in Europe, class 0 transport is required.

This conflicts with the NIST agreements, since class 0 is only required if one of the partners in a connection is an ADMD.

7.15 APPENDIX E: WORLDWIDE X.400 CONFORMANCE PROFILE MATRIX

Y CONFORMANCE (E)
implies a conformance problem for European products in the United States.

Y CONFORMANCE (US)
implies a conformance problem for U.S. products in Europe.

- The A/311 profile is specified in Env 41 202, the A/3211 profile in Env 41 201
- No TTC protocol classification for RTS exists.
- The notation X/Y indicates "X" for PRMDs and "Y" for ADMDs, i.e., "M/G" would be Mandatory for PRMDs and Generatable for ADMDs.

Table 7E.1. Protocol element comparison of RTS

RTS element	NIST	A/311	A/3211	PROBLEM Y/N
PConnect	M	M	M	N
DataTransferSyntax	M 0	M 0	M 0	N
PUserData	M	M	M	N
checkpointSize	H	H	H	N
windowSize	H	H	H	N
dialogueMode	H	H	H	N
connectdata	M	M	M	N
applicationProtocol	G 1 H 8883	H 1	R 8883	N
ConnectionData				
Open	G	G	G	N
Recover	G	H	G	N
Open				
RTSUserData	G	G	G	N
Recover				
SessionConnectionID	G	G	G	N
RTSUserData				
MTAName	G	G	G	N
Password	G	G	G	N
null	G	G	G	N
SessionConnectionID				
CallingUserReference	M	M	M	N
CommonReference	M	M	M	N
AdditionalRefInfo	H	H	H	N
PAccept	G	G	G	N
DataTransferSyntax	M 0	M 0	M 0	N

(Continued on next page.)

Table 7E.1. Protocol element comparison of RTS, continued

RTS element	NIST	A/311	A/3211	PROBLEM (Y/N)
PUserData	M	M	M	N
CheckpointSize	H	H	H	N
WindowSize	H	H	H	N
ConnectionData	M	M	M	N
PRefuse	G	G	G	N
RefuseReason	M	M	M	N
SSUserData (in S-TOKEN-PLEASE)	G	G	G	N
AbortInformation (in S-U-ABORT)	G	G	G	N
AbortReason	H	H	H	N
reflectedParameter	X	X	X	N

Table 7E.2. Protocol element comparison of P1

P1 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
ORname					
StandardAttributeList	M	M	M	M	N See Note 4
DomainDefAttributeList	X	X	X	G	Y See Note 5
StandardAttributeList					
CountryName	R	R	R	M	N
		ISO R	R		N
		X.121 H	H		Y Conformance (E)
		Other X	X		Y Prot Vio
AdministrationDomainName	R	R	G	M	N
... if PrintableString		R	G		N
... if numericString		H	H		Y Conformance (E)
X.121 Address	X	X/R	X		Y Conf(US)See Note 1
Terminal ID	X	X/G	X		Y Conf(US)See Note 1
PrivateDomainName	G	G	G	G	N
OrganizationName	G	G	G	G	N
UniqueUAidentifier	X	X/G	X		Y Conf(US)See Note 1
PersonalName	G	G	G	G	N
OrganizationalUnit	G	G	G	G	N
DomainDefinedAttribute	X	X	X	G	N
Type	M	M	M	M	N
Value	M	M	M	M	N
PersonalName					
Surname	M	M	M	M	N
GivenName	G	G	G	G	N
Initials	G	G	G	G	N
GenerationQualifier	G	X	X	X	Y Conformance (E)
GlobalDomainIdentifier					
CountryName	M	M	M	M	N
AdministrationDomainName	M	M	G	M	Y Proto Vio
PrivateDomainIdentifier	R/H	H	R	M/X	N
MPDU					
UserMPDU	G	G	G	G	Y TTC required MPDU size is 32K
DeliveryReportMPDU	G	G	G	G	N
ProbeMPDU	H	H	H	H	N

Table 7E.2. Protocol element comparison of P1, continued

P1 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
UserMPDU					
UMPDUenvelope	M	M	M	M	N
UMPDUcontent	M	M	M	M	N
UMPDUenvelope					
MPDUidentifier	M	M	M	M	N
originatorORname	M	M	M	M	N
originalEncodedTypes	G	H	H	G	Y Conformance (E)
ContentType	M	M	M	M	N
UAcontentID	H	H	H	H	N
Priority	G	G	G	G	N
PerMessageFlag	G	G	G	G	N
DeferredDelivery	X	X	X	X	N
PerDomainBilatInfo	X	X	X	X	N
RecipientInfo	M	M	M	M	Y TTC MPDU 32K
TraceInformation	M	M	M	M	N
MOTIS-> LatestDelivery			X		N
MOTIS-> InternalTraceInfo	M/P		P		N
UMPDUcontent	M	M	M	M	N
MPDUidentifier					
GlobalDomainIdent	M	M	M	M	N
IA5string	M	M	M	M	N
PerMessageFlag					
DiscloseRecipients	H	G @ MTL	H	H	Y Conformance (US)
		H at UA	?		Y Conformance (US)
ConversionProhibited	G	G	G	G	N
AlternatRecipAllowed	H	G @ MTL	H	X	Y Conformance (US)
		H at UA	?		Y Conformance (US)
ContentReturnRequest	X	X	X	X	
MOTIS-> redirectionProhibited			X		N
PerDomainBilateralInfo					
CountryName	M	M	M	M	N
AdminDomainName	M	M	G	M	Y Prot Vio
MOTIS-> PrivateDomainName			G		N
BilateralInfo	M	M	M	M	N

(Continued on next page

Table 7E.2. Protocol element comparison of P1, continued

P1 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
DeliveryReportContent					
original MPDUident	M	M	M	M	N
intermediate Trace	X/G	X	X	X	Y Conformance (E)
UAcontentID	G	G	G	G	N
ReportedRecipientInfo	M	M	M	M	Y TTC 256 max
returned	H	H	X	X	Y Conformance (E)
billing information	X	X	X	X	N
ReportedRecipientInfo					
recipient ORname	M	M	M	M	N
extensionsIdentifier	M	M	M	M	N
PerRecipientFlag	M	M	M	M	N
LastTraceInformation	M	M	M	M	N
intendedRecipient	H	H	H	H	N
SupplementaryInfo	X/H	X	X	X	Y Conformance (E)
MOTIS-> ReassignmentInfo			X		N
MOTIS-> ReassignmentInfo					
MOTIS-> intendedRecipient			M		N
MOTIS-> reasonForReassignment			H		N
LastTraceInformation					
arrival	M	M	M	M	N
convertedEncInfoTypes	G	G	H	G	Y Conformance (E)
Report	M	M	M	M	N
Report					
DeliveredInfo	G	G	G	M	N See Note 6
NonDeliveredInfo	G	G	G	M	N
DeliveredInfo					
delivery	M	M	M	M	N
TypeofUA	R/H	H	R	M/G	N
NonDeliveredInfo					
ReasonCode	M	M	M	M	N
DiagnosticCode	H	H	H	H	N
MOTIS-> UAprofileIdentifier			X		N
MOTIS-> UAprofileIdentifier					
MOTIS-> ContentType			M		N
MOTIS-> EncodedInfoTypes			M		N

(Continued on next page)

Table 7E.2. Protocol element comparison of P1, continued

P1 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
ProbeEnvelope					
probe	M	M	M	M	N
originator	M	M	M	M	N
ContentType	M	M	M	M	N
UAcontentID	H	H	H	H	N
originalEncInfoTypes	G	H	H	G	Y Conformance (E)
TraceInformation	M	M	M	M	N
PerMessageFlag	G	G	G	G	N
ContentLength	H	H	H	H	N
PerDomainBilatInfo	X	X	X	X	N
RecipientInfo	M	M	M	M	Y TTC 256 max
MOTIS-> InternalTraceInfo	M/P		P		N
RecipientInfo					
RecipientORname	M	M	M	M	N
ExtensionIdentifier	M	M	M	M	N
PerRecipientFlag	M	M	M	M	N
ExplicitConversion	X	X	X	X	N
MOTIS-> OriginatorReqAlternatRecip			X		N
MOTIS-> ReassignmentInfo			X		N
PerRecipientFlag					
ResponsibilityFlag	M	M	M	M	N
ReportRequest	M	M	M	M	N
UserReportRequest	M	M	M	M	N
TraceInformation					
GlobalDomainIdent	M	M	M	M	N
DomainSuppliedInfo	M	M	M	M	N

(Continued on next page

Table 7E.2. Protocol element comparison of P1, continued

P1 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
DomainSuppliedInfo					
arrival	M	M	M	M	N
deferred	X	X	X	X	N
action	M	M	M	M	N
(0=relayed)	G	G	G		N Note: Re-routing not required.
(1=rerouted)	H	H	H		N
MOTIS-> (2=recipientReassigned)			H		N
converted	H	G	H	H	Y Conformance(US)
previous	H	G	G	X	Y Conformance(US) (Note: G is inconsistent with action (relayed) being "H.")
ORname					
EncodedInformationTypes					
BitString	M	M	M	M	N See Note 3
G3NonBasicParameters	X	X	X	X	N
TeletexNonBasicParams	X	R	X	X	Y Conformance(US)
PresentationAbilities	X	X	X	X	N
DeliveryReportMPDU	G	G	M	G	N
DeliveryReportEnvelop	M	M	M	M	N
DeliveryReportContent	M	M	M	M	N
DeliveryReportEnvelope					
report	M	M	M	M	N
originator ORname	M	M	M	M	N
TraceInformation	M	M	M	M	N
InternalTraceInfo	M/P		P		N

(Continued on next page)

Table 7E.3. Protocol element comparison of P2

P2 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
UAPDU					
IM_UAPDU	G	G	G	G	N
SR_UAPDU	X	X	X	X	N
IM_UAPDU					
Heading	M	M	M	M	N
Body	M	M	M	M	N
Heading					
IPmessageID	M	M	M	M	N
Originator ORname	R	R	R	M/G	N
AuthorizingUsers	H	H	H	H	Y TTC 16 max
PrimaryRecipients	G	G	G	G	Y TTC 256 max
CopyRecipients	G	G	G	G	Y TTC 256 max
BlindCopyRecipients	H	H	H	H	Y TTC 256 max
InReplyTo	G	G	G	G	N
Obsoletes	H	H	H	H	Y TTC 8 max
CrossReferences	H	H	H	H	Y TTC 8 max
Subject	G	G	G	G	N
ExpiryDate	H	H	H	H	N
ReplyBy	H	H	H	H	N
ReplyToUsers	H	H	H	H	Y TTC 32 max
Importance	H	H	H	H	N
Sensitivity	H	H	H	H	N
Autoforwarded	H	H	H	H	N
MOTIS-> CirculationList			X		N
MOTIS-> ObsoletingTime			X		N
IPmessageID					
ORname	H	H	H	H	N
PrintableString	M	M	M	M	N
ORdescriptor					
ORname	H	H	H	┐ ├M	N See Note 6
FreeFormName	H	H	H	┘	N
TelephoneNumber	H	H	H	G	N
Recipient					
ORdescriptor	M	M	M	M	N
ReportRequest	X	X	X	X	N
ReplyRequest	H	H	H	H	N

(Continued on next page

Table 7E.3. Protocol element comparison of P2, continued

P2 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
MOTIS-> CirculationList					
MOTIS-> CirculationMember			X		N
MOTIS-> checkmark			M		N
MOTIS-> membername			M		N
MOTIS-> OBsoletingTime					
MOTIS-> Time			H		N
MOTIS-> IP_MessageID			H		N
Body					
BodyPart	G	M	M	G	Y Conformance (US)
SR_UAPDU					
NonReceipt	H	H	H	┐ ├M	N
Receipt	H	H	H	┘	N
Reported	M	M	M	M	N
ActualRecipient	R	R	R	G	N
IntendedRecipient	H	H	H	H	N
Converted	X	X	X	G	N
MOTIS-> CirculationStatus			X		N
NonReceiptInformation					
Reason	M	M	M	M	N
NonReceiptQualifier	H	H	H	H	N
=expired (value)	0	0	0	0	N
=obsoleted (value)	1	1	1	1	N
=subscriptionTerminated	2	2	2	2	N
MOTIS-> =timeobsoleted (value)			X		N
Comments	H	H	H	X	N
returned	H	X	X	X	Y Conformance (E)
ReceiptInformation					
Receipt	M	M	M	M	N
TypeOfReceipt	H	H	H	G	N
SupplementaryInfo	X	X	X	X	N

(Continued on next page

Table 7E.3. Protocol element comparison of P2, continued

P2 Protocol	NIST	A/311	A/3211	TTC	PROBLEM (Y/N)
BODYPART SUPPORT					
o IA5 Text	G	G	G		N See Note 7
o TLX	X	X	X		N
o Voice	X	X	X		N
o G3FAX	X	X	X		N
o TIFO	X	X	X		N
o TTX	X	X/H	X		Y Conf(US)See Note 2
o VideoTex	X	X	X		N
o NationallyDefined	X	X	X		N
o Encrypted	X	X	X		N
o ForwardedIPmessage	H	H	H		N
o SFD	X	X	X		N
o TIFI	X	X	X		N
MOTIS-> o ODA			X		N
MOTIS-> o ISO6937 Text			H		N

Note 1: It should be noted that the A/311 profile states: For routing all ADMDs should support all Form 1 Variants of O/R Name. All PRMDs should support at least Form 1, Variant 1 form of OR Name.

Note 2: It should also be noted that the A/311 profile requires that all ADMDs should support the reception of Teletex body parts for delivery to their own UAEs.

Note 3: An A/3211 implementation may generate MOTIS encoded information types. See 7.6.11.

Note 4: Only Form 1 Variant 1 of O/Rname shown for TTC, but TTC defines other forms and variants. Form 1 Variant 1 recommended for PRMDs and ADMDs, Form 1 Variant 2 also recommended for ADMDs.

Note 5: DDA's can be used to specify recipients in any Japanese domains other than TTC. Assignment of DDAs for UAs within TTC domains is not recommended.

Note 6: One of [DeliveredInfo/NonDeliveredInfo] must be present. TTC encodes this as shown. Other profiles represent this by classifying both protocol elements as generatable. A similar situation exists with the P2 ORdescriptor.

Note 7: TTC is expected to support IA5 for some international MHS communications.

7.16 APPENDIX F: INTERWORKING WARNINGS

ADMD name is to be encoded as a single space when configurations with no ADMD's are present. It should be noted that this may change in January 1988 so that the ADMD name is encoded as a zero length element in such cases.

The NIST agreements allow implementation to generate MPDUs with no body parts. Such MPDUs will be rejected by European-conformant systems. (Note this situation may change in January 1988)

In order to optimize the number of recipients you can read and reply to, it is advisable to be able to generate all standard O/R name attributes.

8. MESSAGE HANDLING SYSTEMS

8.1 INTRODUCTION

This is an Implementation Agreement developed by the Implementor's Workshop sponsored by the National Institute of Standards and Technology to promote the useful exchange of data between devices manufactured by different vendors. This Agreement is based on, and employs protocols developed in accord with, the OSI Reference Model. It provides detailed guidance for the implementor and eliminates ambiguities in interpretations.

This is an Implementation Agreement for Message Handling Systems (MHS) based on both the CCITT X.400(1988) series of Recommendations and the similar (but not identical) ISO MOTIS standard (see References). The term 'MHS' is used to refer to both sources where a distinction is unnecessary. Similarly, '1984' and '1988' are often used to distinguish between the CCITT X.400(1984) series of Recommendations and the later sources.

This Implementation Agreement seeks to establish a common specification which is conformant with both CCITT and ISO with a view to:

- Preventing a proliferation of incompatible communities of MHS systems which are isolated for protocol reasons,
- Achieving interworking with implementations conforming to the NIST Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems, and
- Facilitating integration of other OSI-based services (e.g., Directory) within a single real system.

This initial Implementation Agreement is designed to encourage early upgrade of existing 1984-based systems as follows:

- To add 1988 functionality (Message Store, remote User Agent, etc), and
- To provide a minimal conformant 1988 MHS as a firm basis for the introduction of further 1988 services and features. Subsequent versions of this Agreement will define such additional 1988 aspects as incremental enhancements.

However, it is considered that the NIST Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems (ch. 7) should not be withdrawn at this stage. It is anticipated that X.400(1984) implementations will continue to provide a viable

alternative for applications that do not require the additional 1988 functionality for some time.

8.2 SCOPE

This Agreement specifies the requirements for MHS implementations based on the 1988 MHS standards.

This Agreement applies equally to Private Management Domains (PRMDs) and Administration Management Domains (ADMDs). Four boundary interfaces are specified, as illustrated in figure 8.1:

- Management Domain (MD) to MD,
- Message Transfer Agent (MTA) to MTA within a domain,
- MTA to remote Message Store (MS) or User Agent (UA), and
- MS to remote UA.

MHS protocols other than the Message Transfer Protocol (P1), the Message Transfer System Access Protocol (P3), the Interpersonal Messaging Protocol (P2), and the Message Store Access Protocol (P7) are beyond the scope of this Agreement. Issues arising from the use of other protocols are outside the scope of this document. This Agreement describes the services provided at each interface shown in figure 8.1.

MHS implementations may be configured as any single or multiple occurrence or combination of MTA, MS and UA, as illustrated in figure 8.1. It is not intended to restrict the types of system that may be configured for conformance to this Agreement (although it is equally recognized that not all configuration types may be commercially viable).

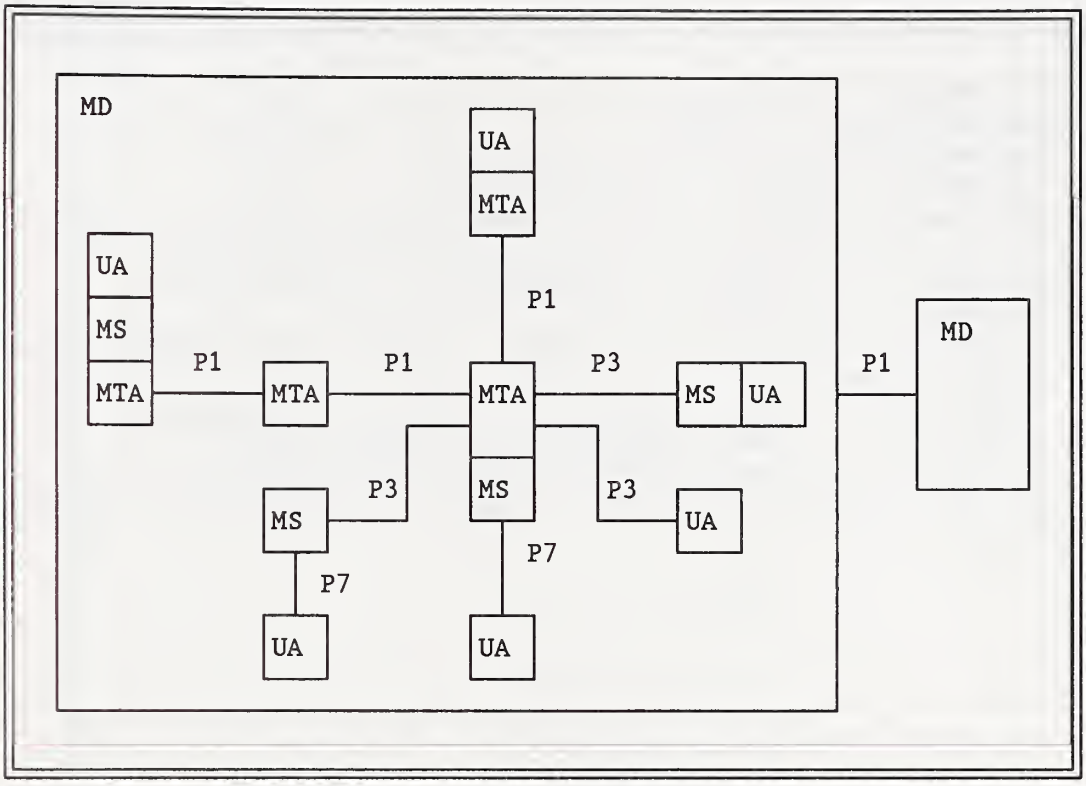


Figure 8.1. Scenario Definition.

The 1988 MHS standards cover a wide and diverse range of functional areas, not all of which would be relevant to every implementation. In order to achieve a more precise definition of conformance requirements according to the functionality supported by an implementation, and additionally to facilitate future enhancement of this initial specification, the concept of 'Functional Groups' has been introduced. Figure 8.2 shows the Functional Groups covered by this Agreement and indicates where they are defined in this chapter. Conformance requirements for support of Functional Groups by particular configurations are specified in section 8.16.

In the context of these agreements, the term "Support" means that the service provider makes the element of service (and related elements of protocol) available to the service user. The service user provides adequate access to invoke the elements of service and/or makes information associated with the service element available. Additionally, for "Not Defined" or "Not Applicable" elements, the service provider is not required to make the element available to the service user. However, the service provider should not regard the occurrence of the corresponding protocol elements as an error and should relay those elements. Naturally, protocol elements marked critical for transfer must be processed per-X.400 ~~series~~ according to the base standards.

The following functional groups are covered by this Implementors Agreement:

a) The MT Kernel in clause 5;
b) The IPM Kernel in clause 6;
c) The Message Store in clause 7;
d) Remote User Agent support in clause 8;
e) Distribution Lists in clause 9.2 (which are for further study);
f) Use of Directory in clause 9.3;
g) MHS Management in clause 10 (which is for further study);
h) Security in clause 11;
i) The Physical Delivery Access Unit in clause 12.1 (which is for further study);
j) Other Access Units in clause 12.2 (which are for further study);
k) Conversion in clause 13 (which is for further study);
l) Redirection in clause 14 (which is for further study); and,
m) The EDI Messaging Service in clause 15 (which is for further study).

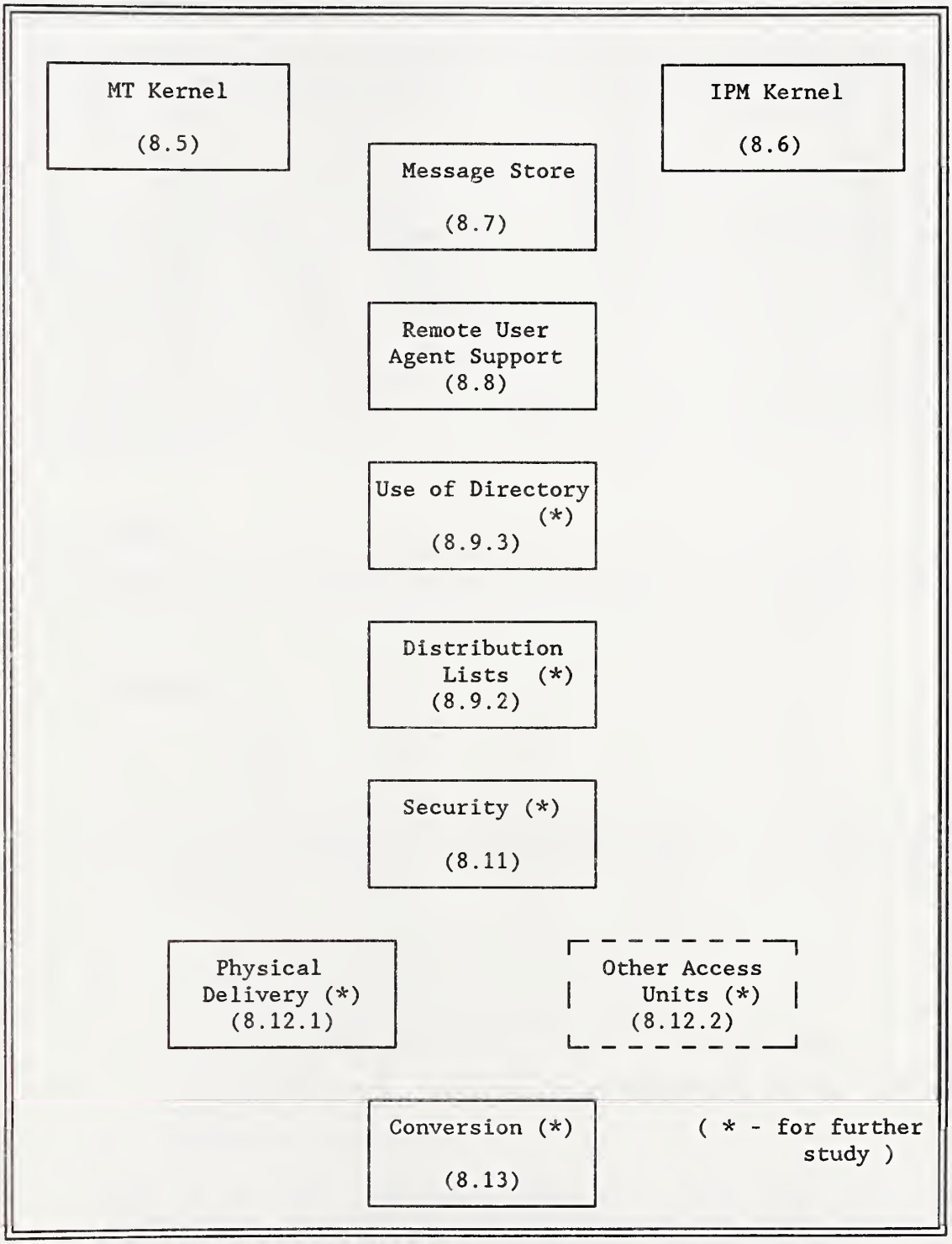


Figure 8.2. MHS Functional Groups.

8.3 STATUS

This version of the Implementation Agreements for Message Handling Systems (MHS) is under development. It is based on the CCITT X.400(1988) Recommendations and ISO MOTIS (10021, parts 1-7) standards.

It is intended that the Stable Implementation Agreements will initially include an Agreement which specifies a minimal 1988-based MHS implementation and support for Message Stores and remote User Agents, and which addresses interworking with 1984-based implementations. The remaining features specified in the 1988 standards will be covered in subsequent versions of this Agreement.

This initial version has not yet been aligned with other MHS profiles, so changes may be necesary in the future for international harmonization. (e.g., support for international character repertoires and conversion.)

8.4 ERRATA

~~No Errata to Stable material at this time.~~ Alignment Errata to tables in sections 8.17.1 to 8.17.3 have been included as indicated.

8.5 MT KERNEL

8.5.1 Introduction

This section specifies the requirements for a minimal 1988-based MTS implementation (i.e., MTA) which is capable of interworking with 1984-based MTAs. The 'base' MT Service specified in this section does not include:

- Message Store (see 8.7)
- Remote UA (see 8.8)
- Use of Directory Services (see 8.9.3)
- Distribution Lists (see 8.9.2)
- Security (see 8.11)
- Interworking with Physical Delivery systems or Specialized Access (see 8.12)
- Conversion (see 8.13)

Such a minimal 1988-based MTA will have the following capabilities in order to achieve interworking with 1984-based MTAs and to facilitate migration to full 1988 operation:

- It will be protocol-conformant to 1988 P1;

- o It will downgrade 1988 P1 to 1984 P1 when relaying to 1984-based MTAs, as specified in Annex B of X.419 (see 8.5.5);

- o It will support both 'normal' mode and 'X.410-1984' ('passthrough') mode protocol stacks (i.e., as required by ISO and CCITT respectively).

8.5.2 Elements of Service

This section specifies the requirements for support of MT Elements of Service by an MTA conforming to the MT Kernel Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as follows:

Mandatory (M) - the Element of Service must be supported and made available to the service user;

Optional (O) - the Element of Service may be supported, but is not required for conformance to this Agreement;

Not Defined/Not Applicable (-) - the Element of Service is not defined by this Agreement or is otherwise not applicable in the particular context;

To Be Determined (*) - the support classification for the Element of Service has yet to be determined.

The requirements for support of MT Elements of Service for origination and reception and (where relevant) relaying are distinguished. Elements of Service which are new in the 1988 MHS standards are indicated as (1988).

An MTA must support those Basic MT Elements of Service and MT Optional User Facilities defined in clause 19 of X.400(1988) as listed and qualified in tables 8.1 and 8.2 below.

Table 8.1. MT Kernel: Basic MT Elements of Service

Element of Service	Origination	Reception	Relaying
Access Management	M[1]	M[1]	-
Content Type Indication	M	M	-
Converted Indication	M	M	M
Delivery Time Stamp Indication	-	M	-
Message Identification	M	M	-
Non-delivery Notification	M	M	M
Original Encoded Information Types Indication	M	M	-
Submission Time Stamp Indication	M	M	-
User/UA Capabilities Registration (1988)	-	M[1]	-

Notes:
1) A local matter in the case of co-located UA/MTA and/or MS/MTA configurations.

Table 8.2. MT Kernel: MT Service Optional User Facilities

Element of Service	Origination	Reception	Relaying
Alternate Recipient Allowed	M	M[2]	-
Alternate Recipient Assignment	-	O[2]	-
Conversion Prohibition	M	M	M
Conversion Prohibition in Case of Loss of Information (1988)	O	O	O
Deferred Delivery	M[3]	O	O
Deferred Delivery Cancellation	M[6]	-	-
Delivery Notification	M	M	-
Disclosure of Other Recipients	M	M	M
DL Expansion History Indication	-	M[4]	-
DL Expansion Prohibited	M[5]	-	-
Explicit Conversion	O	O	O
Grade of Delivery Selection	M	M	M
Hold for Delivery	-	M[1]	-
Implicit Conversion	O	O	O
Latest Delivery Designation (1988	O	O	O
Multi Destination Delivery	M	M	M
Originator Requested Alternate Recipient (1988)	O	O	-
Prevention of Non-delivery Notification	O	-	-
Probe	M	M	M
Redirection Disallowed by Originator (1988)	O	O	-
Redirection of Incoming Messages (1988)	-	O	-
Requested Delivery Method (1988)	M	M	-
Restricted Delivery (1988)	-	O	-
Return of Content	O	O	O

Notes:

1) A local matter in the case of co-located UA/MTA and/or MS/MTA configurations.

2) If Alternate Recipient Assignment is supported on reception, then support of Alternate Recipient Allowed is Mandatory on reception; otherwise, support of Alternate Recipient Allowed is not applicable on reception.

3) Support of this MT Element of Service is Mandatory for conformance reasons, but may be performed as a local matter to the originating MTA.

Table 8.2. MT Kernel: MT Service Optional User Facilities (Cont.)

4) Support of this MT Element of Service refers only to the delivery of DL expansion history and not to the performing of DL expansion (see 8.9.2). 5) Support of this MT Element of Service does not imply the capability to perform DL expansion (see 8.9.3). 6) Messages should be held in the originating MTA to provide support for this element of service.

8.5.3 MTS Transfer Protocol (P1)

The requirements for support of MTS Transfer Protocol (P1) elements are detailed in section 8.17.1 (Appendix A).

Support of MTS Transfer Protocol application contexts by an MTA is classified as in table 8.3.

Table 8.3. Application Contexts Classification

Application Context	Support
mts-transfer-protocol-1984	Mandatory
mts-transfer-protocol	Mandatory
mts-transfer	Mandatory

Use of the underlying services to support these application contexts is specified in section 8.14.

8.5.4 MTS - APDU Size

See Working Document.

8.5.5 1988/84 Interworking Considerations

See Working Document.

8.6 IPM KERNEL

8.6.1 Introduction

This section specifies the requirements for a minimal 1988-based IPMS implementation (i.e., UA) which is capable of interworking with 1984-based UAs. The 'base' IPM Service specified in this section does not include:

- Message Store (see 8.7)
- Remote UA (see 8.8)
- Use of Directory Services (see 8.9.3)
- Distribution Lists (see 8.9.2)
- Security (see 8.11)
- Interworking with Physical Delivery systems or Specialized Access (see 8.12)

Such a minimal 1988-based UA will have the following capabilities in order to achieve interworking with 1984-based UAs and to facilitate migration to full 1988 operation:

- It will continue to support content type P2 (encoded as integer 2) on origination and reception;

- It will support receipt of P2 (encoded as integer 22);

- It may originate P2 encoded as integer 22, but the guidelines specified in clause 20.2 of X.420(1988) are to be followed, i.e., the content type shall be encoded as integer 2 unless 1988 P2 protocol elements are present.

All IPM UAs must support either MTS Submission and Delivery based on the protocol classifications in section 8.17.3, or MS Submission and Retrieval based on the protocol classifications in section 8.17.4. However, how such information is conveyed to/from the MTS or MS in the case of a co-located UA is a local matter, and will not necessarily be subject to conformance verification.

8.6.2 Elements of Service

This section specifies the requirements for support of IPM Elements of Service by a UA conforming to the IPM Kernel Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in section 8.5.2.

The requirements for support of IPM Elements of Service for origination and reception are distinguished. Elements of Service which are new in the 1988 MHS standards are indicated as (1988).

A UA must support those Basic IPM Elements of Service and IPM Optional User Facilities defined in Clause 19 of X.400(1988) as listed and qualified in tables 8.4 and 8.5 below.

Table 8.4. IPM Kernel: Basic IPM Elements of Service

Element of Service	Orig	Recep
Access Management	M[1]	M[1]
Content Type Indication	M	M
Converted Indication	-	M
Delivery Time Stamp Indication	-	M
IP-message Identification	M	M
Message Identification	M	M
Non-delivery Notification	M	-
Original Encoded Information Types Indication	M	M
Submission Time Stamp Indication	M	M
Typed Body	M	M
User/UA Capabilities Registration (1988)	-	M[1]

Notes:
1) In the case of a co-located UA/MTA or co-located UA/MS, the method and extent to which this Element of Service is provided is a local matter; it is not necessarily testable in the absence of support for the P3 or P7 protocol.

Table 8.5. IPM Kernel: IPM Service Optional User Facilities

Element of Service	Orig	Recep
Alternate Recipient Allowed	O	-
Alternate Recipient Assignment	-	O
Authorizing Users Indication	O	M
Auto-forwarded Indication	O	M
Blind Copy Recipient Indication	O	M
Body Part Encryption Indication	O	M
Conversion Prohibition	M	M
Conversion Prohibition in Case of Loss of Information (1988)	O	O
Cross Referencing Indication	O	M
Deferred Delivery	M	-
Deferred Delivery Cancellation	O	-
Delivery Notification	M	-
Disclosure of Other Recipients	O	M
DL Expansion History Indication (1988)	-	M
DL Expansion Prohibited (1988)	M	-
Expiry Date Indication	O	M
Explicit Conversion	O	-
Forwarded IP-message Indication	O	M
Grade of Delivery Selection	M	M
Hold for Delivery	-	O
Implicit Conversion	-	O
Importance Indication	O	M
Incomplete Copy Indication (1988)	O	O
Language Indication (1988)	O	M
Latest Delivery Designation (1988)	O	-
Multi-Destination Delivery	M	-
Multi-part Body	O	M
Non-receipt Notification Request	O	M[1]
Obsoleting Indication	O	M
Originator Indication	M	M
Originator Requested Alternate Recipient (88)	O	-
Prevention of Non-delivery Notification	O	-
Primary and Copy Recipients Indication	M	M
Probe	O	-
Receipt Notification Request Indication	O	O
Redirection Disallowed by Originator (1988)	O	-
Redirection of Incoming Messages (1988)	-	O
Reply Request Indication	O	M
Replying IP-message Indication	M	M
Requested Delivery Method (1988)	M	-
Restricted Delivery (1988)	-	O
Return of Content	O	-
Sensitivity Indication	O	M
Subject Indication	M	M
Use of Distribution List (1988)	O	-

Table 8.5. IPM Kernel: IPM Service Optional User Facilities (Cont.)

Notes:

1) Support of Non-Receipt Notification Request on reception does not require the capability to generate a non-receipt notification in the case of an implementation in which a non-receipt condition cannot occur.

8.6.3 Interpersonal Messaging Protocol (P2)

The requirements for support of Interpersonal Messaging Protocol (P2) elements are detailed in section 8.17.2 (Appendix A).

8.6.4 Body Part Support

This section specifies the requirements for support of IPM body part types by a UA conforming to this Agreement.

The classification scheme for support of IPM body part types is as defined in section 8.5.2.

The requirements for support of IPM body part types for origination and reception are distinguished. Body part types which are new in the 1988 MHS standards are indicated as (1988).

A UA must support those IPM body part types defined in Annex E of X.420(1988) as listed and qualified in table 8.6 below. If an implementation supports a particular body part type for reception, it should also be able to support that body part type for reception if it is part of a forwarded message.

Any basic body part type that is supported on reception must be supported as integer encoding (ASN.1 context-specific identifier) and as object identifier (externally-defined) encoding.

All body parts with integer-encoded identifiers in the range 0 up to and including 16K-1 are legal. Body part integer-encoded identifiers corresponding to X.121 country codes should be interpreted as described in figure 8.5. These privately-defined body part types are specified as an interim measure to provide backward compatibility with 1984 MHS implementations. For interworking between UAs based on the 1988 (or later) MHS standards, it is strongly recommended that the externally-defined body part be used instead.

Table 8.6. IPM Kernel: Body Part Types

Body Part Type		Orig	Recep
IA5Text		M	M
Voice		O	O
G3Facsimile		O	O
G4Class1	(TIF0)	O	O
Teletex		O	O
Videotex		O	O
Encrypted		O	O
Message	(ForwardedIPMessage)	O	M
MixedMode	(TIF1)	O	O
BilaterallyDefined	(Unidentified)	O	O
NationallyDefined		O	O
ExternallyDefined	(1988)	O	O/M[1]
PrivatelyDefined	(see figure 8.5)	O	O
GeneralText	(1988 - extended)	*	*

Notes:

1) Any basic body part type that is supported on reception as integer encoding must also be supported as object identifier encoding. Support for all other externally defined body parts is optional.

```
BodyPart              ::=  CHOICE {
  ia5-text                   [0] IA5TextBodyPart,
                             .
  externally-defined         [15] ExternallyDefinedBodyPart,
                             .
                             [234] UKBodyParts,
                             .
                             [310] USABodyParts,
                             .        }

Where UKBodyParts and USABodyParts (privately defined) are
defined as:

                             SEQUENCE {BodyPartNumber, ANY}
BodyPartNumber        ::=  INTEGER

These privately-defined body part types are specified as an
interim measure to provide backward compatibility with 1984
MHS implementations.  For interworking between UAs based on
the 1988 (or later) MHS standards, it is strongly
recommended that the externally-defined body part be used
instead.

The undefined bit in P1 EncodedInformationTypes must be set
when a message contains a privately defined body part. Each
UA that expects such body parts should include undefined in
the set of deliverable EncodedInformationTypes it registers
with the MTA.

Body part numbers are interpreted relative to the body part
type in which they are used.  NIST registers body part
numbers for privately-defined formats within the United
States.
```

Figure 8.5. Privately-Defined Body Parts.

8.7 MESSAGE STORE

8.7.1 Introduction

This section specifies Agreements for implementation of the Message Store (MS) Functional Group. The MS is responsible for accepting delivery of messages on behalf of a single end-user, and retaining the messages until the end-user's UA is able to retrieve them. Message submission and some administration services are provided via "pass-through" to the MTS. Figure 8.6 illustrates the logical relationship of the MS to the UA and MTS.

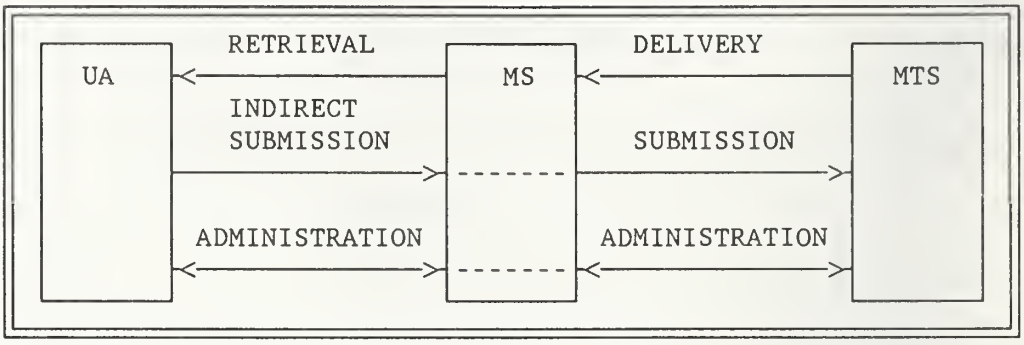


Figure 8.6. Message Store Model.

The Agreements in this section specify the Message Store's use of the retrieval, delivery, and administration services. Agreements on submission services are specified in section 8.8, which describes support for the remote UA.

The goal of the Agreements in this section is to define the minimal set of features which are necessary to provide useful Message Store services, independent of the MTA implementation version (i.e., 1984 or 1988).

8.7.2 Scope

The scope of the Agreements in this section is depicted in figure 8.7 below, and is confined to the services and protocols between the boundaries shown (marked with asterisks). Requirements for the UA and MTA are addressed only to the extent that they affect the Message Store and remote User Agent services and protocols. This reflects the additional services required at the UA to support MS access and at the MTA to support a remote MS.

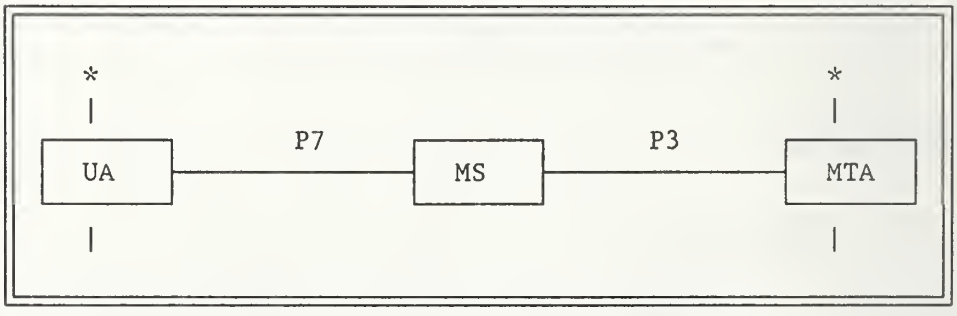


Figure 8.7. Scope of Message Store Agreements.

The UA, MS and MTA configuration is not restricted; any of these components may be co-located, although they are depicted as logically separate. In the case of a co-located UA and MS, a proprietary interface may be used instead of P7. In the case of

a co-located MS and MTA, a proprietary interface may be used instead of P3.

8.7.3 Elements of Service

This section specifies the requirements for support of Elements of Service to provide a Message Store conforming to the Message Store Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in section 8.5.2.

Support for Elements of Service is specified both for the Message Store itself and for the User Agent.

Table 8.7. Message Store: Elements of Service

Element of Service	UA	MS
Stored Message Deletion	M	M
Stored Message Fetching	M	M
Stored Message Listing	M	M
Stored Message Summary	M	M
Stored Message Alert	O	O
Stored Message Auto Forward	O	O

8.7.4 Attribute Types

Requirements for support of the attributes used in the Message Store are detailed in sections 8.17.5 and 8.17.6 (Appendix A). section 8.17.5 specifies support for the General Attributes of the Message Store, while section 8.17.6 specifies support for the IPM Message Store Attributes.

There are two classes of support for General Attributes in the Message Store: Basic and IPM.

The Basic MS is intended to support the use of the MS as a continuously available, reliable device (such as a spooling entity) for receiving, storing, and forwarding messages and reports. The Basic MS is not required to support any IPM attributes.

The IPM MS provides more flexible access to the General Attributes as well as supporting IPM Attributes.

IPM User Agents can make use of either the Basic or IPM MS. section 8.17.6 is to be read in accordance with Annex C of X.420 (1988).

8.7.5 Pragmatic Constraints for Attribute Types

There are no additional pragmatic constraints for attribute types beyond those of the basic standards.

8.7.6 Implementation of the MS with 1984 Systems

While the Message Store is part of the 1988 MHS standards, implementation of MS services with a 1984 MTA is possible. In order to interoperate with other 1984 MHS systems, implementations with this configuration should adhere to the following guidelines:

- The UA must generate 1984 P2 PDUs;
- The UA must identify the content protocol as integer 2 to the MS;
- The MS must be co-located with the MTA unless 1988 P3 support is provided on the 1984 MTA as well.

To meet these guidelines, the UA may be implemented as follows:

- The UA could conform to X.420(1984), with 1988 UA extensions for utilizing the MS services;
- The UA could be a 1988 UA with restrictions on protocol elements generated and by identifying the content type as integer 2 rather than 22. No 1988-specific elements should be generated.

Details of the interface between the 1988 MS and the 1984 MTA when co-located are beyond the scope of these Agreements.

8.7.7 MS Access Protocol (P7)

The requirements for support of MS Access Protocol (P7) elements by an MS and a remote MS-user are detailed in section 8.17.4 (Appendix A).

The requirements for support of MS Access Protocol (P7) application contexts by an MS and an MS-user are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the

additional requirement that an MS-user must at least support the ms-access application context, as defined in table 8.8.

Table 8.8. Application Contexts Support for P7

Application Context	MS	MS-user
ms-access	Mandatory	Mandatory
ms-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in section 8.14.

8.7.8 MTS Access Protocol (P3)

The requirements for support of MTS Access Protocol (P3) elements by an MTA and an MS where the MS is not co-located with the MTA are detailed in section 8.17.3 (Appendix A).

The requirements for support of MTS Access Protocol (P3) application contexts by an MTA and an MS in such a scenario are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the additional requirement that a remote MS must at least support the mts-access and mts-forced-access application contexts, as defined in table 8.9.

Table 8.9. Application Contexts Support for P3

Application Context	MTA	MS
mts-access	Mandatory	Mandatory
mts-forced-access	Mandatory	Mandatory
mts-reliable-access	Optional	Optional
mts-forced-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in section 8.14.

8.8 REMOTE USER AGENT SUPPORT

8.8.1 Introduction

This section specifies Agreements for implementation of the Remote User Agent Functional Group, i.e., for support of an IPM UA that is not co-located with its MTA. Support of other classes of UA is for further study.

The goal of the Agreements in this section is to define the minimal set of features which are necessary to provide useful remote User Agent services, independent of the MTA implementation version (i.e., 1984 or 1988).

8.8.2 Scope

The scope of the Agreements in this section is depicted in figure 8.8, and is confined to the services and protocols between the boundaries shown (marked with asterisks). Requirements for the UA and MTA are addressed only to the extent that they affect the remote User Agent services and protocols. Access to a Message Store by a remote User Agent is covered in section 8.7.

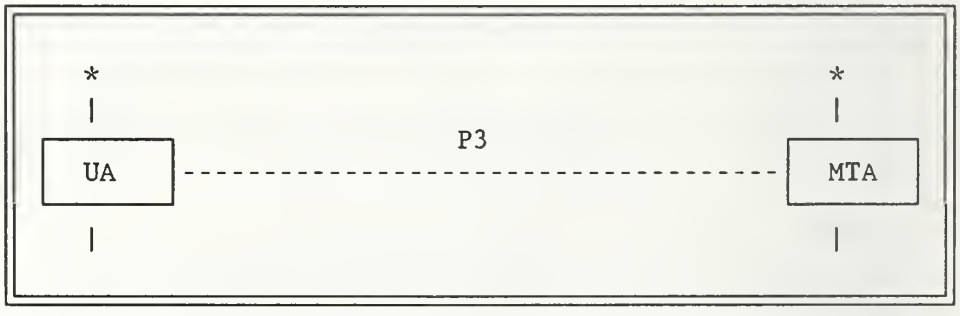


Figure 8.8. Scope of Remote User Agent Agreements.

8.8.3 Elements of Service

This section specifies the requirements for support of Elements of Service for conformance to the Remote User Agent Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in section 8.5.2.

Support for Elements of Service is specified both for the MT Service and for the IPM Service, and is in addition to the support requirements specified in sections 8.5 and 8.6 if this Functional Group is supported.

Table 8.10. Remote User Agent Support: MT Elements of Service

Element of Service	Origination	Reception
Access Management	M	M
Hold for Delivery	-	M
User/UA Capabilities Registration	-	M

Table 8.11. Remote User Agent Support: IPM Elements of Service

Element of Service	Origination	Reception
Access Management	M	M
Hold for Delivery	-	M
User/UA Capabilities Registration	-	M

8.8.4 MTS Access Protocol (P3)

The requirements for support of MTS Access Protocol (P3) elements by an MTA and an MTS-user (whether UA or UA/MS) where the MTS-user is not co-located with the MTA are detailed in section 8.17.3 (Appendix A).

The requirements for support of MTS Access Protocol (P3) application contexts by an MTA and an MTS-user in such a scenario are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the additional requirement that a remote MTS-user must at least support the mts-access and mts-forced-access application contexts, as defined in table 8.12.

Table 8.12. Application Contexts Support for P3

Application Context	MTA	MTS-user
mts-access	Mandatory	Mandatory
mts-forced-access	Mandatory	Mandatory
mts-reliable-access	Optional	Optional
mts-forced-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in section 8.14.

8.9 NAMING, ADDRESSING & ROUTING

See Working Document.

8.10 MHS MANAGEMENT

For further study.

8.11 MHS SECURITY

See Working Document.

8.12 SPECIALIZED ACCESS

See Working Document.

8.13 CONVERSION

See Working Document.

8.14 USE OF UNDERLYING LAYERS

8.14.1 MTS Transfer Protocol (P1)

The P1 protocol is mapped onto the Reliable Transfer Service Element (RTSE) either in X.410-1984 mode or in normal mode, as specified in section 8.5.3. In X.410-1984 mode, the RTSE makes direct use of the services provided by the Session Layer, as specified in chapter 5 (Upper Layers) of the Stable Implementation Agreements. In normal mode, the RTSE makes use of the services provided by the Association Control Service Element (ACSE) and Presentation Layer, as defined in chapter 5 (Upper Layers) of these Agreements.

8.14.2 MTS Access Protocol (P3) and MS Access Protocol (P7)

The P3 and P7 protocols make use of the services provided by the Remote Operations Service Element (ROSE), Association Control Service Element (ACSE), Presentation Layer, and, optionally, the Reliable Transfer Service Element (RTSE), as defined in chapter 5 (Upper Layers) of these Agreements. It is recommended that RTSE be used for recovery purposes when the implementation uses a Transport Class other than 4.

8.15 ERROR HANDLING

Editor's Note: The material in 8.15 was not explicitly included in a proposal brought to the OIW Plenary in December 1989.

This section describes appropriate actions to be taken upon receipt of protocol elements which are not supported in this profile: malformed PDUs, unrecognized O/R Name forms, content errors, errors in reports, and unexpected values for protocol elements.

An implementation must be able to report all error conditions which may occur with the appropriate error information as defined in the referenced base standards. An implementation must be able to handle receipt of all error indications which are defined in the referenced base standards. An implementation must also be tolerant of any additional error indications which are not currently defined, but is not required to be able to interpret such error information.

8.16 CONFORMANCE

For this section, the term conformance is as defined in ISO 9646.

Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this Agreement requires the ability to exchange messages without use of bilateral agreements.

In order to achieve a more precise definition of conformance requirements according to the functionality supported by an implementation, the concept of Functional Groups has been introduced. A Functional Group is a set of related Elements of Service and associated protocol elements which provide a discrete area of functionality.

Conformance to this Agreement requires as a minimum that all Mandatory Elements of Service listed in this chapter are supported in the manner defined in the MHS standards, as qualified in this Agreement, for each of the Functional Groups claimed. Any Optional Elements of Service for which support is claimed must also be supported as defined in the MHS standards and as qualified in this Agreement. Pragmatic constraints shall be observed as specified in the CCITT X.400 (1988) Series of Recommendations. It is not necessary to implement the recommended practices of Appendix C in order to claim conformance to this Agreement.

Conformance requirements for support of Functional Groups by particular configuration types (see sec. 8.2) are listed below. An

implementation may claim conformance to multiple configuration types (e.g., "MTA+UA" and "Class B MTA only").

Table 8.23. Conformance Requirements

Configuration[3]	Functional Groups					
	MT Kernel	IPM Kernel	MS[4]	Remote UA	DL	Directory
MTA + UA[2]	M[2]	M	-	O	O	O
MTA + MS	M	-	M	O	O	O
MTA only[1]: class A	M	-	-	-	O	O
class B	M	-	-	M	O	O
class C	M	-	-	-	O	O
MS + UA	-	M	M	-	-	O
MS only	-	-	M	-	-	*
UA only: P7	-	M	M	-	-	O
P3	-	M	-	M	-	O

Notes:

1) There are three conformance levels defined for the MT Kernel in this Agreement:

- o A class 'A' MT Kernel implementation conveys a message, probe, or report to another MT Kernel using standard means. A class 'A' MT Kernel is specifically implemented in order to transfer messages, probes, and reports which have previously been transferred and need not support submission and delivery. A class 'A' MT Kernel may perform other activities such as originate reports, expand distribution lists, and perform conversions.

- o A class 'B' MT Kernel implementation supports submission, delivery, and transfer using standard means, i.e., P3 and P1. A class 'B' MT Kernel need not support the transfer of previously transferred messages, probes, or reports.

- o A class 'C' MT Kernel implementation requires support for transfer of messages, probes, and reports to another MT Kernel using standard means. A class 'C' MT Kernel does not require support for the transfer of previously transferred messages, probes, and reports, and message submission and delivery is achieved by non-standard means.

An MTA may conform to one or more of the MT Kernel classes. For example, a class 'B' or 'C' MT Kernel which supports the transfer of previously transferred messages, probes, and reports is also conformant to a class 'A' MT Kernel. Figure 8.10 illustrates several combinations of MT Kernel conformance classes. Additional combinations are possible.

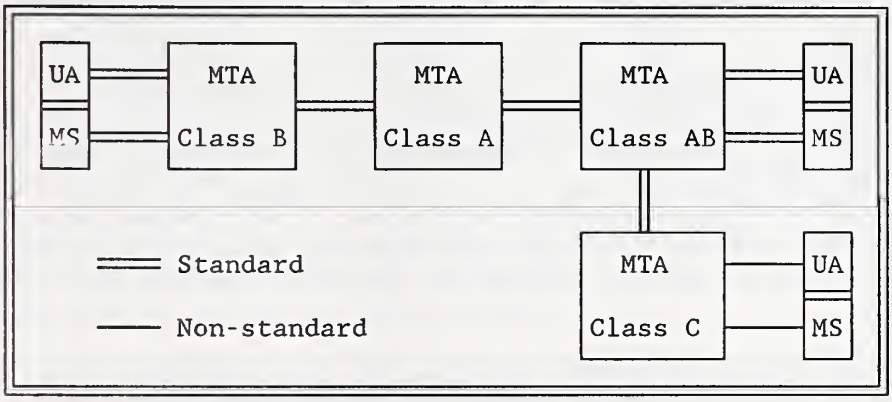


Figure 8.10. MT Kernel Conformance Classes.

2) Optional elements of the IPM Kernel need not be supported in the MT Kernel in this configuration, for example Probe and Deferred Delivery Cancellation.

3) The designation of a '+' in a configuration (e.g., 'MTA+MS') implies that there is no exposed protocol in the interface between the two components.

4) There are two conformance levels defined for MS support:

o A Basic MS only requires support for the General Attributes as specified in section 8.17.5

o An IPM MS requires support from both the General Attributes and IPM Attributes as specified in sections 8.17.5 and 8.17.6, respectively.

8.17 APPENDIX A: MHS PROTOCOL SPECIFICATIONS

The following tables (8.25-8.30) specify the requirements for support of MHS protocol elements for conformance to this Agreement. It should be noted that the tables specify minimum support for conformance to the relevant Kernel functional groups and where appropriate also specify enhanced support requirements where one or more further functional groups are claimed. All element support is subject to further review and may be upgraded in later versions of this Agreement.

The protocol support classification scheme used in this version of this Agreement is described below. However, it should be noted that the scheme is currently under review both within the NIST X.400 SIG and in the EWOS/ETSI MHS groups and is likely to be revised for later versions of this Agreement.

The classification of support for a protocol element specifies the requirements for implementations conforming to this Agreement to be able to generate, receive and process that protocol element, as appropriate (the 'receiving' role includes relaying where appropriate). The classification of support for each protocol element is relative to that for its containing element. Where subelements within a containing element are not listed, then their support classification shall be assumed to be that of the containing element. Where the range of values to be supported for an element is not specified, then all values defined in the base standard shall be supported.

The classifications have been revised. The new classifications relate to the classifications in the 1988 Stable Agreements as shown in table 8.24.

Table 8.24. Classification Changes

Former Category	New	
	Originator Category	Recipient Category
Generatable (G)	Mandatory (M)	Mandatory (M)
Supported (H)	Optional (O)	Mandatory (M)
Mandatory (M)	Mandatory (M)	Mandatory (M)
Required (R)	Mandatory (M)	Mandatory (M)
Unsupported (X)	Optional (O)	Optional (O)

The support classifications are stated for both Origination and Reception (O/R) in the following tables (8.25-8.30). The defined support for each is stated within each classification.

Mandatory (M)

Origination: Implementations conforming to this Agreement shall generate this element in all information objects in which, according to the base standards, it shall occur.

Reception: Implementations conforming to this Agreement shall process this element appropriately, and shall regard its absence as a protocol violation unless otherwise specified in the base standards. When an MS or MTA receives a protocol element that according to the base standard

and this profile should be conveyed to another MHS entity (MTA, MS, or UA), the MS or MTA is required to preserve the semantics of that protocol element in the message conveyed.

Optional (O)

Origination: Implementations conforming to this Agreement may optionally be capable of generating this protocol element, but are not required to do so.

Reception: Implementations conforming to this Agreement may, but are not required to be capable of processing this protocol element other than to observe any criticality indication. When an MS or MTA receives a protocol element that according to the base standard and this profile should be conveyed to another MHS entity (MTA, MS, or UA), the MS or MTA is required to preserve the semantics of that protocol element in the message conveyed. The absence of this element should not be assumed to convey any significance.

Note: Some protocol elements may not be conveyed, if downgrading rules are applied.

To Be Determined (*) - the support classification for this protocol element has yet to be determined.

8.17.1 MTS Transfer Protocol (P1)

Table 8.25. Classification of the P1 Protocol Elements

MTS Transfer Protocol (P1)				Part 1 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
Operations				
MTABind	M	M/M	M/M	MTABind
MTAUnbind	M	M/M	M/M	
MTSE				See protocol elements
MessageTransfer	M	M/M	M/M	
ProbeTransfer	M	M/M	M/M	
ReportTransfer	M	M/M	M/M	
Arguments/Results				
MTABind				
ARGUMENT				
<NULL>	O	O/M	O/M	See Note 2
<SET>	O	M/M	M/M	
initiator-name	M	M/M	M/M	
initiator-credentials	M	M/M	M/M	
simple	O	M/M	M/M	
strong	O	O/O	O/O	
security-context	O	O/O	O/O	
RESULT				
<NULL>	O	O/M	O/M	See Note 2
<SET>	O	M/M	M/M	
responder-name	M	M/M	M/M	
responder-credentials	M	M/M	M/M	
simple	O	M/M	M/M	
strong	O	O/O	O/O	

Notes:

1) The MT Kernel implementation classes are defined in sec. 8.16.

2) The action to be taken on receipt of null MTABind authentication is that an implementation must understand the semantics, but the form of authentication that is acceptable is a local matter.

MTS Transfer Protocol (P1)				Part 2 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
MTS-APDU				
message	M	M/M	O/M	
envelope	M	M/M	M/M	MessageTransferEnvelope
content	M	M/M	M/M	See P2 - else undefined
probe	M	M/M	O/M	ProbeTransferEnvelope
report	M	M/M	M/M	
envelope	M	M/M	M/M	ReportTransferEnvelope
content	M	M/M	M/M	ReportTransferContent
MessageTransferEnvelope				
message-identifier	M	M/M	M/M	MTSIdentifier
originator-name	M	M/M	M/M	ORName
original-encoded-information-types	O	M/M	O/O	EncodedInformationTypes
content-type	M	M/M	M/M	
built-in	O	M/M	O/O	
external	O	O/M	O/O	
content-identifier	O	O/M	O/O	
priority	O	M/M	O/M	All values
per-message-indicators	O	M/M	O/M	
disclosure-of-recipients	O	O/M	O/M	
implicit-conversion-prohibited	O	M/M	O/M	
alternate-recipient-allowed	O	M/M	O/O	
content-return-request	O	O/O	O/O	
deferred-delivery-time	O	O/O	O/O	
per-domain-bilateral-information	O	O/O	O/O	PerDomainBilateralInfo
trace-information	M	M/M	M/M	TraceInformation
extensions	O	M/M	M/M	ExtensionField
recipient-reassignment-prohibited	O	M/M	M/M	
dl-expansion-prohibited	O	M/M	O/M	
conversion-with-loss-prohibited	O	O/M	O/M	
latest-delivery-time	O	O/O	O/O	See X.411, 14.1.1 note 2
originator-return-address	O	O/O	O/O	
originator-certificate	O	O/O	O/O	
content-confidentiality-algorithm-identifier	O	O/O	O/O	
message-origin-authentication-check	O	O/O	O/O	
message-security-label	O	O/O	O/O	
content-correlator	O	O/O	O/O	

MTS Transfer Protocol (P1)				Part 3 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
dl-expansion-history	O	O/M	O/M	DLExpansionHistory
internal-trace-information	O	M/M	M/M	InternalTraceInfo
per-recipient-fields	M	M/M	M/M	
recipient-name	M	M/M	M/M	ORName
originally-specified-recipient-number	M	M/M	M/M	
per-recipient-indicators	M	M/M	M/M	
explicit-conversion	O	O/O	O/O	
extensions	O	O/M	O/M	ExtensionField
originator-requested-alternate-recipient	O	O/O	O/O	
requested-delivery-method	O	M/M	O/M	
physical-forwarding-prohibited	O	O/O	O/O	
physical-forwarding-address-request	O	O/O	O/O	
physical-delivery-modes	O	O/O	O/O	
registered-mail-type	O	O/O	O/O	
recipient-number-for-advice	O	O/O	O/O	
physical-rendition-attributes	O	O/O	O/O	
physical-delivery-report-request	O	O/O	O/O	
message-token	O	O/O	O/O	
content-integrity-check	O	O/O	O/O	
proof-of-delivery-request	O	O/O	O/O	
redirection-history	O	O/M	O/M	
ProbeTransferEnvelope				
probe-identifier	M	M/M	M/M	MTSIdentifier
originator-name	M	M/M	M/M	ORName
original-encoded-information-types	O	M/M	O/O	EncodedInformationTypes
content-type	M	M/M	M/M	
built-in	O	M/M	O/O	
external	O	O/M	O/O	
content-identifier	O	O/M	O/O	
content-length	O	M/M	O/O	
per-message-indicators	O	M/M	O/M	
disclosure-of-recipients	O	O/O	O/O	
implicit-conversion-prohibited	O	M/M	O/M	
alternate-recipient-allowed	O	M/M	O/O	
content-return-request	O	O/O	O/O	

MTS Transfer Protocol (P1)				Part 4 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
per-domain-bilateral-information	O	O/O	O/O	PerDomainBilateralInfo
trace-information	M	M/M	M/M	TraceInformation
extensions	O	M/M	M/M	ExtensionField
recipient-reassignment-prohibited	O	O/O	O/O	
dl-expansion-prohibited	O	M/M	O/M	
conversion-with-loss-prohibited	O	O/O	O/O	
originator-certificate	O	O/O	O/O	
message-security-label	O	O/O	O/O	
content-correlator	O	O/O	O/O	
probe-origin-authentication-check	O	O/O	O/O	
dl-expansion-history	O	O/M	O/M	DLExpansionHistory
internal-trace-information	O	M/M	M/M	InternalTraceInfo
per-recipient-fields	M	M/M	M/M	
recipient-name	M	M/M	M/M	ORName
originally-specified-recipient-number	M	M/M	M/M	
per-recipient-indicators	M	M/M	M/M	
explicit-conversion	O	O/O	O/O	
extensions	O	O/M	O/M	ExtensionField
originator-requested-alternate-recipient	O	O/O	O/O	
requested-delivery-method	O	M/M	O/M	
physical-rendition-attributes	O	O/O	O/O	
redirection-history	O	O/M	O/M	
ReportTransferEnvelope				
report-identifier	M	M/M	M/M	MTSIdentifier
report-destination-name	M	M/M	M/M	ORName
trace-information	M	M/M	M/M	TraceInformation
extensions	O	M/M	M/M	ExtensionField
message-security-label	O	O/O	O/O	
originator-and-DL-expansion-history	O	M/M	O/O	OriginatorAndDL ExpansionHistory
reporting-DL-name	O	O/O	O/O	
reporting-MTA-certificate	O	O/O	O/O	
report-origin-authentication-check	O	O/O	O/O	
internal-trace-information	O	M/M	M/M	InternalTraceInfo

MTS Transfer Protocol (P1)				Part 5 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
ReportTransferContent				
subject-identifier	M	M/M	M/M	MTSIdentifier
subject-intermediate-trace-information	O	M/M	M/M	TraceInformation
original-encoded-information-types	O	M/M	M/M	EncodedInformationTypes
content-type	O	M/M	M/M	
built-in	O	M/M	M/M	
external	O	M/M	M/M	
content-identifier	O	M/M	M/M	
returned-content	O	O/M	O/O	
additional-information	O	O/O	O/O	
extensions	O	O/M	O/M	ExtensionField
content-correlator	O	O/M	O/M	
per-recipient-fields	M	M/M	M/M	
actual-recipient-name	M	M/M	M/M	ORName
originally-specified-recipient-number	M	M/M	M/M	
per-recipient-indicators	M	M/M	M/M	
last-trace-information	M	M/M	M/M	
arrival-time	M	M/M	M/M	
converted-encoded-information-types	O	M/M	M/M	EncodedInformationTypes
report	M	M/M	M/M	
delivery	O	M/M	O/O	
message-delivery-time	O	M/M	M/M	
type-of-MTS-user	O	M/M	O/O	All values =O/M
non-delivery	O	M/M	M/M	
non-delivery-reason-code	O	M/M	M/M	
non-delivery-diagnostic-code	O	O/M	O/M	
originally-intended-recipient-name	O	M/M	M/M	ORName
supplementary-information	O	O/O	O/O	
extensions	O	M/M	M/M	ExtensionField
redirection-history	O	M/M	M/M	RedirectionHistory
physical-forwarding-address	O	O/O	O/O	
recipient-certificate	O	O/O	O/O	
proof-of-delivery	O	O/O	O/O	

MTS Transfer Protocol (P1)				Part 6 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
Common Data Types				
EncodedInformationTypes				
built-in-encoded-information-types	M	M/M	M/M	See Note 3
non-basic-parameters	O	O/O	O/O	
external-encoded-information-types	O	O/M	O/M	
MTSIdentifier				
global-domain-identifier	M	M/M	M/M	GlobalDomainIdentifier
local-identifier	M	M/M	M/M	
PerDomainBilateralInfo				
country-name	M	M/M	M/M	
administration-domain-name	O	M/M	M/M	DomainName
private-domain-identifier	O	M/M	M/M	DomainName (only encoded as SEQ if both present)
bilateral-information	M	M/M	M/M	
TraceInformation				
TraceInformationElement	M	M/M	M/M	
global-domain-identifier	M	M/M	M/M	GlobalDomainIdentifier
domain-supplied-information	M	M/M	M/M	
arrival-time	M	M/M	M/M	
routing-action	M	M/M	M/M	
relayed	O	M/M	M/M	
rerouted	O	O/M	O/M	
attempted-domain	O	O/M	O/M	GlobalDomainIdentifier
deferred-time	O	M/M	M/M	
converted-encoded-information-types	O	O/M	O/M	EncodedInformationTypes
other-actions	O	O/M	O/M	
redirected	O	O/M	O/M	
dl-operation	O	O/M	O/M	
ExtensionField				
type	M	M/M	M/M	
criticality	O	O/M	O/M	
for-submission	O	O/O	O/O	
for-transfer	O	M/M	M/M	
for-delivery	O	M/M	M/M	

MTS Transfer Protocol (P1)				Part 7 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
value	M	M/M	M/M	
DLExpansionHistory				
DLExpansion	M	M/M	M/M	
ORAddressAndOptionalDirectory Name	M	M/M	M/M	ORName
dl-expansion-time	M	M/M	M/M	
InternalTraceInformation				
InternalTraceInformationElement	M	M/M	M/M	
global-domain-identifier	M	M/M	M/M	GlobalDomainIdentifier
mta-name	M	M/M	M/M	
mta-supplied-information	M	M/M	M/M	
arrival-time	M	M/M	M/M	
routing-action	M	M/M	M/M	
relayed	O	M/M	M/M	
rerouted	O	O/M	O/M	
attempted	O			
mta	O	O/M	O/M	
domain	O	O/M	O/M	GlobalDomainIdentifier
deferred-time	O	O/M	O/M	
other-actions	O	O/M	O/M	
redirected	O	O/M	O/M	
dl-operation	O	O/M	O/M	
OriginatorAndDLExpansionHistory				
originator-or-dl-name	M	M/M	M/M	
origination-or-expansion-time	M	M/M	M/M	
RedirectionHistory				
Redirection	M	M/M	M/M	
intended-recipient-name	M	M/M	M/M	
ORAddressAndOptionalDirectory Name	M	M/M	M/M	ORName
redirection-time	M	M/M	M/M	
redirection-reason	M	M/M	M/M	
ORName				
address	M	M/M	M/M	
standard-attributes	M	M/M	M/M	
country-name	O	M/M	O/M	CountryName
administration-domain-name	O	M/M	O/M	DomainName
network-address	O	M/M	O/M	

MTS Transfer Protocol (P1)				Part 8 of 9
MT Kernel Support by MTS Class				Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
terminal-identifier	O	M/M	O/M	
private-domain-name	O	M/M	O/M	DomainName
organization-name	O	M/M	O/M	
numeric-user-identifier	O	M/M	O/M	
personal-name	O	M/M	O/M	
surname	M	M/M	M/M	
given-name	O	M/M	O/M	
initials	O	M/M	O/M	
generation-qualifier	O	M/M	O/M	
organizational-unit-names	O	M/M	O/M	
OrganizationUnitName	M	M/M	O/M	
domain-defined-attributes	O	M/M	O/M	
DomainDefinedAttribute	M	M/M	O/M	
type	M	M/M	M/M	
value	M	M/M	M/M	
extension-attributes	O	O/M	O/M	ExtensionAttribute
common-name	O	O/M	O/M	
teletex-common-name	O	O/M	O/M	
teletex-organization-name	O	O/M	O/M	
teletex-personal-name	O	O/M	O/M	
teletex-organizational-unit-names	O	O/M	O/M	
teletex-domain-defined-attributes	O	O/M	O/M	
pds-name	O	O/M	O/M	
physical-delivery-country-name	O	O/M	O/M	
postal-code	O	O/M	O/M	
physical-delivery-office-name	O	O/M	O/M	
physical-delivery-office-number	O	O/M	O/M	
extension-OR-address-components	O	O/M	O/M	
physical-delivery-personal-name	O	O/M	O/M	
physical-delivery-organization-name	O	O/M	O/M	
extension-physical-delivery-address-components	O	O/M	O/M	
unformatted-postal-address	O	O/M	O/M	
street-address	O	O/M	O/M	
post-office-box-address	O	O/M	O/M	
poste-restante-address	O	O/M	O/M	

MTS Transfer Protocol (P1)				Part 9 of 9
	MT Kernel Support by MTS Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
unique-postal-name	O	O/M	O/M	
local-postal-attributes	O	O/M	O/M	
extended-network-address	O	O/M	O/M	
terminal-type	O	O/M	O/M	
directory-name	O	O/O	O/O	
ExtensionAttribute				
extension-attribute-type	M	M/M	M/M	
extension-attribute-value	M	M/M	M/M	
GlobalDomainIdentifier				
country-name	M	M/M	M/M	CountryName
administration-domain-name	M	M/M	M/M	DomainName
private-domain-identifier	O	M/M	O/M	DomainName
CountryName				
x121-dcc-code	O	O/M	O/M	
iso-3166-alpha2-code	O	M/M	O/M	
DomainName				
numeric	O	O/M	O/M	
printable	O	M/M	O/M	

Notes (continued):

3) An implementation is only required to generate EITs that correspond to the body parts it is capable of generating.

8.17.2 Interpersonal Messaging Protocol (P2)

Table 8.26. Classification of the P2 Protocol Elements

Interpersonal Messaging Protocol (P2)			Part 1 of 3
	Support by UA		
Protocol Element	S	O/R	Comments/References
InformationObject			
ipm	O	M/M	IPM
ipn	O	M/M	IPN - see Note 4
IPM			
heading	M	M/M	
this-IPM	M	M/M	IPMIdentifier
originator	O	M/M	ORDescriptor
authorizing-users	O	O/M	RecipientSpecifier
primary-recipients	O	M/M	RecipientSpecifier
copy-recipients	O	M/M	RecipientSpecifier
blind-copy-recipients	O	O/M	RecipientSpecifier
replied-to-IPM	O	M/M	IPMIdentifier
obsoleted-IPMs	O	O/M	IPMIdentifier
related-IPMs	O	O/M	IPMIdentifier
subject	O	M/M	See Note 1, 8
expiry-time	O	O/M	
reply-time	O	O/M	
reply-recipients	O	O/M	ORDescriptor
importance	O	O/M	
sensitivity	O	O/M	
auto-forwarded	O	O/M	
extensions	O	O/M	HeadingExtension
incomplete-copy	O	O/O	
languages	O	O/M	
body	M	M/M	BodyPart
IPN			
common-fields	M	M/M	
subject-ipm	M	M/M	
ipn-originator	O	M/M	ORDescriptor
ipm-preferred-recipient	O	M/M	ORDescriptor
conversion-eits	O	O/M	EncodedInformationTypes
non-receipt-fields	O	M/M	See Note 5
non-receipt-reason	M	M/M	
discard-reason	O	M/M	
auto-forward-comment	O	O/M	
returned-ipm	O	O/O	See Note 2
receipt-fields	O	O/M	
receipt-time	M	M/M	

Interpersonal Messaging Protocol (P2)			Part 2 of 3
	Support by UA		
Protocol Element	S	O/R	Comments/References
acknowledgment-mode	O	O/M	
suppl-receipt-info	O	O/O	
HeadingExtension			
type	M	M/M	
value	M	M/M	
IPMIdentifier			
user	O	O/M	
user-relative-identifier	M	M/M	
ORDescriptor			
formal-name	O	O/M	ORName - see Note 3
free-form-name	O	O/M	See Note 8
telephone-number	O	O/M	
RecipientSpecifier			
recipient	M	M/M	ORDescriptor
notification-requests	O	O/M	
reply-requested	O	O/M	
BodyPart			
ia5-text	O	M/M	
parameters	M	M/M	
repertoire	O	O/M	See Note 6
data	M	M/M	
voice	O	*	See Note 7
g3-facsimile	O	O/O	
parameters	M	M/M	
number-of-pages	O	O/M	
non-basic-parameters	O	O/M	
data	M	M/M	
g4-class1	O	O/O	
teletex	O	O/O	
parameters	M	M/M	
number-of-pages	O	O/O	
telex-compatible	O	O/O	
non-basic-parameters	O	O/O	
data	M	M/M	
videotex	O	O/O	
parameters	M	M/M	
syntax	O	O/M	
data	M	M/M	
encrypted	O	*	See Note 7

Interpersonal Messaging Protocol (P2)			Part 3 of 3
	Support by UA		
Protocol Element	S	O/R	Comments/References
message	O	O/M	
parameters	M	M/M	
delivery-time	O	O/M	
delivery-envelope	O	O/M	See P3 OtherMessage DeliveryFields
data	M	M/M	
mixed-mode	O	O/O	
bilaterally-defined	O	O/O	
nationally-defined	O	O/O	
externally-defined	O	O/M	
parameters	M	M/M	
data	M	M/M	
GeneralTextBodyPart	O	*	

Notes:

1) The ability to generate the maximum size subject is not required.

2) May only be included if specifically requested by the originator.

3) The ORName should be specified wherever possible.

4) The ability to generate an IPN is optional in the case of an implementation in which a non-receipt condition cannot occur and receipt notification is not supported (see table 8.5).

5) The ability to generate non-receipt-fields is optional in the case of an implementation in which a non-receipt condition cannot occur (see Note 4).

6) Only the IA5 repertoire has to be supported for an ia5-text body part on reception.

7) The definition of these body parts is for further study in CCITT and ISO.

8) Only the IA5 subset of the T.61 character repertoire need be generated. All T.61 characters should be supported on reception.

8.17.3 MTS Access Protocol (P3)

Note: The support classifications for the IPM UA, MS and MTA below indicate the minimum level of support required by implementations conforming to these Agreements, and should not be misconstrued as a redefinition of any of the MHS application contexts.

Table 8.27. Classification of the P3 Protocol Elements

MTS Access Protocol (P3)					Part 1 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
Operations					
MTSBind	M	M/M	M/M	M/M	MTSBind
MTSUnbind	M	M/M	M/M	M/M	
MSSE					
message-submission	M	M/-	M/M	-/M	MessageSubmission
probe-submission	M	O/-	M/M	-/M	ProbeSubmission
cancel-deferred-delivery	M	O/-	M/M	-/M	CancelDeferredDelivery
submission-control	M	-/M	M/M	O/-	SubmissionControl
MDSE					
message-delivery	M	-/M	M/M	M/-	MessageDelivery
report-delivery	M	-/M	M/M	M/-	ReportDelivery
delivery-control	M	O/-	O/-	-/M	DeliveryControl
MASE					
register	M	O/-	M/M	-/M	Register
change-credentials					
(MTS to MTSuser)	M	-/M	M/M	O/-	ChangeCredentials
(MTSuser to MTS)	M	O/-	M/M	-/M	ChangeCredentials

Note: A Message Store must pass through all MSSE and MASE operations unaltered.

MTS Access Protocol (P3)					Part 2 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
Arguments/Results					
MTSBind					MTS to MTS User
ARGUMENT					
initiator-name	M	-/M	-/M	M/-	
mTS-user	-	-/-	-/-	-/-	
mTA	O	-/O	-/M	M/-	
isMessageStore	-	-/-	-/-	-/-	
messages-waiting	O	-/O	-/O	O/-	
initiator-credentials	M	-/M	-/M	M/-	
simple	O	-/M	-/M	M/-	
strong	O	-/O	-/O	O/-	
security-context	O	-/O	-/O	O/-	1-256
RESULT					
responder-name	M	M/-	M/-	-/M	
mTS-user	O	M/-	M/-	-/M	
mTA	-	-/-	-/-	-/-	
isMessageStore	O	M/-	M/-	-/M	
messages-waiting	-	-/-	-/-	-/-	
responder-credentials	M	M/-	M/-	-/M	
simple	O	M/-	M/-	-/M	
strong	O	O/-	O/-	-/O	
MTSBind					MTS User to MTS
ARGUMENT					
initiator-name	M	M/-	M/-	-/M	
mTS-user	O	M/-	M/-	-/M	
mTA	-	-/-	-/-	-/-	
isMessageStore	O	M/M	M/-	-/M	
messages-waiting	-	-/-	-/-	-/-	
initiator-credentials	M	M/-	M/-	-/M	
simple	O	M/-	M/-	-/M	
strong	O	O/-	O/-	-/O	
security-context	O	O/-	O/-	-/O	1-256
RESULT					
responder-name	M	-/M	-/M	M/-	
mTS-user	-	-/-	-/-	-/-	
mTA	O	-/M	-/M	M/-	
isMessageStore	-	-/-	-/-	-/-	
messages-waiting	O	-/O	-/O	O/-	
responder-credentials	M	-/M	-/M	M/-	
simple	O	-/M	-/M	M/-	
strong	O	-/O	-/O	O/-	

MTS Access Protocol (P3)					Part 3 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
MessageSubmission					
ARGUMENT					
envelope	M	M/-	M/-	-/M	MessageSubmission Envelope
content	M	M/-	M/-	-/M	
RESULT					
message-submission-identifier	M	-/M	-/M	M/-	See P1 MTSIdentifier
message-submission-time	M	-/M	-/M	M/-	
content-identifier	O	-/M	-/M	M/-	
extensions	O	-/O	-/O	O/-	
originating-MTA-certificate	O	-/O	-/O	O/-	
proof-of-submission	O	-/O	-/O	O/-	
ProbeSubmission					
ARGUMENT					
envelope	M	M/-	M/-	-/M	ProbeSubmission Envelope
RESULT					
probe-submission-identifier	M	-/M	-/M	M/-	See P1 MTSIdentifier
probe-submission-time	M	-/M	-/M	M/-	
content-identifier	O	-/M	-/M	M/-	
CancelDeferredDelivery					
ARGUMENT					
message-submission-identifier	M	M/-	M/-	-/M	See P1 MTSIdentifier
SubmissionControl					
ARGUMENT					
controls	M	-/M	-/M	M/-	See Note 1
restrict	O	-/M	-/M	O/-	
permissible-operations	O	-/M	-/M	O/-	
permissible-maximum-content-length	O	-/M	-/M	O/-	
permissible-lowest-priority	O	-/M	-/M	O/-	
permissible-security-context	O	-/O	-/O	O/-	
RESULT					
waiting	M	M/-	M/-	-/M	See Note 2
waiting-operations	O	O/-	O/-	-/M	0-16
waiting-messages	O	O/-	O/-	-/M	
waiting-content-types	O	O/-	O/-	-/M	0-1024
waiting-encoded-information-types	O	O/-	O/-	-/M	See P1 Encoded InformationTypes

MTS Access Protocol (P3)					Part 4 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
MessageDelivery					
ARGUMENT					
envelope	M	-/M	-/M	M/-	MessageDeliveryEnvelope
content	M	-/M	-/M	M/-	
RESULT					
recipient-certificate	O	O/-	O/-	-/O	
proof-of-delivery	O	O/-	O/-	-/O	
ReportDelivery					
ARGUMENT					
envelope	M	-/M	-/M	M/-	ReportDeliveryEnvelope
returned-content	O	-/M	-/M	O/-	
DeliveryControl					
ARGUMENT					
controls	M	M/-	M/-	-/M	See Note 3
restrict	O	O/-	O/-	-/M	
permissible-operations	O	O/-	O/-	-/M	
permissible-maximum-content-length	O	O/-	O/-	-/M	
permissible-lowest-priority	O	O/-	O/-	-/M	
permissible-content-types	O	O/-	O/-	-/M	
permissible-encoded-information-types	O	O/-	O/-	-/M	See P1 Encoded InformationTypes
permissible-security-context	O	O/-	O/-	-/O	
RESULT					
waiting	M	-/M	-/M	M/-	See Note 4
waiting-operations	O	-/M	-/M	O/-	
waiting-messages	O	-/M	-/M	O/-	
waiting-content-types	O	-/M	-/M	O/-	
waiting-encoded-information-types	O	-/M	-/M	O/-	See P1 Encoded InformationTypes
Register					See Note 5
ARGUMENT					
user-name	O	O/-	O/-	-/O	See X.411, 8.4.1.1.1.1
user-address	O	O/-	O/-	-/O	
deliverable-encoded-information-types	O	O/-	M/-	-/M	See P1 Encoded InformationTypes
deliverable-maximum-content-length	O	O/-	M/-	-/M	
default-delivery-controls	O	O/-	O/-	-/M	
restrict	O	O/-	O/-	-/M	

MTS Access Protocol (P3)					Part 5 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
permissible-operations	O	O/-	O/-	-/M	
permissible-maximum-content-length	O	O/-	O/-	-/M	
permissible-lowest-priority	O	O/-	O/-	-/M	
permissible-content-types	O	O/-	O/-	-/M	1-1024
permissible-encoded-information-types	O	O/-	O/-	-/M	See P1 Encoded InformationTypes
deliverable-content-types	O	O/-	M/-	-/M	1-1024
labels-and-redirections	O	O/-	O/-	-/O	1-256
user-security-label	O	O/-	O/-	-/O	
recipient-assigned-alternate-recipient	O	O/-	O/-	-/O	
ChangeCredentials					MTS to MTSuser
ARGUMENT					
old-credentials	M	-/M	-/M	M/-	
simple	O	-/M	-/M	O/-	
strong	O	-/O	-/O	O/-	
new-credentials	M	-/M	-/M	M/-	
simple	O	-/M	-/M	O/-	
strong	O	-/O	-/O	O/-	
ChangeCredentials					MTSuser to MTS
ARGUMENT					
old-credentials	M	M/-	M/-	-/M	
simple	O	O/-	O/-	-/M	
strong	O	O/-	O/-	-/O	
new-credentials	M	M/-	M/-	-/M	
simple	O	O/-	O/-	-/M	
strong	O	O/-	O/-	-/O	
MessageSubmissionEnvelope					See Note 6
originator-name	M	M/-	M/-	-/M	See P1 ORName
original-encoded-information-types	O	M/-	M/-	-/M	See P1 Encoded InformationTypes
content-type	M	M/-	M/-	-/M	
built-in	O	O/-	M/-	-/M	
external	O	O/-	M/-	-/M	
content-identifier	O	O/-	M/-	-/M	1-16
priority	O	M/-	M/-	-/M	All values
per-message-indicators	O	M/-	M/-	-/M	
disclosure-of-recipients	O	O/-	M/-	-/M	

MTS Access Protocol (P3)					Part 6 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
implicit-conversion-prohibited	O	M/-	M/-	-/M	
alternate-recipient-allowed	O	M/-	M/-	-/M	
content-return-request	O	O/-	M/-	-/M	
deferred-delivery-time	O	M/-	M/-	-/M	
extensions	O	M/-	M/-	-/M	
recipient-reassignment-prohibited	O	O/-	M/-	-/M	
dl-expansion-prohibited	O	M/-	M/-	-/M	
conversion-with-loss-prohibited	O	O/-	M/-	-/M	
latest-delivery-time	O	O/-	M/-	-/M	
originator-return-address	O	O/-	M/-	-/M	
originator-certificate	O	O/-	O/-	-/O	
content-confidentiality-algorithm-identifier	O	O/-	O/-	-/O	
message-origin-authentication-check	O	O/-	O/-	-/O	
message-security-label	O	O/-	O/-	-/O	
proof-of-submission-request	O	O/-	O/-	-/O	
content-correlator	O	O/-	M/-	-/M	
forwarding-request	O	O/-	M/-	-/M	MS Abstract Service only
PerRecipientMessageSubmission Fields	M	M/-	M/-	-/M	1-32767
recipient-name	M	M/-	M/-	-/M	See P1 ORName
originator-report-request	M	M/-	M/-	-/M	
explicit-conversion	O	O/-	M/-	-/M	
extensions	O	M/-	M/-	-/M	
originator-requested-alternate-recipient	O	O/-	O/-	-/O	
requested-delivery-method	O	M/-	M/-	-/M	
physical-forwarding-prohibited	O	O/-	M/-	-/M	
physical-forwarding-address-request	O	O/-	O/-	-/O	
physical-delivery-modes	O	O/-	O/-	-/O	
registered-mail-type	O	O/-	O/-	-/O	
recipient-number-for-advice	O	O/-	O/-	-/O	
physical-rendition-attributes	O	O/-	O/-	-/O	
physical-delivery-report-request	O	O/-	O/-	-/O	
message-token	O	O/-	O/-	-/O	
content-integrity-check	O	O/-	O/-	-/O	
proof-of-delivery-request	O	O/-	O/-	-/O	

MTS Access Protocol (P3)					Part 7 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
ProbeSubmissionEnvelope					See Note 6
originator-name	M	M/-	M/-	-/M	See P1 ORName
original-encoded-information-types	O	M/-	M/-	-/M	See P1 Encoded InformationTypes
content-type	M	M/-	M/-	-/M	
built-in	O	O/-	M/-	-/M	0-32767
external	O	O/-	M/-	-/M	
content-identifier	O	O/-	M/-	-/M	1-16
content-length	O	M/-	M/-	-/M	0-'7FFFFFF'H
per-message-indicators	O	M/-	M/-	-/M	
implicit-conversion-prohibited	O	M/-	M/-	-/M	
alternate-recipient-allowed	O	O/-	M/-	-/M	
extensions	O	M/-	M/-	-/M	
recipient-reassignment-prohibited	O	O/-	M/-	-/M	
dl-expansion-prohibited	O	M/-	M/-	-/M	
conversion-with-loss-prohibited	O	O/-	M/-	-/M	
originator-certificate	O	O/-	O/-	-/O	
message-security-label	O	O/-	O/-	-/O	
content-correlator	O	O/-	M/-	-/M	
probe-origin-authentication-check	O	O/-	O/-	-/O	
PerRecipientProbeSubmissionFields	M	M/-	M/-	-/M	1-32767
recipient-name	M	M/-	M/-	-/M	See P1 ORName
originator-report-request	M	M/-	M/-	-/M	
explicit-conversion	O	O/-	M/-	-/M	0-256
extensions	O	M/-	M/-	-/M	
originator-requested-alternate-recipient	O	O/-	O/-	-/O	
requested-delivery-method	O	M/-	M/-	-/M	0-256
physical-rendition-attributes	O	O/-	M/-	-/M	
MessageDeliveryEnvelope					See Note 7
message-delivery-identifier	M	-/M	-/M	M/-	See P1 MTSIdentifier
message-delivery-time	M	-/M	-/M	M/-	
other-fields	M	-/M	-/M	M/-	
content-type	M	-/M	-/M	M/-	
built-in	O	-/M	-/M	M/-	0-32767
external	O	-/M	-/M	M/-	
originator-name	M	-/M	-/M	M/-	See P1 ORName

MTS Access Protocol (P3)					Part 8 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
original-encoded-information-types	O	-/M	-/M	M/-	See P1 Encoded InformationTypes
priority	O	-/M	-/M	M/-	All values
delivery-flags	O	-/M	-/M	M/-	
implicit-conversion-prohibited	O	-/M	-/M	M/-	
other-recipient-names	O	-/M	-/M	M/-	See P1 ORName
this-recipient-name	M	-/M	-/M	M/-	See P1 ORName
originally-intended-recipient-name	O	-/M	-/M	M/-	See P1 ORName
converted-encoded-information-types	O	-/M	-/M	M/-	See P1 Encoded InformationTypes
message-submission-time	M	-/M	-/M	M/-	
content-identifier	O	-/M	-/M	M/-	1-16
extensions	O	-/M	-/M	M/-	
conversion-with-loss-prohibited	O	-/M	-/M	M/-	
requested-delivery-method	O	-/M	-/M	M/-	
physical-forwarding-prohibited	O	-/M	-/M	M/-	
physical-forwarding-address-request	O	-/M	-/M	M/-	
physical-delivery-modes	O	-/M	-/M	M/-	0-16
registered-mail-type	O	-/M	-/M	M/-	0-256
recipient-number-for-advice	O	-/M	-/M	M/-	1-32
physical-rendition-attributes	O	-/M	-/M	M/-	
physical-delivery-report-request	O	-/M	-/M	M/-	0-256
originator-return-address	O	-/M	-/M	M/-	
originator-certificate	O	-/O	-/O	O/-	
message-token	O	-/O	-/O	O/-	
content-confidentiality-algorithm-identifier	O	-/O	-/O	O/-	
content-integrity-check	O	-/O	-/O	O/-	
message-origin-authentication-check	O	-/O	-/O	O/-	
message-security-label	O	-/O	-/O	O/-	
proof-of-delivery-request	O	-/O	-/O	O/-	
redirection-history	O	-/M	-/M	M/-	1-512
dl-expansion-history	O	-/M	-/M	M/-	1-512

MTS Access Protocol (P3)					Part 9 of 10
	Support by: IPM				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
ReportDeliveryEnvelope					See Note 7
subject-submission-identifier	M	-/M	-/M	M/-	See P1 MTSIdentifier
content-identifier	O	-/M	-/M	M/-	
content-type	O	-/M	-/M	M/-	
built-in	O	-/M	-/M	M/-	0-32767
external	O	-/M	-/M	M/-	
original-encoded-information-types	O	-/M	-/M	M/-	See P1 Encoded InformationTypes
extensions	O	-/M	-/M	M/-	
message-security-label	O	-/O	-/O	O/-	
content-correlator	O	-/M	-/M	M/-	
originator-and-DL-expansion-history	O	-/M	-/M	M/-	See P1 OriginatorAndDL ExpansionHistory
reporting-DL-name	O	-/M	-/M	M/-	
reporting-MTA-certificate	O	-/O	-/O	O/-	
report-origin-authentication-check	O	-/O	-/O	O/-	
PerRecipientReportDelivery-Fields	M	-/M	-/M	M/-	1-32767
actual-recipient-name	M	-/M	-/M	M/-	See P1 ORName
report	M	-/M	-/M	M/-	
delivery	O	-/M	-/M	M/-	
message-delivery-time	M	-/M	-/M	M/-	
type-of-MTS-user	O	-/M	-/M	M/-	
non-delivery	O	-/M	-/M	M/-	
non-delivery-reason-code	M	-/M	-/M	M/-	
non-delivery-diagnostic-code	O	-/M	-/M	M/-	
converted-encoded-information-types	O	-/M	-/M	M/-	See P1 Encoded InformationTypes
originally-intended-recipient-name	O	-/M	-/M	M/-	See P1 ORName
supplementary-information	O	-/M	-/M	M/-	1-256
extensions	O	-/M	-/M	M/-	
redirection-history	O	-/M	-/M	M/-	See P1 Redirection History, 1-512
physical-forwarding-address	O	-/M	-/M	M/-	
recipient-certificate	O	-/O	-/O	O/-	
proof-of-delivery	O	-/O	-/O	O/-	

MTS Access Protocol (P3)	Part 10 of 10

Notes:

1) The MTS-user may interpret any restriction as simply withhold 'all' submissions.

2) No explicit action needs to be taken by the MTA.

3) The MTA may interpret any restriction as simply withhold 'all' deliveries.

4) No explicit action needs to be taken by the MTS-user.

5) The Register operation may be performed locally (see X.411). Although not required for the UA for conformance, it is considered to be a useful service and support is recommended.

6) The action to be taken by a submitting MTA is as defined in X.411 (ISO 10021-4). In the absence of any specific processing requirements for a particular element in a submission envelope, the action to be taken is simply the faithful mapping of such element to the corresponding element of the appropriate transfer envelope.

7) The action to be taken by a delivering MTA is as defined in X.411 (ISO 10021-4). In the absence of any specific processing requirements for a particular element in a delivery envelope, the action to be taken is simply the creation of such element from the corresponding element of the appropriate transfer envelope.

8.17.4 MS Access Protocol (P7)

Table 8.28. Classification of the P7 Protocol Elements

MS Access Protocol (P7)				Part 1 of 6
	Support by: IPM			
Protocol Element	S	UA O/R	MS O/R	Comments/References
Operations				
MSBind	M	M/-	-/M	MSBind
MSUnbind	M	M/-	-/M	
MSSE				
message-submission	M	M/-	-/M	See P3 MessageSubmission
probe-submission	M	O/-	-/M	See P3 ProbeSubmission
cancel-deferred-delivery	M	O/-	-/M	See P3 CancelDeferred Delivery
submission-control	M	-/M	M/-	See P3 SubmissionControl
MASE				
register	M	O/-	-/M	See P3 Register
change-credentials (MS to UA)	M	-/M	M/-	See P3 ChangeCredentials
change-credentials (UA to MS)	M	O/-	-/M	See P3 ChangeCredentials
MRSE				
summarize	M	M/-	-/M	Summarize
list	M	M/-	-/M	List
fetch	M	M/-	-/M	Fetch
delete	M	M/-	-/M	Delete
register-ms	M	O/-	-/M	Register-MS
alert	M	-/O	O/-	Alert
Arguments/Results				
MSBind				
ARGUMENT				
MSBindArgument	M	M/-	-/M	
initiator-name	M	M/-	-/M	
initiator-credentials	M	M/-	-/M	
simple	O	M/-	-/M	
strong	O	O/-	-/O	
security-context	O	O/-	-/O	
fetch-restrictions	O	O/-	-/M	
allowed-content-types	O	O/-	-/M	
allowed-EITs	O	O/-	-/M	
maximum-content-length	O	O/-	-/M	
MS-configuration-request	O	O/-	-/M	

MS Access Protocol (P7)				Part 2 of 6
	Support by: IPM			
Protocol Element	S	UA O/R	MS O/R	Comments/References
RESULT				
MSBindResult	M	-/M	M/-	
responder-credentials	M	-/M	M/-	
simple	O	-/M	M/-	
strong	O	-/O	O/-	
available-auto-actions	O	-/M	M/-	1-16
available-attribute-types	O	-/M	M/-	1-1024
alert-indication	O	-/M	O/-	
content-types-supported	O	-/M	M/-	
Summarize				
ARGUMENT				
SummarizeArgument	M	M/-	-/M	
information-base-type	O	O/-	-/M	InformationBase
selector	M	M/-	-/M	Selector
summary-requests	O	O/-	-/M	1-16
RESULT				
SummarizeResult	M	-/M	M/-	
next	O	-/M	M/-	
count	M	-/M	M/-	0-'7FFFFFFF'H
span	O	-/M	M/-	
lowest	M	-/M	M/-	
highest	M	-/M	M/-	
summaries	O	-/M	M/-	1-16
absent	O	-/M	M/-	1-'7FFFFFFF'H
present	O	-/M	M/-	1-'7FFFFFFF'H
type	M	-/M	M/-	
value	M	-/M	M/-	
count	M	-/M	M/-	
List				
ARGUMENT				
ListArgument	M	M/-	-/M	
information-base-type	O	O/-	-/M	InformationBase
selector	M	M/-	-/M	Selector
requested-attributes	O	M/-	-/M	AttributeSelection
RESULT				
ListResult	M	-/M	M/-	
next	O	-/M	M/-	
requested	O	-/M	M/-	EntryInformation, 0-'7FFFFFFF'H

MS Access Protocol (P7)				Part 3 of 6
	Support by: IPM			
Protocol Element	S	UA O/R	MS O/R	Comments/References
Fetch				
ARGUMENT				
FetchArgument	M	M/-	-/M	
information-base-type	O	O/-	-/M	InformationBase
item	M	M/-	-/M	
search	O	M/-	-/M	Selector
precise	O	M/-	-/M	
requested-attributes	O	M/-	-/M	AttributeSelection
RESULT				
FetchResult	M	-/M	M/-	
entry-information	O	-/M	M/-	EntryInformation
list	O	-/M	M/-	0-'7FFFFFFF'H
next	O	-/M	M/-	
Delete				
ARGUMENT				
DeleteArgument	M	M/-	-/M	
information-base-type	O	O/-	-/O	InformationBase
items	M	M/-	-/M	
selector	O	M/-	-/M	Selector
sequence-numbers	O	M/-	-/M	1-'7FFFFFFF'H
RESULT				
DeleteResult	M	-/M	M/-	
Register-MS				
ARGUMENT				
Register-MSArgument	M	M/-	-/M	
auto-action-registrations	O	O/-	-/O	1-1024
type	M	M/-	-/M	
registration-identifier	O	M/-	-/M	
registration-parameter	M	M/-	-/M	See auto action registration parameters
auto-action-deregistrations	O	O/-	-/O	1-1024
type	M	M/-	-/M	
registration-identifier	O	M/-	-/M	
list-attribute-defaults	O	M/-	-/M	1-1024
fetch-attribute-defaults	O	M/-	-/M	1-1024
change-credentials	O	M/-	-/M	
old-credentials	M	M/-	-/M	
new-credentials	M	M/-	-/M	
user-security-labels	O	O/-	-/O	1-256
RESULT				
Register-MSResult	M	-/M	M/-	

MS Access Protocol (P7)				Part 4 of 6
	Support by: IPM			
Protocol Element	S	UA O/R	MS O/R	Comments/References
Alert				
ARGUMENT				
AlertArgument	M	-/M	M/-	
alert-registration-identifier	M	-/M	M/-	
new-entry	O	-/M	M/-	EntryInformation
RESULT				
AlertResult	O	M/-	-/M	
Auto Action Registration Parameters				
AutoForwardRegistrationParameter				
filter	O	O/-	-/M	Filter
auto-forward-arguments	M	M/-	-/M	
originator-name	M	M/-	-/M	
content-identifier	O	O/-	-/M	
priority	O	O/-	-/M	
per-message-indicators	O	O/-	-/M	See P3
deferred-delivery-time	O	O/-	-/M	
extensions	O	O/-	-/M	See P3
per-recipient-fields	M	M/-	-/M	
recipient-name	M	M/-	-/M	
originator-report-request	M	M/-	-/M	
explicit-conversion	O	O/-	-/M	
extensions	O	O/-	-/M	See P3
delete-after-auto-forwarding	O	O/-	-/M	
other-parameters	O	O/-	-/M	See Note 1
auto-forwarding-comment	O	O/-	-/M	
cover-note	O	O/-	-/M	
this-ipm-prefix	O	O/-	-/M	
AutoAlertRegistrationParameter				
filter	O	O/-	-/M	Filter
alert-addresses	O	O/-	-/O	
address	M	M/-	-/M	
alert-qualifier	O	O/-	-/O	
requested-attributes	O	O/-	-/M	AttributeSelection

Notes:

1) The specified syntax of other-parameters is for IPMS use only - see X.413 clause 12.1 and X.420 clause 19.4.

MS Access Protocol (P7)				Part 5 of 6
	Support by: IPM	UA	MS	
Protocol Element	S	O/R	O/R	Comments/References
Common Data Types				
AttributeSelection				
type	M	M/-	-/M	
from	O	O/-	-/M	1-32767
count	O	O/-	-/M	1-32767
AttributeValueAssertion				
type	M	M/-	-/M	
value	M	M/-	-/M	
EntryInformation				
sequence-number	M	-/M	M/-	
attributes	O	-/M	M/-	1-1024
type	M	-/M	M/-	
values	M	-/M	M/-	
Filter				
item	O	M/-	-/M	FilterItem
and	O	O/-	-/O	1-32
or	O	O/-	-/O	1-32
not	O	O/-	-/O	
FilterItem				
equality	O	M/-	-/M	AttributeValueAssertion (Support is O if ORname)
substrings	O	O/-	-/O	
type	M	M/-	-/M	
strings	M	M/-	-/M	
greater-or-equal	O	O/-	-/M	AttributeValueAssertion
less-or-equal	O	O/-	-/M	AttributeValueAssertion
present	O	O/-	-/M	
InformationBase				
stored-messages	O	M/-	-/M	
inlog	O	O/-	-/O	
outlog	O	O/-	-/O	
Range				
sequence-number-range	O	O/-	-/M	
from	O	O/-	-/M	
to	O	O/-	-/M	

MS Access Protocol (P7)				Part 6 of 6
	Support by: IPM			
Protocol Element	S	UA O/R	MS O/R	Comments/References
creation-time-range	O	O/-	-/M	
from	O	O/-	-/M	
to	O	O/-	-/M	
Selector				
child-entries	O	O/-	-/M	
range	O	O/-	-/M	Range
filter	O	O/-	-/M	Filter
limit	O	O/-	-/M	
override	O	O/-	-/M	

8.17.5 Message Store General Attribute Support

Table 8.29. Classification of the Message Store General Attributes

Message Store General Attribute Support					Part 1 of 2
Support by:		IPM UA	Bas MS	IPM MS	
Attribute	S	Rec	Org	Org	Comments/References
child-sequence-numbers	M	M	M	M	
content	M	M	M	M	
content-confidentiality-algorithm-identifier	O	O	O	O	
content-correlator	O	O	O	M	
content-identifier	O	O	O	M	
content-integrity-check	O	O	O	O	
content-length	O	O	O	M	
content-returned	O	O	O	M	
content-type	M	M	M	M	
conversion-with-loss-prohibited	O	O	O	M	
converted-eits	O	O	O	M	
creation-time	M	M	M	M	
delivered-eits	O	O	O	M	
delivery-flags	O	O	O	M	
dl-expansion-history	O	O	O	M	
entry-status	M	M	M	M	
entry-type	M	M	M	M	
intended-recipient-name	O	O	O	M	
message-delivery-envelope	M	M	M	M	
message-delivery-identifier	O	O	O	M	
message-delivery-time	O	O	O	M	
message-origin-authentication-check	O	O	O	O	
message-security-label	O	O	O	O	
message-submission-time	O	O	O	M	
message-token	O	O	O	O	
original-eits	O	O	O	M	
originator-certificate	O	O	O	O	
originator-name	O	O	O	M	
other-recipient-names	O	O	O	M	
parent-sequence-number	M	M	M	M	
per-recipient-report-delivery-fields	M	M	M	M	
priority	O	O	O	M	
proof-of-delivery-request	O	O	O	O	
redirection-history	O	O	O	M	
registration-indication	O	O	O	O	
report-delivery-envelope	M	M	M	M	
reporting-dl-name	O	O	O	O	
reporting-mta-certificate	O	O	O	O	

Message Store General Attribute Support					Part 2 of 2
	Support by:	IPM UA Rec	Bas MS Org	IPM MS Org	
Attribute	S				Comments/References
report-origin-authentication-check	O	O	O	O	
security-classification	O	O	O	O	
sequence-number	M	M	M	M	
subject-submission-identifier	M	M	M	M	
this-recipient-name	O	O	O	M	

Note: Enhanced MS support for optional Functional Groups is for further study. Attributes which are relevant to these areas are currently specified as Unsupported.

8.17.6 Message Store IPM Attribute Support

This section is to be read in accordance with Annex C of X.420 (1988).

Table 8.30. Classification of the Message Store IPM Attributes

Message Store IPM Attribute Support				Part 1 of 2
	Support by:	IPM UA	IPM MS	
Attribute	S	Rec	Org	Comments/References
Summary Attributes:				
ipm-entry-type	O	O	M	
ipm-synopsis	O	O	M	
Heading Attributes:				
authorizing-users	O	O	M	
auto-forwarded	O	O	M	
blind-copy-recipients	O	O	M	
copy-recipients	O	O	M	
expiry-time	O	O	M	
heading	M	M	M	
importance	O	O	M	
incomplete-copy	O	O	O	
languages	O	O	M	
nrn-requestors	O	O	M	
obsoleted-ipms	O	O	M	
originator	O	O	M	
primary-recipients	O	O	M	
related-ipms	O	O	M	
replied-to-ipm	O	O	M	
reply-recipients	O	O	M	
reply-requestors	O	O	M	
reply-time	O	O	M	
rn-requestors	O	O	M	
sensitivity	O	O	M	
subject	O	O	M	
this-ipm	M	M	M	
Body Attributes:				
bilaterally-defined-body-parts	O	O	O	
body	M	M	M	
encrypted-body-parts	O	O	O	
encrypted-data	O	O	O	
encrypted-parameters	O	O	O	
extended-body-part-types	O	O	O	

Message Store IPM Attribute Support				Part 2 of 2
	Support by:	IPM UA	IPM MS	
Attribute	S	Rec	Org	Comments/References
g3-facsimile-body-parts	O	O	O	
g3-facsimile-data	O	O	O	
g3-facsimile-parameters	O	O	O	
g4-class1-body-parts	O	O	O	
ia5-text-body-parts	O	O	M	
ia5-text-data	O	O	O	
ia5-text-parameters	O	O	O	
message-body-parts	O	O	M	
message-data	O	O	O	
message-parameters	O	O	O	
mixed-mode-body-parts	O	O	O	
nationally-defined-body-parts	O	O	O	
teletex-body-parts	O	O	O	
teletex-data	O	O	O	
teletex-parameters	O	O	O	
videotex-body-parts	O	O	O	
videotex-data	O	O	O	
videotex-parameters	O	O	O	
voice-body-parts	O	O	O	
voice-data	O	O	O	
voice-parameters	O	O	O	
Notification Attributes:				
acknowledgment-mode	O	O	M	
auto-forward-comment	O	O	M	
conversion-eits	O	O	M	
discard-reason	O	O	M	
ipm-preferred-recipient	O	O	M	
ipn-originator	O	O	M	
non-receipt-reason	O	O	M	
receipt-time	O	O	M	
returned-ipm	O	O	O	
subject-ipm	M	M	M	
suppl-receipt-info	O	O	O	

8.18 APPENDIX B: INTERPRETATION OF ELEMENTS OF SERVICE

See Working Document.

8.19 APPENDIX C: RECOMMENDED PRACTICES

This section provides guidelines on areas not addressed by the base standards. These guidelines have been produced in order to promote awareness of interim solution to problems as agree by members of the NIST X.400 SIG. However implementors of these recommended practices should note that it is not necessary to follow the recommended practices when claiming conformance to these agreements.

Implementors should also note that future standardization by CCITT and ISO/IEC on area covered by this section may result in different solutions to those proposed in this section.

8.19.1 Printable String

There are existing mail systems that include a small set of non-Printable String characters in their identifiers. For these systems to communicate with MHS systems, either for pass-through service or delivery to MHS users, gateways will be employed to encode these special characters into a sequence of Printable String characters. This conversion should be performed by the gateway according to a common scheme and before insertion in Domain Defined Attributes, which are intended to carry electronic mail identifiers. MHS UAs may also perform such conversions.

It is recommended that the following symmetrical encoding and decoding algorithm for non-Printable String characters be employed. The encoding algorithm maps an ASCII representation to a PrintableString representation. Any non-printable string characters not specified in table 8.31 are covered by the category "other."

Table 8.31. Printable string to ASCII mapping

ASCII Character	Printable String Character
% (percent)	(p)
@ (at sign)	(a)
! (exclamation)	(b)
" (quote mark)	(q)
_ (underline)	(u)
((left paren.)	(l)
) (right paren.)	(r)
other	(3DIGIT)

where 3DIGIT has the range 000 to 377 and is interpreted as the octal encoding of an ASCII character.

To encode an ASCII representation to a PrintableString, the table and the following algorithm should be used:

```
IF current character is in the encoding set THEN
  encode the character according to the table above
ELSE
  write the current character;
continue reading;
```

To decode a PrintableString representation to an ASCII representation, the table and the following algorithm should be used:

```
IF current character is not "(" THEN
  write character
ELSE
  {
  look ahead appropriate characters;
  IF composite characters are in the above table THEN
     decode per above table
  ELSE
  write current character;
  }
continue reading;
```

8.19.2 Rendition of IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations as defined in table 8.32.

Table 8.32. Interpretation of Format Effector Combinations

Combination	Interpretation
CR LF	to start a new line
CR FF	to start a new page (and line)
LF .. LF	to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect. The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition. Note that X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

8.19.3 EDI Use of MHS

8.19.3.1 Introduction and Scope

This section presents a carry-forward of the Recommended Practices in chapter 7 of the NIST Stable Implementation Agreements for support of EDI data transfer in an MHS(1988) environment. These recommended practices outline an interim procedure for use in transferring EDI transactions between trading partner applications in order to facilitate further MHS implementation by EDI users. It is the stated objective of the NIST X.400 SIG to migrate towards the CCITT target solution of P_{EDI} once it is defined (currently expected to be completed by late 1990). The approach for carrying EDI interchanges over MHS systems as recommended in this section provides a mechanism for a smooth migration to the target CCITT P_{EDI} solution.

The scope of this guideline is to describe specific recommended practices for extending MHS as a data transfer mechanism between EDI applications. This interim solution, as in the existing X.400(1984) Agreements, is referred to as the P0 approach and differs slightly from the European solution which uses a P2 mechanism to package the EDI data stream. However, if adhered to, P0 messages may be delivered to P2 recipients and vice-versa, by the MTA making a minor envelope conversion as recommended in this section.

8.19.3.2 Model

The model used is consistent with that defined in section 7.12.5 of the Agreements for X.400(1984) systems, with several minor exceptions. As before, the model provides for a peer-to-peer EDI Messaging (EDIM) UA service.

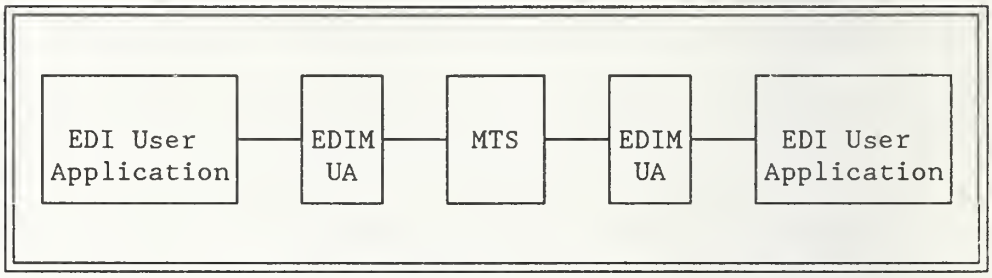


Figure 8.11. EDI Messaging Functional Model.

In figure 8.11, the EDIM UA may support delivery of EDI messages formatted according to the NIST P0(88), P0(84), P2(84) or P22(88) Agreements. It is recommended that implementations supporting EDI over MHS generate P0 formatted messages as described below, but be prepared to deliver any of the four recognized formats for which Agreements have been achieved. Whether the EDI transaction is carried using the P0 or P2 (88 or 84) approaches, the EDI content is restricted to be only one EDI interchange per MHS message. The NIST approach carries forward the 1984 X.400 recommended practices as a interim interworking solution. The use of object identifiers in either the EIT or Content Type fields is not recommended due to the current definition of the 1984 interworking rules specified in CCITT Recommendation X.419 (see below).

The EDIM UA must support the essential MT and MS Elements of Service as defined in this Agreement. It is recognized that a Message Store may not convey much information to a remote EDIM UA with the P0 approach. It is further recognized that MS Elements of Service are not necessarily made available by the EDIM UA to the EDI user application.

8.19.3.3 Protocol Elements Supported for EDI

The following P1 protocol elements will be used as a minimum to support EDI applications:

Content Type

For EDIM applications, the content type will continue to be as specified in section 7.12.5.3 of the NIST Stable Agreements for X.400(1984), i.e., Undefined (0). The interchange contained in the P1 envelope and identified by this Content Type is carried as an octet string.

Note: For interworking with 1984 X.400 systems, the use of an externally registered object identifier is not recommended. Such use would create interworking problems as well as loss of information when the X.419 downgrading rules are applied. In the current definition of the latter, an object identifier would be mapped to the Content Type value of 1 (External), instead of 0 (Undefined). Additionally, the downgrading rules require the object identifier value to be inserted into the Content, which would cause further interworking problems since 1984 EDIM UAs will expect only the EDI interchange in the Content.

Original Encoded Information Types (EITs)

EDIM applications will continue to use the EITs as specified in section 7.12.5.3 of the NIST Stable Agreements for X.400(1984), i.e., IA5Text and Undefined (for EBCDIC encoding). However, it is recognized that any EIT defined in the 1988 MHS standards may be used to specify the encodings of the EDI content.

Note: For interworking with 1984 X.400 systems, the use of an externally registered object identifier to signify the EDI encoding is not recommended. According to the current definition of the downgrading rules in CCITT Recommendation X.419, a 1988 MTA must downgrade an object identifier to Undefined and then discard the object identifier. Such loss of EDI encoding information conflicts with the existing Agreements for X.400(1984) systems that the semantics of the Undefined EIT is always EBCDIC.

8.19.3.4 Addressing and Routing

As in the Stable Implementation Agreements for X.400(1984), EDI messages entering a 1988 MHS environment will need to have MHS O/R Names and/or O/R Addresses in the P1 envelope to identify the originator and recipient trading partners. The mapping of the EDI originator and recipient interchange

addresses to an O/R Name or O/R Address may be achieved either by local means or through services provided by the OSI Directory (see sec. 8.9.1).

If the EDI message is originated and delivered without transiting a X.400(1984) MTA, then any of the O/R Address forms specified in this Agreement may be used. However, since it cannot be assumed that EDI messages will never transit an X.400(1984) MTA, it will often be useful to include additional Domain Defined Attributes as specified in section 7.12.5.4 of the Stable Agreements. This DDA is needed to ensure that a message is deliverable to an EDIM UA associated with a X.400(1984) MTA and that delivery and receipt reports can be flagged as a new report type.

8.19.4 Textual Representation of O/R Names

See Working Document.

8.20 APPENDIX D: LIST OF ASN.1 OBJECT IDENTIFIERS

See Working Document.

8.20.1 Content Types

See Working Document.

8.20.2 Body Part Types

See Working Document.

9. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 2

Editor's Note: In document type names, constraint set names, and abstract syntax definitions, the "NBS" designation will be preserved.

9.1 INTRODUCTION

This section defines Implementors' Agreements based on ISO File Transfer, Access and Management (FTAM), as defined in ISO 8571. This International Standard has four parts. Part 1 of the IS gives general concepts, Part 2 defines the Virtual Filestore (VFS), Part 3 defines the File Service, and Part 4 defines the File Protocol.

FTAM, as described in the IS, 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.

The general agreements reached with respect to the ISO File Transfer, Access and Management Protocol (FTAM) are that the Phase 2 FTAM specification (this sec.) is based on the International Standard (IS).

9.2 SCOPE AND FIELD OF APPLICATION

These FTAM Phase 2 Agreements cover transfer of and access to files between the Filestores of two end systems, including the management of a Virtual Filestore. One end system acts in the Initiator role and initiates the file transfer/access, while the other end system acts in the Responder role and provides access to the file in the Virtual Filestore. This paper describes Agreements for the actions and attributes of the Virtual Filestore, and the service provided by the file service provider to file service users, together with the necessary communications between the Initiator and Responder.

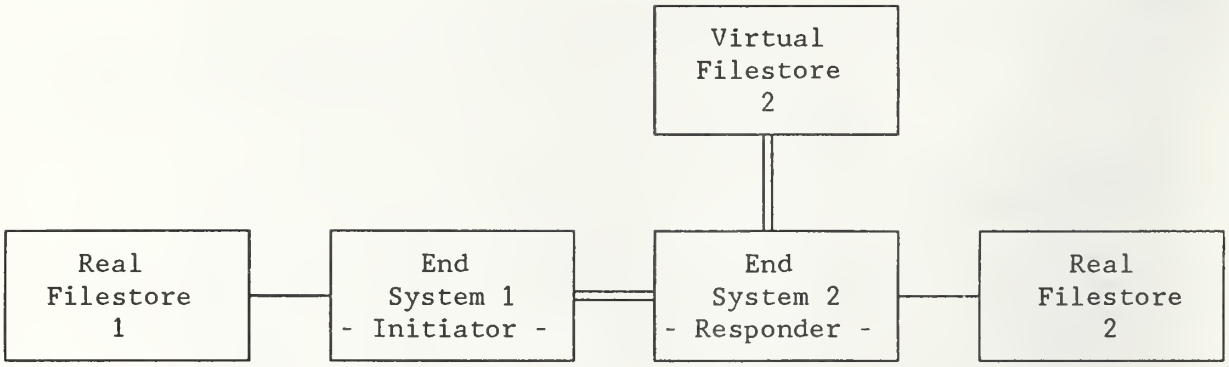


Figure 9.1. Model of file transfer/access.

Note: Agreements apply on the double lines of figure 1. The mapping between the Virtual Filestore and the Real Filestore together with the local data management system is not part of these Agreements.

These Agreements define General Agreements in section 9.5 through 9.16, minimum functionality (Conformant Implementations) in section 9.17, and functionality for several Implementation Profiles which are tailored to different classes of user requirements in sections 9.18 and 9.19.

9.3 STATUS

Version 2 of the FTAM Phase 2 Implementation Agreements was completed December 16, 1988, and published in version 2 of the Stable Implementation Agreements (NIST Special Publication 500-162, December 1988). Some editorial clarifications, technical changes and alignment changes (due to international harmonization of Profiles) were added to Version 2 Agreements in the course of 1989. All these changes are now fully incorporated in this version 3 of the FTAM Phase 2 Stable Agreements.

No further enhancements will be made to this version 3 of FTAM Phase 2 (see the next sec. ERRATA).

This version 3 of the FTAM Phase 2 Agreements fully replaces the version 2 as published in NIST SP 500-162. Therefore, the old version 2 of FTAM Phase 2 will no longer be maintained.

The following list summarizes the technical errata changes to FTAM Phase 2, Version 1 which were incorporated in Version 2. It also states the degree of possible interworking and backward compatibility to FTAM Phase 2, Version 1.

Technical Changes in FTAM Phase 2, Version 2 (Dec. '88)	Backward Compatibility to FTAM Phase 2, Version 1 (Dec. '87)
1. Check of already established presentation contexts for a document type not at CREATE time but at OPEN time	Interworking problems could occur when creating a document type which is not opened.
2. Receivers shall not require sending of AETtitles	Interworking problems could occur if AETitles are not sent
3. Minimum requirement for FADU identities for document types which use the sequential flat constraint set	Interworking problems could occur if FADU identities beyond the minimum requirement are used

The following list summarizes the technical errata and alignment changes which were incorporated in FTAM Phase 2, Version 3. It also states the degree of possible interworking and backward compatibility to FTAM Phase 2, Version 2.

Technical Changes in FTAM Phase 2, Version 3 (Dec. '89)	Backward Compatibility to FTAM Phase 2, Version 2 (Dec. '88)
1. Unconstrained Service Class outside the scope of the Implementation Profiles.	Profiles: Full compatibility, since change only from "optional" "to "outside scope"
2. <contents-type-list> parameter: Both types of values optional for Initiators	Full compatibility, since this clarification is only for Initiators
3. Specification of supported values for <override> parameter in F-CREATE	Interworking problems may occur, if different values supported
4. Parameters <filesize>, <future filesize>, <fadu-number> may be encoded with up to 8 contents octets	Interworking problems may occur, since no minimum requirement was defined for Version 2.
5. For NBS-7 and NBS-8 in conjunction with T or TM Service Class, the FADU identities "current," "next," "previous" are not required	Full compatibility, since these identities were never useful in conjunction with T or TM Service Class
6. For NBS-8 files the minimum range for keys of string-type is (1-16) instead of (0-16)	Interworking problems may occur for key-length parameter 0.
7. Restriction "NBS-9 files may not be Created or Deleted" removed from the document type definition. But both actions are outside the scope of the Agreements.	Full compatibility, since Creation, Deletion was never in the scope of the Agreements
8. Constraint set NBS-ordered-flat: The location after an Insert is "end"	Full compatibility, since this specification was already implicitly clear

9.4 ERRATA

Editor's Note: ERRATA applicability information may be given here.

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	SECTION	
CP 3/90-1	TECHNICAL	NIST-SP 500-177	APPENDICES A, B, C	NBS-6, NBS-7, NBS-8, NBS-9, NBS-AS1, NBS-AS2, NBS-ordered-flat constraint set with Obj. Id. using icd(9999) withdrawn. Above FTAM objects newly defined with Obj. Id. using oiw(14) ftamsig(5)
CP 3/90-2	EDITORIAL		9.17.9	Clarification that INTEGER encoding of more than 4 contents octets may be rejected by receiver
CP 3/90-3	EDITORIAL			Appendices instead of section 9.21, structuring into subsections. All references to these definitions updated accordingly
CP 3/90-4	EDITORIAL		9.3	New interworking table for Phase 2, Version 1 and Version 2
CP 6/90-1	EDITORIAL		9.1	Notes 7, 8 added to clarify the use of Escape Sequences

9.4 ERRATA

Editor's Note: ERRATA applicability information may be given here.

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	SECTION	
CP 3/90-1	TECHNICAL	NIST-SP 500-177	APPENDICES A, B, C	NBS-6, NBS-7, NBS-8, NBS-9, NBS-AS1, NBS-AS2, NBS-ordered-flat constraint set with Obj. Id. using icd(9999) withdrawn. Above FTAM objects newly defined with Obj. Id. using oiw(14) ftamsig(5)
CP 3/90-2	EDITORIAL		9.17.9	Clarification that INTEGER encoding of more than 4 contents octets may be rejected by receiver
CP 3/90-3	EDITORIAL			Appendices instead of section 9.21, structuring into subsections. All references to these definitions updated accordingly
CP 3/90-4	EDITORIAL		9.3	New interworking table for Phase 2, Version 1 and Version 2
CP 6/90-1	EDITORIAL		Table 9.1	Notes 7, 8 added to clarify the use of Escape Sequences

9.5 ASSUMPTIONS

1. FTAM protocol machines must be able to parse and process at a minimum 7K octets of FTAM PCI, FTAM structuring (FTAM-FADU) and FTAM user data (including grouped FPDUs) as they would be encoded with the ASN.1 Basic Encoding Rules. It is recommended, however, that Presentation user data not be restricted in size.

2. In order to maximize interoperability, it is important that the implementations of FTAM service providers do not unnecessarily restrict the service user's ability to generate arbitrary file service requests. Otherwise, they may not be able to work with FTAM Responders whose operation is constrained by their mapping of the FTAM Virtual Filestore to their local filestore. For example, error procedures should only be invoked when an error actually occurs, not at the point of the specification of options which might result in an error.

3. Implementations must be able to parse all valid optional parameters if they are present in the PDU. Only those optional parameters specified as supported in these Agreements are required to be implemented. If these parameters are not present, a default value is assigned locally. A Responder should not refuse a request solely because a parameter that is optional in the FTAM standard, but is supported in these Agreements, is not present.

4. Consideration of any standardized service interface is not covered by these Agreements.

5. These Agreements define no restrictions for the values used for the <communication quality of service> parameter in <F-INITIALIZE>.

6. FTAM is defined in phases. The Phase 1 FTAM implementation specification is based on the second ISO Draft Proposal, dated April 1985, and the ISO Draft Proposal 8824 and 8825.

   The Phase 2 FTAM specification (this sec.) is based on the International Standard (IS). THERE IS NO BACKWARD COMPATIBILITY WITH NIST FTAM PHASE 1. Backward compatibility is impossible, since Phase 1 uses Session services directly, while Phase 2 uses ACSE and Presentation services. Furthermore, there are differences in Filestore, PDU Abstract Syntax, FADU Abstract Syntax, and Transfer Syntax. There also are differences in the transparency mechanisms and service class negotiations.

   The <implementation information> parameter of <F-INITIALIZE> FPDU as defined in ISO 8571-4, 20.3 is used to pass 'user version' information with respect to different FTAM phases of the NIST Implementors Agreements or with respect to FTAM profiles of other bodies (see sec. 9.12 of this document). It is the goal of

these Agreements to use the 'user version' mechanism to provide at least one level of backward compatibility for all future NIST FTAM Phases, facilitating backward compatibility for future FTAM products, assuming different new versions of the respective IS's also enable backward compatibility.

7. The FTAM specific ASE requirements for ACSE in the Upper Layers chapter of this document define a value (that carries no semantics) for the AETitle that can be used by FTAM ASEs for communication. Other values for the AETitle are outside the scope of these Agreements.

   The association shall not be rejected/aborted if any of the following parameters either contains the defined value or is not sent:

   Called - AETitle
   Calling - AETitle
   Responding - AETitle

   The association may be rejected/aborted if any of these parameters contain other values.

   Use of values outside the scope of these Agreements is discouraged until agreed upon semantics have been associated with AETitles.

8. Use of <shared ASE information> parameter and <charging> parameter is not defined within the scope of the Agreements.

9. Use of <application context name> parameter is not defined within the scope of these Agreements. This parameter does not prohibit the establishment of an FTAM association.

10. These Agreements use the term 'supported' for a parameter to mean that the syntax and semantics of that parameter shall be implemented. However, it is not a requirement that the parameter be used in all instances of communication, unless stated otherwise.

    Also these Agreements use the term 'optionally supported' for a parameter to mean that it is left to the implementation whether the semantics of that parameter are implemented or not.

9.6 PRESENTATION AGREEMENTS

The following Abstract Syntaxes are recognized in these agreements:

"FTAM FADU"
"FTAM PCI"
"FTAM unstructured text abstract syntax"
"FTAM unstructured binary abstract syntax"
"NBS abstract syntax AS1"
"NBS file directory entry abstract syntax"

The following Transfer Syntax is supported:

"Basic Encoding of a single ASN.1 type"
(See Appendix C Part-3)

9.7 SERVICE CLASS AGREEMENTS

Implementation Agreements have been reached for the following service classes.

- File Transfer
- File Access
- File Management
- File Transfer and Management
- Unconstrained

9.8 FUNCTIONAL UNIT AGREEMENTS

Implementation agreements have been reached for the following functional units.

- Kernel
- Read
- Write
- File Access
- Limited File Management
- Enhanced File Management
- Grouping

Implementation of the Recovery, Restart Data Transfer, and FADU Locking functional units is not specified.

9.9 FILE ATTRIBUTE AGREEMENTS

Implementation of the Kernel Group of file attributes is defined. If the optional Storage Group and Security Group are implemented, aspects of their implementation are defined. Implementation of the Private Group is not specified.

Responses to an attribute value request shall always include one of the following (as specified in ISO 8571-2, Clause 9.4):

- An actual file attribute value.
- A value indicating that no value is available, optionally with a diagnostic.
- No value and an error code, optionally with a diagnostic indicating that the attribute is not supported.

9.9.1 Mandatory Group

Only the Kernel Group of attributes is required. A value for <filename>, <permitted actions>, and <contents type> will always be available.

A minimum range is required for <filename> values as specified in ISO 8571-2. No maximum length or format restrictions apply. A system that does not support <filename> values with a sequence of more than one Graphic String or extended <filename> characteristics may reject a request involving such a <filename>. All systems must be able to interpret a <filename> value with a sequence of one Graphic String.

A Responder shall not require an Initiator to use multiple component Graphic String filenames. Requests using a single component <filename> value with a sequence of one Graphic String shall be responded to using a single component <filename> value. Responses to requests involving <filename> values having two or more Graphic Strings are not defined here but may be interpreted via bilateral or other external agreements. Use of <filename> values with a sequence of more than one Graphic String is discouraged.

Apart from the minimum conformance requirements specified in ISO 8571-2, file names have to be specified in the naming convention of the responding FTAM implementation. It is a local implementation matter of the FTAM Responder, whether or not an additional name mapping onto the real FIlestore's file name convention is supported.

In order to enable interworking with all FTAM Responders' virtual Filestores, it is recommended that FTAM Initiators impose no restrictions on the attribute range supported for file names beyond those specified in ISO 8571-2.

For the purpose of interworking according to these Agreements the

<contents type> attribute is limited to the <document type name> format. The <constraint set name, abstract syntax name> form is outside the scope of these Agreements. It should always be parsed correctly when received, but may result in an error.

9.9.2 Optional Groups

If the optional Security Group of file attributes is implemented, an actual value must be available for the <access control> attribute.

The <access control> attribute is a SET OF <access control element>. The minimum requirement in these Agreements is the support of one <access control element>, according to the base standard. The terms <concurrency access>, <identity>, and <passwords> are each optionally supported. Details of their use shall be specified in the PICS. Use of the <location> term is not specified in these Agreements.

Implementation of the Private Group is not specified.

9.10 DOCUMENT TYPE AGREEMENTS

These document types are defined.

FTAM-1	"ISO FTAM unstructured text"
FTAM-2	"ISO FTAM sequential text"
FTAM-3	"ISO FTAM unstructured binary"
NBS-6	"NBS-6 FTAM sequential file"
NBS-7	"NBS-7 FTAM random access file"
NBS-8	"NBS-8 FTAM indexed file"
NBS-9	"NBS-9 FTAM file directory file"

Detailed document type definitions are given in Appendix A and in ISO 8571-2, Annex B.

Note: Document types NBS-1 to NBS-5 are not defined in these Agreements. The numbering starts with NBS-6 because of the original DIS version of these Agreements.

An implementation claiming conformance to these Agreements which also supports any or all of the document types FTAM-1, FTAM-2, and FTAM-3 as defined in ISO 8571-2, Annex B, must minimally support the combinations of parameter values as specified in table 9.1.

Table 9.1. Parameters for FTAM-1, -2, -3

	Universal Class Number	Maximum String Length[6,7]	String Significance
FTAM-1	General String [1](27) IA5String [2](22)	134 or less	'not-significant'[8]
FTAM-2	Graphic String [3,4](25)	134 or less[5]	'not-significant'[8]
FTAM-3	<not applicable>	512 or less	'not-significant[4]

Notes:

1. The minimum level of support for General String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 character sets, and ISO 646, IRV C0 set.

2. The support for IA5 String is the ISO 646, IRV G0 character set and the ISO 646, IRV C0 set.

3. The minimum level of support for Graphic String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 sets.

4. This is the default when the parameter is not specified.

5. The implementation need not support Data Units whose total character count exceeds 134.

6. As per table 9.3.

7. The length refers to the number of characters from the applicable character set. It does not include any escape sequences or overhead from the encoding.

8. If escape sequences are used in the encoding of the data and string boundaries are not maintained when the file is stored, it may be necessary for the escape sequences to occur at different locations when the file is re-sent.

For document types which use the sequential flat constraint set, conformant implementations must minimally support FADU identities as follows:

- for Transfer service class: 'begin', 'end'
- for Transfer and Management service class: 'begin', 'end'
- for Access service class: 'begin', 'end', 'first', 'next'

For the document types NBS-7 and NBS-8 used in conjunction with the Transfer service class or the Transfer and Management service class, the support of the FADU identities of 'current', 'next' and 'previous' and for NBS-8 the support of the FADU identity of 'end' are outside the scope of these Agreements.

For the document types NBS-6, NBS-7 and NBS-8 parameters are used for which the Agreements apply as specified in table 9.2.

Table 9.2. Parameters for NBS-6, NBS-7, NBS-8

Parameter	PrimType	String-length	Length-1	Length-2
int	INTEGER	Number of octets required to represent, in 2's complement format, the largest integer to be passed		
bit	BIT STRING	Number of bits in string (non-varying)		
ia5	IA5String	Max number of characters in string		
graphic	Graphic String	Max number of characters in string		
general	General String	Max number of characters in string		
octet	OCTET STRING	Max numbers of octets in string		
private-class-number	Floating Point Number		The minimum number of bits required to be maintained in the mantissa for relative precision	Number of bits required to represent the largest unbiased integer exponent in 2's complement
univer-time	UTCTime	<not applicable>		
gen-time	Generalized Time	<not applicable>		
boolean	BOOLEAN	<not applicable>		
null	NULL	<not applicable>		

Note: The string length parameter specifies the actual number of from the referenced character set. It does not include any escape sequences or overhead from the encoding.

The primitive data types and minimal size ranges that an implementation must accept for storage are given in table 9.3.

Table 9.3. FTAM primitive data types

Primitive Data Type		Minimum Range (Octets)
ASN.1	INTEGER	1 - 2
ASN.1	BIT STRING	0 - 1
ASN.1	IA5String	0 - 134
ASN.1	GeneralString	0 - 134
ASN.1	GraphicString	0 - 134
ASN.1	OCTET STRING	0 - 512
ASN.1	BOOLEAN	
ASN.1	NULL	
ASN.1	GeneralizedTime	
ASN.1	UniversalTime	
NBS-AS1	FloatingPointNumber	mantissa 1-23 bits exponent 0-8 bits

Notes:

1. The primitive data types and their maximum ranges for a specific file as described by the parameters above are maintained in the <contents type> file attribute. The <contents type> file attribute value is established at the file's creation and cannot be changed via FTAM for the life of the file. This implies that the data element types and ranges and data unit formats are fixed for all accessors of that file as long as the file exists.

2. The syntax for floating point numbers is part of the definition of NBS abstract syntax AS1 in Appendix C.3 Annex-9A-Part-3. It is derived from existing standards IEC 559 and IEEE 754.

9.10.1 Character Sets

Implementation of a character set in FTAM is understood as:

- a transfer syntax is defined for the character set
- document types are defined using the character set in their abstract syntactic definition

- o documents of those types are stored in the Virtual File Store as defined in the character set specification. They are written into the VFS and read from the VFS as defined by the abstract syntax and the transfer syntax for the document type. It is not in the scope of FTAM Agreements to specify the local representation of those documents in the Real Filestore, nor to specify rendition of graphic characters or control characters on character imaging devices. These renditions are possible agreements between applications using FTAM for their communication.

The character sets ISO 646, IRV and ISO 8859-1 shall always be implemented.

9.10.1.1 ISO 646 Character Set

The International Reference Version (IRV) of ISO 646 is available for use when there is no requirement to use a national or an application-oriented version. In information interchange, the IRV is assumed unless a particular agreement exists between sender and receiver of the data. The graphic characters allocated to the IRV are as specified in table 9.4.

Table 9.4. IRV Graphic Character Allocations

Graphic	Name	Coded Representation
#	Number sign	2/3
o	Currency sign	2/4
@	Commercial at	4/0
[	Left square bracket	5/11
\	Reverse solidus	5/12
]	Right square bracket	5/13
^	Circumflex accent	5/14
'	Grave accent	6/0
{	Left curly bracket	7/11
|	Vertical line	7/12
}	Right curly bracket	7/13
~	Tilde, overline	7/14

It should be noted that no substitution is allowed when using the IRV and that the facility of combined vertical and horizontal movements of the active position does not apply to any format effectors.

It is permitted to use composite graphic characters and there is no limit to their number. Because of this freedom, their processing and imaging may cause difficulties at the receiving end. Therefore agreement between sender and receiver of the data is recommended if composite characters are used.

Note: Attention is drawn to the fact that different national character sets exist.

(See ISO 646 or CCITT Recommendation T.50 for more information)

9.10.1.2 Format Effectors

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 the single format effector for effecting the combined vertical and horizontal movement.

Notes:

1. For further information see ISO 646:1983, clauses 4.1.22 and 6.4, ISO 6429:1988, clause E.1.2 and ISO 4873:1986, clause A.3.2.

2. The Agreements require only support of C0 control characters of ISO 646, containing among others the format effectors <CR> and <LF>. It is recommended that NIST implementations use <CR> <LF> pairs as line terminators.

9.10.1.3 8859-1 Character Set

The Latin Alphabet No.1 (ISO 8859-1) is used to specify the printable characters of G0 and G1. C0 control characters and their associated rules are taken from the ISO 646 definition.

9.10.2 Document Type Negotiation Rules

9.10.2.1 Connection Establishment

In connection establishment the <contents type list> parameter is used only to establish presentation contexts. Both the <document type name> form and the <abstract syntax name> form are supported for responders; they are optionally supported for initiators.

9.10.2.2 File Creation

An <F-CREATE request> FPDU must contain a <document type name> value in its <initial attributes> parameter.

If the specified document type requires parameterization, then these parameters must be supplied, otherwise the <F-CREATE request> may be rejected.

Notes:

1. It is understood that <permitted actions> sub-field of <initial attributes> parameter will always be used at <F-CREATE request>. The value may be changed by the Responder.

2. If the <document type name> used requires DU syntax parameters and one of the parameters specifies 'FloatingPointNumber' as a primitive data type, the request may be rejected, in case the optional type 'FloatingPointNumber' is not supported by the Responder.

9.10.2.3 File Opening

The <document type name> form (with appropriate parameters as specified in 8871-2, Clause 12.3) shall always be used when proposing a <contents type>; as an alternative the 'ContentsTypeUnknown' value may be used in the <F-OPEN request>. An <F-OPEN response> shall use the <document type name> option (with appropriate parameters) in the <contents type> field.

This allows the receiving entity to use the <document type name> attributed to the file instead of receiving a <constraint set name> and <abstract syntax name> pair, which does not reflect the file information contained in the FTAM and NBS document types.

This document type name is either a value from the set of base document type names as negotiated upon connection establishment or a document type name, for which an appropriate presentation context was established.

Notes:

1. An <F-OPEN response> without a <document type name> (but carrying the <constraint set name> and <abstract syntax name> form) may cause the Initiator to issue an <F-CLOSE request>.

2. If the <document type name> used requires DU syntax parameters and one of the parameters specifies 'FloatingPointNumber' as a primitive data type, the request may be rejected, in case the optional type 'FloatingPointNumber' is not supported by the Responder.

9.10.3 Relationship Between DUs, DEs and Document Types

"Abstract Syntax" is used to refer to the syntactic information which is architecturally passed between the Application and Presentation Layers. The Abstract Syntax defines Data Element (DE) types which are not necessarily ASN.1 primitive types. A Data Element (DE) is the smallest piece of data whose identity is necessarily preserved by the Presentation Service. Data types may be made up of other data types. Data Elements are not defined in terms of other Data Elements.

A Data Unit (DU) is a sequence of one or more Data Elements. Architecturally, entire, single DEs are passed into and out of the application process. In a real implementation, DUs may be passed.

To maintain DU boundaries during transfer, file structuring information must be passed (ISO8571-FADU definition in ISO 8571-2, Clause 7.5). A Data Element is referred to as a File-Contents- Data-Element in the ISO8571-FADU definition.

Document types refer to aspects of local processing and storage. They describe:

- o structural relationship between DUs,
- o structure of DUs, called DU syntax, and
- o DE types found in the file.

Because document types pertain to local processing and storage, the DU syntax makes assertions about the syntax and the size of DUs (records) in storage. Parameters on the document types provide this information about the syntax and size of the DUs.

9.11 F-CANCEL ACTION

When an F-CANCEL is sent or received, the following occurs:

- o no more data is sent,
- o checkpoint numbers are removed, and
- o state of the file is implementation dependent.

Note: When mapping F-CANCEL on P-RESYNCHRONIZE (abandon) it is required that P-SYNC-MINOR be used after F-READ/F-WRITE (see ISO 8571-4 Clauses 13, 14).

9.12 IMPLEMENTATION INFORMATION AGREEMENTS

- The <implementation information> parameter of <F-INITIALIZE> FPDU is not required by these Agreements.

- It may be used to pass user version information as a series of values, separated by ';'.

- The following will indicate conformance to the NIST Phase 2 Agreements: NBS-Phase2.

  Note: The list of possible values may be enlarged for future FTAM phases or FTAM profiles of other bodies.

- This parameter is for information only; it is not used for negotiation.

  The establishment of an FTAM regime should not be rejected only because of an unknown <implementation information> value.

9.13 DIAGNOSTIC AGREEMENTS

- The <diagnostic> parameter is supported; a value in the <response> PDU is needed when the <action result> or <state result> is not zero. (The nature of these agreements is to provide <diagnostic> information when any result parameter is not 'success'.)

- General catch-all diagnostic action is discouraged.

- The <further details> subfield is supported. It will be encoded as GraphicString, but is restricted to ISO 646 (IRV, graphic characters) and ISO 8859-1 only.

- Use of F-P-ABORT for other than protocol errors and catastrophic situations is discouraged.

- When returning an error status in a file management related diagnostic (i.e.,, <F-READ-ATTRIBUTE response> or <F-CHANGE-ATTRIBUTE response>), identify the erroneous attribute by using the first two characters of <further details> to hold a 2-digit number (encoded as IA5String) from the <F-READ-ATTRIBUTE request> attributes abstract syntax definition (ISO 8571-4, Clause 20.3).

00 Filename
01 Permitted Actions
02 Contents Type
03 Storage Account
04 Date and Time of Creation
05 Date and Time of Last Modification
06 Date and Time of Last Read Access
07 Date and Time of Last Attribute Modification
08 Identity of Creator
09 Identity of Last Modifier
10 Identity of Last Reader
11 Identity of Last Attribute Modifier
12 File Availability
13 File Size
14 Future Filesize
15 Access Control
16 Legal Qualifications
17 Private Use

The set of file management diagnostics, found in ISO 8571-3 Annex A, must be supported.

- In the case where a specific parameter can in no way be accommodated then the request fails and a <diagnostic> indicating one such parameter should be returned by the responder. In the case where a negotiable parameter cannot be accommodated with exactly the value requested but is negotiated to a different value (as defined in the standard) then the request formally succeeds but informative <diagnostics> indicating those parameters negotiated should be returned.

- In order to provide for robust applications using FTAM, well defined and precise diagnostics are required to be returned by responding implementations whenever an action cannot be carried out precisely as requested with respect to non-negotiable parameters. All such applicable diagnostics will be returned in those cases. An action is carried out precisely as requested with respect to a parameter when the value of that parameter on the <request> FPDU is equal to the value in effect during or subsequent to the action, depending on whether the action is regime control.

Diagnostics exist to signal 'parameter not supported' and Responder implementations shall issue all appropriate diagnostics. The <further details> subfield of the <diagnostic> parameter shall specify the parameter which is not implemented.

9.14 CONCURRENCY

The <concurrency control> used by default on actions requested by an <F-SELECT indication> or <F-CREATE indication> service are:

'shared' for read and read attribute

'exclusive' for all other actions

The default for actions not requested is specified as 'not required' as per ISO 8571-3.

Note: A local implementation may choose to be more restrictive in order to assure file consistency for concurrent accessors.

FADU locking is not required.

9.15 REQUESTED ACCESS

The <requested access> parameter on <F-SELECT> or <F-CREATE> is used to specify the actions which the Initiator may perform during the file selection. The value of the <requested access> parameter is compared by the Responder to the <access control> and <permitted actions> file attributes and concurrency controls (including those requested by the Initiator) currently in place on the file. If the value of the <requested access> parameter is not consistent with either <access control>, <permitted actions>, or concurrency controls in place, then the <F-SELECT> or <F-CREATE> must be rejected.

<requested access> is consistent with <access control> if, for each action requested, that action either requires no password, or the required password has been specified on the <F-SELECT request> or <F-CREATE request>.

<requested access> is consistent with <permitted actions> if, for each action requested, that action is allowed by the <permitted actions> file attribute.

<requested access> is consistent with <concurrency control> requested on the <F-SELECT> or <F-CREATE> if, for each action requested, that action has not been specified as 'not required' or 'no access' in the <concurrency control> parameter.

<requested access> is consistent with concurrency controls in place on the file if for each action requested no other accessor of the file has set the concurrency control for that action to either 'exclusive' or 'no access'.

9.16 SECURITY

9.16.1 Initiator Identity and Filestore Password

The <initiator identity> and <filestore password> parameters for an implementation acting as an Initiator are supported. These parameters are optional for an implementation acting as a Responder.

The syntax of <initiator identity> and <filestore password> is system-dependent. <initiator identity> and <filestore password> will represent account information on the local system, which may be different from the <account> parameter.

9.16.2 Access Passwords

The <access passwords> and <create password> parameters for an implementation acting as an Initiator are supported if the Security Group of attributes is supported. These parameters for an implementation acting as a Responder are optionally supported if the Security Group is supported.

9.16.3 Implementation Responsibilities

It is the responsibility of each local system to provide security for its own real filestore. Encryption of passwords will not be done by FTAM.

A user of the file service must be known by the Responder. "Known" is defined by the local Filestore, and is dependent on the level of security provided by the local Filestore.

9.17 REQUIREMENT FOR CONFORMANT IMPLEMENTATIONS

This section gives the criteria to be satisfied by every implementation of FTAM that conforms to these Agreements.

Conformance to these Agreements is stated in terms of the different roles occupied by FTAM implementations. The interoperability of certain configurations of these roles motivates this approach. Interoperable configurations of these roles are given in section 9.17.1.

The only function provided by every conformant implementation is the transfer of unstructured binary files in their entirety. It must be recognized that such simple transfer, while commonly understood and generally important, will not support all applications of FTAM. section 9.18 defines Implementation Profiles of FTAM services and protocol that can provide other specific functions. Those other

functions exploit the access and management capabilities of FTAM. The unconstrained service class (with appropriately chosen functional units) can be used to provide the functions of any of the Implementation Profiles. Users of FTAM must consider carefully what functions they require. They must examine all the Implementation Profiles and select according to their needs.

Implementation conforming to these Agreements require adherence to the General Agreements in secs. 9.5 through 9.16 of these Agreements.

9.17.1 Interoperable Configurations

Any implementation conforming to this specification must be able to act in at least one of the following role combinations:

1. initiator and receiver,

2. initiator and sender,

3. responder and sender,

4. responder and receiver.

Minimal implementations of combination 1 will interoperate with minimal implementations of combination 3. Minimal implementations of combination 2 will interoperate with minimal implementations of combination 4.

Any implementations of roles 1 and 3 will be able to interoperate at the intersection of their capabilities (which will be at least the minimal capabilities described in secs. 9.17.3 to 9.17.8). Any implementations of roles 2 and 4 will be able to interoperate at the intersection of their capabilities (which will be at least the minimal capabilities described in sects. 9.17.3 to 9.17.8).

These role combinations and this interoperability are shown in table 9.5 below.

Table 9.5. Interoperable configurations

		Initiator		Responder	
		sender	receiver	sender	receiver
Initiator	sender				x
	receiver			x	
Responder	sender		x		
	receiver	x			

9.17.2 Relationship to ISO 8571--The FTAM Standard

Any implementation in conformance to ISO 8571 (as defined in ISO 8571-4, Clause 22 (Conformance)), in addition to the implementation of the minimal protocols and roles enumerated in sections 9.17.3 to 9.17.8, is considered to be in conformance with these Agreements. Any implementation violating any of the conformance statements in ISO 8571-4 is considered to be in violation of these Agreements.

9.17.3 Requirements for Document Type Support

The document type FTAM-3 shall be supported for purposes of transfer and storage. The details regarding support for FTAM-3 in the FTAM dialogue are given in section 9.10.

Support of document types other than FTAM-3 is not required for conformant implementations. Support for document types described in these Agreements also entails support for:

- the semantics given in their description and further qualified in 9.10
- the preferred transfer syntax "Basic Encoding of a single ASN.1 type"

9.17.4 Initiators

Every implementation of an FTAM Initiator shall support:

- the kernel protocol and its mandatory parameters with

minimum ranges [Minimum required ranges are specified in sec. 9.17.8.],

- the grouping protocol and the <threshold> parameter with a value of at least 2 for use in the file transfer class,

- at least one of the read or write protocols [Specific conformance for reading and writing is defined in secs. 9.17.6 and 9.17.7.],

and support the applicable procedures defined in ISO 8571-4 clauses 8.1 (FTAM regime establishment), 8.2 (FTAM regime termination), 8.3 (File selection), 8.4 (File deselection), 8.9 (File open), 8.10 (File close), 8.11 (Begin group), 8.12 (End group), and 10 (File general actions). To support the above protocols and procedures the implementation shall always support the kernel functional unit and additionally shall be able to:

- request the grouping and at least one of the read or write functional units,

- request the file transfer class with the <service class> parameter,

- request the document type FTAM-3 using the <document type name> form of the <contents type> parameter,

- request the <FTAM quality of service> parameter with value 0 and accept in all cases the returned value 0, and

- request a <communication quality of service> consistent with the transport definition in these Agreements

as part of the Filestore initialization procedures in ISO 8571-4 clause 8.1, FTAM regime establishment.

Initiators must be able to operate under all circumstances if the above minimum values are successfully negotiated and returned on an <F-INITIALIZE response> PDU. Initiators must be able to operate with any downward negotiation of requested parameter values as described in the standard.

Should the supporting services break down, such that FTAM communication is impossible, the FTAM protocol machine shall notify the user with an <F-P-ABORT indication> and <diagnostic> value with identifier 1011, as well as any known <further details>.

Note: Interworking may not be possible between Initiators not supporting attributes of the Storage Group and Security

Group, and Responders requiring these attributes to be used.

9.17.5 Responders

Every implementation of an FTAM Responder shall support:

- the kernel protocol and its mandatory parameters with minimum ranges [Minimum required ranges are specified in sec. 9.17.8.],
- the grouping protocol and the <threshold> parameter with a value of at least 2 for use in the file transfer class,
- at least one of the read or write protocols [Specific conformance for reading and writing is defined in sections 9.17.6 and 9.17.7],

and support the applicable procedures, defined in ISO 8571-4 clauses 9.1 (FTAM regime establishment), 9.2 (FTAM regime termination), 9.3 (File selection), 9.4 (File deselection), 9.9 (File open), 9.10 (File close), 9.11 (Begin group), 9.12 (End group), and 10 (File general actions). To support the above protocols and procedures the implementation shall always support the kernel functional unit and additionally shall be able to:

- accept requests for the grouping and at least one of the read or write functional units,
- accept requests for the file transfer class with the <service class> parameter,
- accept the document type FTAM-3 using the <document type name> form of the <contents type> parameter,
- accept requests for an <FTAM quality of service> parameter with any value but may respond with the value 0, and
- accept requests for a <communication quality of service> consistent with the transport definition in these agreements

as part of the filestore initialization procedures in ISO 8571-4 clause 9.1, FTAM regime establishment.

Responders must be able to operate under all circumstances if the above minimum values are requested on an <F-INITIALIZE request> PDU. Responders must not negotiate upward in the sense described in the standard.

Responders must complete each action requested and supported in a manner consistent with its description in ISO 8571-2 Clauses 10 (Actions on complete files) and 11 (Actions for file access), and must interpret each supported attribute in a manner consistent with its definition in ISO 8571-2 Clause 12 (File attributes).

Under circumstances where actions cannot be carried out either as requested or consistently with ISO 8571-2 Clause 10 (Actions on complete files) and 12 (Actions for file access), the Responder must return at least one diagnostic indicating:

- o if the failure was due to either a protocol or Filestore failure, and then:
  - -- precisely which action failed,
  - -- at least one of the parameters that could not be accommodated with the diagnostic type indicating at least the degree of failure, as given by the action and state result parameter, or
- o that the failure was due to unforeseen system shutdown.

Should the supporting services break down, such that FTAM communication is impossible, the FTAM protocol machine shall notify the user with an <F-P-ABORT indication> and <diagnostic> with identifier 1011, as well as inform the user of any known <further details>.

9.17.6 Senders

Every implementation of an FTAM sender shall support the read functional unit as Responder or the write functional unit as Initiator, and support the applicable procedures defined in ISO 8571-4 Clauses 11 (State of the bulk data transfer activity), 12 (Bulk data transfer protocol data units), 15 (Bulk data transfer sending entity actions), 17.1 (Discarding), and 17.2 (Cancel).

To support those procedures the implementation shall be able to send files of the document type FTAM-3 and shall be able to send them as user data in PPDUs in blocks of up to 7168 octets.

9.17.6.1 Initiator Senders

Every implementation of an FTAM sender which is also an FTAM Initiator shall support:

- o the write functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU operation	replace
FADU identity	first

and support the applicable procedures, defined in ISO 8571-4 Clause 13 (Bulk data transfer initiating entity actions).

9.17.6.2 Responder Senders

Every implementation of an FTAM sender which is also an FTAM Responder shall support:

- o the read functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU identity	first
Access context	UA

and support the applicable procedures, defined in ISO 8571-4 Clause 14 (Bulk data transfer responding entity actions).

9.17.7 Receivers

Every implementation of an FTAM receiver shall support the read functional unit as Initiator or the write functional unit as Responder, and support the applicable procedures, defined in ISO 8571-4 Clauses 11 (State of the bulk data transfer activity), 12 (Bulk data transfer protocol data units), 16 (Bulk data transfer receiving entity actions), 17.1 (Discarding), and 17.2 (Cancel).

To support those procedures the implementation shall be able to receive files of the document type FTAM-3 and shall be able to receive them as user data in PPDUs in blocks of at least 7168 octets.

9.17.7.1 Initiator Receivers

Every implementation of an FTAM receiver which is also an FTAM Initiator shall support:

- o the read functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU identity	first
Access context	UA

and support the applicable procedures, defined in ISO 8571-4 Clause 13 (Bulk data transfer initiating entity actions).

9.17.7.2 Responder Receivers

Every implementation of an FTAM receiver which is also an FTAM Responder shall support:

- o the write functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU operation	replace
FADU identity	first

and support the applicable procedures, defined in ISO 8571-4 Clause 14 (Bulk data transfer responding entity actions).

9.17.8 Minimum Ranges

Any implementation of any conformant FTAM configuration shall be able to receive and meaningfully process all mandatory parameters for all functional units supported as well as the <diagnostic> parameter within at least the minimum ranges of values given in table 9.6. A conforming implementation may support a wider range of values for any parameter.

Table 9.6. Required minimal parameter support

Parameter		Minimum Range
general	diagnostic	Values as specified in ISO 8571-3 Annex A (Diagnostic parameter values) tables 44, 45 and 46 which correspond directly to mandatory parameters.
	action result	All values.
	state result	All values.
F_INITIALIZE		
	functional units[1]	'read' (for initiator/receivers and responder/senders) and 'grouping' or 'write' (for initiator/senders and responder/receivers) and 'grouping'
	presentation context management[2]	'False'
	all others	As specified in sections 9.17.4 and 9.17.5 above.
F_SELECT		
	attributes	Only <filename> is used with a minimum supportable length of 8 characters. Any other attribute supported for other services must have minimum supported lengths as in ISO 8571-2 Clause 15 (Minimum attribute ranges) table 2.
	requested access	'read' for initiator/receivers 'read' for responder/senders 'replace' for initiator/senders 'replace' for responder/receivers
F-CREATE		
	override	For Responders, the values 'create-failure', 'select-old-file' and 'delete-and-create-with-new-attributes' are supported. The value 'delete-and-recreate-with-old-attributes' is optionally supported. All values are optional for Initiators.

(Continued on next page.)

Table 9.6. Required minimal parameter support, continued

Parameter	Minimum Range
F_OPEN	
processing mode	'read' for initiator/receivers 'read' for responder/senders 'replace' for initiator/senders 'replace' for responder/receivers
contents type	'FTAM-3'
F_READ	
FADU identity	'first'
access context	'UA'
F_WRITE	
FADU operation	'read' for initiator/receivers 'read' for responder/senders 'replace' for initiator/senders 'replace' for responder/receivers
FADU identity	'first'
F_BEGIN_GROUP	
threshold[3]	For file transfer (a minimal required function) 2.

For any other supported parameters, minimum ranges are taken from the minimum ranges for the attribute corresponding to each as in ISO 8571-2 table 4.

Notes: 1. The parameters, functional units, and presentation context management are not ordered, so "minimum value" cannot be formally defined. The above values are those required for conformance to these Agreements but no value conformant to ISO 8571 for use in other applications is regarded to be in violation of these Agreements.

2. Other functional units (and service classes) for defined implementations may also be valid provided that they are implemented in accordance with these Agreements, specifically section 9.17.8.

3. Every implementation must support the <threshold> value 2 to provide the basic required function of file transfer; any other value in other applications is acceptable.

9.17.9 Range of Values for INTEGER Type Parameters

The general range for parameters of type INTEGER to the FTAM PCI is as specified in the UNIVERSAL ASN.1 ENCODING RULES section of the Upper Layers chapter.

The parameters

FTAM Attributes
filesize
future-filesize

FADU-Identity
fadu-number

may be encoded so that the length of its contents octets is no more than eight octets.

In the case of receiving more than 4 contents octets, the receiver may reject the corresponding FTAM PDU.

Note: To guarantee interworking, encoding should be restricted to the range -2^{31} to $2^{31}-1$.

9.17.10 Use of Lower Layer Services

- Support for the Presentation Context Management functional unit is not required.

- Implementations will support the Session, Presentation, and ACSE requirements as stated in section 5 of this document.

Note: Implementation of the Session Resychronize and the Minor Synchronize functional units is highly recommended, since the F-CANCEL service may be less effective when mapped to S-DATA.

9.18 IMPLEMENTATION PROFILES

This section defines Implementation Profiles for the specific functions of:

- File Transfer
- File Access
- File Management.

Those definitions are expressed in terms of:

- Document Types
- Attributes

- Service Classes (both service elements and their parameters).

This by no means defines all possible Implementation Profiles. The following Implementation Profiles are defined:

T1: Simple File Transfer

T2: Positional File Transfer
T3: Full File Transfer
A1: Simple File Access
A2: Full File Access
M1: Management.

Implementation Agreements have been reached for the following service classes. Note that any given implementation may support more than one service class.

- File Transfer
- File Access
- File Management
- Unconstrained
- File Transfer and Management

Support of an Implementation Profile requires adherence to:

1. corresponding definition in 8571-3 Clause 8 and any related procedures in 8571-4 Clause 8-17,

2. requirements given in sections 9.5-9.18 of these Agreements, and

3. requirements for parameter and attribute support as defined in section 9.17.8.

9.18.1 General Requirements for the Defined Implementation Profiles

- Implementations will be able to act either as Initiator or Responder or both.

- Implementations must support diagnostics as described in section 9.13 of these Agreements.

- Implementations that support the file access service class will support access to sequential files. Support of sequential files entails hierarchy of depth and arc length equal to 1. Other hierarchy depth and arc lengths are not precluded by these agreements.

9.18.2 (deleted)

9.18.3 Document Type Requirements for the Defined Implementation Profiles

Implementations conformant to Implementation Profiles defined in table 9.7 will support the following document types with the caveats and procedures given. Those document types are defined

in Appendix 9A and section 9.10 of these Agreements, and in ISO 8571-2.

- o FTAM-1
- o FTAM-2
- o FTAM-3
- o NBS-6
- o NBS-7

  Note: Support of this document type entails the naming of FADUs by their position in preorder traversal.

  Caveat: Other methods of naming FADUs depend on the system, application, and specific file, and as such are not described here.
- o NBS-8
- o NBS-9

Support for any document type requires the ability to transfer and store the abstract syntax given in its definition. These Agreements do not specify techniques or formats for storage.

Caveat: Specific abstract syntaxes for the parameterized document types NBS-6,7,8 are not specified in these Agreements.

Any document type supported must be identifiable by its document type name as given in ISO 8571-2 and in Appendix 9A of these Agreements and, where defined, the parameterization scheme given in section 9.10 of these Agreements.

For conformance to NBS-9 a Responder is only required to return the <filename> attribute, subject to local security and access control. All other requested attributes need not be returned.

Systems supporting the NBS-9 document type shall make available an NBS-9 document called 'DIRLIS'. This document can be used to obtain a listing of files and their associated attributes from a remote Filestore.

Creation and deletion of NBS-9 files are outside the scope of these Agreements.

File security issues related to NBS-9 are subject to the security agreements outlined in section 9.16.

9.18.4 Parameters for the Defined Implementation Profiles

- Implementations will support the <contents type list> parameter on the <F-INITIALIZE> service element. The initiating service must supply a value for this parameter.

- Implementations will support the <diagnostic> parameter as stated in section 9.13 of these Agreements.

- The <initiator identity> parameter is supported. Use must be consistent with section 9.16 of these Agreements.

- Implementations are not precluded from using other parameters for security and/or accounting. Respondersmust state the syntax and the semantics applying to <account> and <charging> parameters. The Responder's minimum implementation is to accept but ignore the <account>.

9.18.5 Parameter Ranges for the Defined Implementation Profiles

Parameter ranges for Implementations Profiles are as stated for primitive data types in section 9.10 of these agreements.

9.18.6 File Attribute Support for Implementations

Implementations of the Implementation Profiles will support file attributes or attribute groups in the following ways.

- mandatory

  This feature is mandatory in the ISO 8571-2 standard and shall therefore be implemented by all implementations claiming conformance to these Agreements.

- supported

  This feature shall be implemented by all implementations claiming conformance to these Agreements (for attributes, this implies that at least the minimum range of attribute values, as defined in ISO 8571-2 Clause 15, shall be supported). Conformant implementations shall also be able to interwork with other implementations that do not support this feature by negotiating out the corresponding features.

- optionally supported

Implementations claiming conformance to these Agreements may or may not implement this feature (for attributes, this implies that at least either the minimum range of attribute values, as defined in ISO 8571-2 Clause 15, shall be supported or that the 'no value available' result shall be supplied). If an attribute group with a support level of 'optionally supported' is chosen to be supported, then all the attributes of this group that are classified as 'mandatory' or 'supported' shall be supported.

- not supported

  This feature is outside the scope of these Agreements.

Kernel Group	mandatory
o Filename	mandatory
o Permitted Actions	mandatory
o Contents Type	mandatory
Storage Group	optionally supported
o Storage Account	optionally supported
o Date and Time of Creation	optionally supported
o Date and Time of Last Modification	optionally supported
o Date and Time of Last Read Access	optionally supported
o Date and Time of Last Attribute Modification	optionally supported
o Identity of Creator	optionally supported
o Identity of Last Modifier	optionally supported
o Identity of Last Reader	optionally supported
o Identity of Last Attribute Modifier	optionally supported
o File Availability	supported
o Filesize	supported
o Future Filesize	optionally supported
Security Group	optionally supported
o Access Control	supported
o Legal Qualifications	optionally supported
Private Group	not supported

Table 9.7. Implementation profile support requirements

Functional Unit	Service Class (See Note 8) T	M	A	T&M	UNCST
Kernel	T1,T2,T3	M1	A1,A2		
Read (See note 3.)	T1,T2,T3		A1,A2		
Write (See note 3.)	T1,T2,T3		A1,A2		
Limited File Mgmnt.	SeeNote 6	M1	SeeNote 6	See	See
Enhanced File Mgmnt.		M1			
Grouping	T1,T2,T3	M1			
File Access			A1,A2		
Document Types					
FTAM-1	T1,T2,T3	[M1]	A1,A2		
FTAM-2	T2,T3	[M1]	A1,A2		
FTAM-3	T1,T2,T3	[M1]	A1,A2	Note	Note
NBS-6	[T2],T3	[M1]	[A1],A2	4	5
NBS-7	[T2],T3	[M1]	[A1],A2		
NBS-8	T3	[M1]	A2		
NBS-9	[T1],[T2] [T3]	[M1]			

Notes: to 9.18.3 and table 9.7

1. The Management Implementation Profile is only to be implemented in conjunction with one of the Transfer or Access Profiles.

2. Profile T2 is subset of T3. A1 and T1 are subsets of A2 and T2, respectively.

3. Profiles T1, T2, and T3 require the support of read and/or write functional units.

4. Support of the <File Transfer and Management> service class is optional. The rules for including it in a request and for the response to it are as given in ISO 8571-3, Clause 10.1. Any implementation including TM in the request must be prepared for the possibility that it might be removed from the response.

5. The support of the <Unconstrained> service class is outside the scope of these Implementation Profiles.

6. Limited File Management is not required for the T- and A- Implementation Profiles, but very often it will be a user request to have limited file management functionality available together with file transfer and file access functions. So Limited File Management may be added as an option to the T- and A- Implementation Profiles.

7. [] in table 9.7 specifies that the document type is optional for the respective Implementation Profile. For M1 the support level depends on the T- or A- Implementation Profile, in conjunction with which M1 is implemented.

8. The Implementation Profiles specify functionality which includes the requirements for conformant implementations as specified in section 9.17. This is a general basic requirement and is not also reflected in table 9.7.

9.19 PROVISION OF SPECIFIC FUNCTION

9.19.1 Implementation Profile T1: Simple File Transfer

Implementation Profile T1 provides the function of transferring entire files at the external file service level for files with an unstructured constraint set. This includes support of the document types:

- FTAM-1 "ISO FTAM unstructured text"
- FTAM-3 "ISO FTAM unstructured binary"
- NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- read a complete file

  and/or

- write (replace, extend) to a file.

9.19.2 Implementation Profile T2: Positional File Transfer

Implementation Profile T2 provides the function of transferring files at the external file service level for files with an unstructured or flat constraint set. This includes support of the document types:

- FTAM-1 "ISO FTAM unstructured text"
- FTAM-2 "ISO FTAM sequential text"

- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file" (optional)
- o NBS-7 "NBS-7 FTAM random access file" (optional)
- o NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by position

  and/or

- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU.

This Implementation Profile is upwardly compatible to T1 for the transfer of unstructured files.

9.19.3 Implementation Profile T3: Full File Transfer

Implementation Profile T3 provides the function of transferring files at the external file service level for files with an unstructured, flat or general hierarchical constraint set. This includes support of the document types:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-2 "ISO FTAM sequential text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file"
- o NBS-7 "NBS-7 FTAM random access file"
- o NBS-8 "NBS-8 FTAM indexed file"
- o NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by key or by position

  and/or

- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU.

This Implementation Profile is upwardly compatible to T1 for the transfer of unstructured files.

9.19.4 Implementation Profile A1: Simple File Access

Implementation Profile A1 provides the function of transfer of and access to files with unstructured or flat constraint sets at the external file service level. This includes support of the document types:

- FTAM-1 "ISO FTAM unstructured text"
- FTAM-2 "ISO FTAM sequential text"
- FTAM-3 "ISO FTAM unstructured binary"
- NBS-6 "NBS-6 FTAM sequential file" (optional)
- NBS-7 "NBS-7 FTAM random access file" (optional)

This Implementation Profile supports file transfer and file access, that is the ability to:

- read a complete file or FADUs which are identified by position,
- write (replace, extend, insert depending on constraint set and document type) to a file or an FADU,
- locate and erase within files.

9.19.5 Implementation Profile A2: Full File Access

Implementation Profile A2 provides the function of transfer of and access to files with unstructured or flat constraint sets at the external file service level. This includes support of the document types:

- FTAM-1 "ISO FTAM unstructured text"
- FTAM-2 "ISO FTAM sequential text"
- FTAM-3 "ISO FTAM unstructured binary"
- NBS-6 "NBS-6 FTAM sequential file"
- NBS-7 "NBS-7 FTAM random access file"
- NBS-8 "NBS-8 FTAM indexed file"

This Implementation Profile supports file transfer and file access, that is, the ability to:

- read from a complete file, or from a series of FADUs which are identified by key or by position,
- write (replace, extend, insert depending on constraint set and document type) to a file or an FADU,
- locate and erase within files.

9.19.6 Implementation Profile M1: Management

Implementation Profile M1 provides the function for an Initiator to manage the files within the Virtual Filestore, to which access is provided by the Responder. Management includes the services of:

- creating a file
- deleting a file
- reading attributes of a file
- changing attributes of a file.

9.20 HARMONIZATION

The Implementation Profiles for File Transfer, File Access and Management correspond to the Profiles of SPAG (Standards Promotion and Application Group) in Europe, so that interworking will be possible. Those Profiles are described in the 'Guide to the Use of Standards' (GUS); they are the basis for the Functional Standards as defined by CEN/CENELEC (Comite Europeenne de Normalization).

Table 9.8. Implementation Profiles (NIST) and Profiles (SPAG/CEN-CLC)

Implementation Profile	SPAG/CEN-CLC
T1	A/111
T2	A/112
T3	A/113
A1	A/122
A2	A/123
M1	A/13

APPENDICES

APPENDIX A: FTAM DOCUMENT TYPES

APPENDIX B: CONSTRAINT SETS

APPENDIX C: ABSTRACT SYNTAXES

APPENDIX A: FTAM DOCUMENT TYPES

A.1 NBS-6 Sequential file document type

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) document-type(5) sequential(6)}
was withdrawn on March 16, 1990. It was replaced with the object NBS-6 Sequential file document type with the Object Identifier
| {iso identified-organization oiw(14) ftamsig(5) document-type(5) sequential(6)}
defined in chapter 9, A.5.

A.2 NBS-7 Random access file

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) document-type(5) random-file(7)}
was withdrawn on March 16, 1990. It was replaced with the object NBS-7 Random access file document type with the Object Identifier
| {iso identified-organization oiw(14) ftamsig(5) document-type(5) random-access(7)}
defined in chapter 9, A.6.

A.3 NBS-8 Indexed sequential file

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) document-type(5) indexed-file(8)}
was withdrawn on March 16, 1990. It was replaced with the object NBS-8 Indexed sequential file document type with the Object Identifier
| {iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file(8)}
defined in chapter 9, A.7.

A.4 NBS-9 File directory file

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) document-type(5) file directory(9)}
was withdrawn on March 16, 1990. It was replaced with the object NBS-9 File directory file document type with the Object Identifier
| {iso identified-organization oiw(14) ftamsig(5) document-type(5) file-directory(9)}
defined in chapter 9, A.8.

| Part 1: Document Types

| A.5 NBS-6 Sequential file document type

1. Entry Number: NBS-6

2. Information objects

Table 9.9. Information objects in NBS-6

document type name	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999)-organization-code(1)~~ document-type(5) sequential(6)} "NBS-6 FTAM sequential file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization oiw(14) ftamsig(5) ~~ied(999)-organization-code(1)~~ abstract-syntax(2) nbs-as1(1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1)} "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}
  Parameter0 ::=  [0] INTEGER {univer-time (23),
                               gen-time (24),
                               boolean (1),
                               null (5) }
  Parameter1  ::= [1] SEQUENCE  {
                  universal-class-number-1 INTEGER { int       (2),
                                                     bit       (3),
                                                     ia5      (22),
                                                     graphic  (25),
                                                     general  (27),
                                                     octet     (4)},
                  string-length INTEGER }
  Parameter2  ::= [2] SEQUENCE {
                  private-class-number INTEGER {float (0)},
                  length-1     INTEGER,
                  length-2     INTEGER }
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set(4) sequential-flat(2)} "FTAM sequential flat constraint set"

```
file contents:
               Datatype1 ::= PrimType -- as defined in Appendix C.3
                                      ~~Annex-9A, Part-3~~
               Datatype2 ::=          Node-Descriptor-Data-Element
```

3. Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the sequential flat constraint set (see table 9.9) These definitions appear in ISO 8571-2. As additional constraints FADU identity will be limited to 'begin', 'end', 'first' and 'next'.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in the Appendix C.3 Annex-9A,-Part 3 of this document. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

The string length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 9.9, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 9.9, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1," carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2."

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 9.9 for all presentation data values transferred. An implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-6 may be simplified to the document type FTAM-3 (allowed only when reading the file). The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

11.2 Access context selection

A document of type NBS-6 may be accessed in any one of the access contexts defined in the sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied at the end of file the transferred material shall be the series of FADUs which would be generated by reading any NBS-6 document with the same parameter values in access context FA.

A.6 NBS-7 Random access file

1. **Entry number: NBS-7**

2. **Information objects**

Table 9.10. Information objects in NBS-7

document type name	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999) --organization-code(1)~~ document-type(5) random-file(7)} "NBS-7 FTAM random access file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999) organization-code(1)~~ abstract-syntax(2) nbs-as1(1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1)} "Basic Encoding of a single ASN.1 type"

parameter syntax:

```
  PARAMETERS ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}
  Parameter0 ::=  [0] INTEGER {univer-time (23),
                                                    gen-time (24),
                                                    boolean (1),
                                                    null (5) }
  Parameter1  ::= [1] SEQUENCE  {
                  universal-class-number-1 INTEGER { int       (2),
                                                     bit       (3),
                                                     ia5      (22),
                                                     graphic  (25),
                                                     general  (27),
                                                     octet     (4)},
                  string-length INTEGER }
  Parameter2  ::= [2] SEQUENCE {
                  private-class-number INTEGER {float (0)},
                  length-1     INTEGER,
                  length-2     INTEGER }
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999) organization-code(1)~~ constraint-set(4) nbs ordered-flat(1)} "NBS ordered flat constraint set"

file contents:

```
                Datatype1 ::= PrimType -- as defined in Appendix C.3
                                       ~~Annex-9A, Part-3~~
                Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                         Enter-Subtree-Data-Element }
                                         Exit-Subtree-Data-Element }
```

3. **Scope and field of application**

The document type defines the contents of a file for storage, for transfer and access by FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. **References**

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the NBS-ordered-flat constraint set (see table 9.10). These definitions appear in Appendix B.2 ~~9A, Part~~ 2 of this document.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in the Appendix C.3 ~~Annex 9A, Part~~ 3 of this document. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

The string length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 9.10, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 9.10, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1," carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of " Datatype2." All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name " asname2."

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 9.10 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-7 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

A document of type NBS-7 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>.

11.2 Access context selection

A document of type NBS-7 may be accessed in any one of the access contexts defined in the NBS-ordered-flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied at the end of file the transferred material shall be the series of FADUs which would be

generated by reading any NBS-7 document with the same parameter values in access context FA.

A.7 NBS-8 Indexed sequential file

1. **Entry Number: NBS-8**
2. **Information objects**

Table 9.11. Information objects in NBS-8

document type name	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999)-organization-code(1)~~ document type(5) indexed-file(8)} "NBS-8 FTAM indexed file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999)-organization-code(1)~~ abstract syntax(2) nbs-as1(1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE            {DataTypes, KeyType, KeyPosition}

  DataTypes  ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}

  KeyType    ::= CHOICE {Parameter0, Parameter1, Parameter2}

             --  Parameter0, Parameter1, Parameter2, as defined for the
             --  document types NBS-6, NBS-7

  KeyPosition::=  INTEGER
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set(4) ordered-flat(3) } "FTAM ordered flat constraint set"

```
file contents:
                Datatype1 ::= PrimType -- as defined in Appendix C.3
                                          ~~Annex-9A,-Part-3~~
                Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                         Enter-Subtree-Data-Element }
                                         Exit-Subtree-Data-Element }
```

3. **Scope and field of application**

The document type defines the contents of a file for storage, for transfer and access using FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. **References**

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set (see table 9.11). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set:

- The FADU identities 'first', 'last', and 'node number' are not required for conformant implementations
- The identities 'next' and 'previous' are allowed for all FADUs

Each data element is a data type from the set of primitive data types defined in Appendix C.3 ~~9A, Part 3~~ of this document. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the <DataTypes> parameter.

The string length field of Parameter1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the

actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

Each data unit in the file has a key associated with it, which is the user-coded form of Node Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in Appendix C.3 9A, Part 3.

The type and length of the key are defined by the <KeyType> parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in the following table 9.12.

Table 9.12. Datatypes for keys

Key Type	Minimum Range (octets)	Order
ASN.1 INTEGER	(1-2)	increasing numeric value
ASN.1 IA5String	(1-16)	lexical order
ASN.1 GraphicString	(1-16)	lexical order
ASN.1 GeneralString	(1-16)	lexical order
ASN.1 OCTET STRING	(1-16)	increasing value
ASN.1 GeneralizedTime		increasing time value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing numeric value

The position of the key in the data unit is specified by the <position> parameter.
position = 0 implies the key is not part of the data
position > 0 specifies the actual data element in the data unit.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 9.11, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 9.11, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1," carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2."

Notes:
1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice
2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 9.11 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-8 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

A document of type NBS-8 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>. The traversal order of the FADUs must be maintained.

Note: The traversal order is as reading the file as NBS-8 in key order.

11.2 Access context selection

A document of type NBS-8 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied the transferred material shall be the series of FADU which would be generated by reading any NBS-8 document with the same parameter values in access context FA.

The insertion of a new FADU after an already existing FADU will be indicated via a diagnostic on TRANSFER-END.

11.4 The EXTEND operation

This operation is excluded for the use with this document type.

A.8 NBS-9 File directory file

1. **Entry Number: NBS-9**

2. **Information objects**

Table 9.13. Information objects in NBS-9

document type name	{iso identified-organization oiw(14) ftamsig(5) ~~ied(9999)-organization-code(1)~~ document type(5) file-directory(9)} "NBS-9 FTAM file directory file"
abstract syntax names:	{iso identified-organization oiw(14) ftamsig(5) ~~(9999)-organization-code(1)~~ abstract-syntax(2) nbs-as2(2)} "NBS file directory entry abstract syntax"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1)} "Basic Encoding of a single ASN.1 type"

parameter syntax

```
PARAMETERS ::=  [0] IMPLICIT BIT STRING {

            -- Kernel group

                 read-filename (0),
                 read-permitted-actions (1),
                 read-contents-type (2),

            -- Storage group

                 read-storage-account (3),
                 read-date-and-time-of-creation (4),
                 read-date-and-time-of-last-modification (5),
                 read-date-and-time-of-last-read-access (6),
                 read-date-and-time-of-last-attribute-modification(7),
                 read-identity-of-creator (8),
                 read-identity-of-last-modifier (9),
                 read-identity-of-last-reader (10),
                 read-identity-of-last-attribute-modifier (11),
                 read-file-availability (12),
                 read-filesize (13),
                 read-future-filesize (14),

            -- Security group

                 read-access-control (15),
                 read-legal-qualifications (16),

            -- Private group

                 read-private-use (17) }
```

(Continued on next page.)

Table 9.13. Information objects in NBS-9 continued.

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint-set	{iso standard 8571 constraint-set(4) unstructured(1)} "FTAM unstructured constraint set"
File contents: Datatype1 ::= FileDirectoryEntry --As defined by NBS-AS2 in Appendix A, --C.4 Part-3 of this document	

3. **Scope and field of Application**

This document defines the contents of a file for transfer (not for storage) using FTAM.

4. **References**

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management.

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1

6. **Abbreviations**

FTAM File Transfer, Access and Management.

7. **Document Semantics**

The document consists of one file access data unit, which consists only of zero, one or more data elements of type <FileDirectoryEntry> (defined in NBS-AS2).

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the unstructured constraint set. These definitions appear in ISO 8571-1.

The parameter of the document type is used on <F-OPEN request> to specify the desired attributes of each of the files on the Filestore, when reading the document.

8. **Abstract syntactic structure**

The abstract syntactic structure of the document is a series of file directory entries, each of which is defined by the <FileDirectoryEntry> definition in NBS-AS2.

Additional constraints are defined for this document type: File access actions are restricted to Read. File-directory files may not be Written or Modified (except as a side effect of actions performed on individual files contained within a file directory).

9. Definition of transfer

9.1 Datatype definition

The file consists of zero or more values of Datatype1 defined in table 9.13.

9.2 Presentation data values

The document is transferred as a series of presentation data values. Each presentation data value shall consist of one value of the ASN.1 data type "Datatype1," carrying one of the file directory entries from the document.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 9.13.

9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of file directory entries within the Data Unit in the file.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 9.13 for all presentation data values transferred. Implementations shall optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

Relaxation is allowed to any bitstring combination of the document type parameter.

APPENDIX B: CONSTRAINT SETS

B.1 NBS Ordered flat constraint set

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) constraint-set(4) nbs-ordered-flat(1)}
was withdrawn on March 16, 1990. It was replaced with the object NBS Ordered flat constraint set with the Object Identifier
{iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-ordered-flat(1)}
defined in chapter 9, B.2.

B.2 NBS Ordered flat constraint set definition

1. Field of application

The NBS-ordered flat constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made on an FADU basis by position in the sequence.

2. Basic constraints

Table 9.14. Basic constraints for NBS Ordered flat

Constraint set descriptor	"NBS ordered flat constraint set"
Constraint set identifier	{iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-ordered-flat(1)}
Node name	None
File access actions	Locate, Read, Insert, Erase, Replace
Qualified action	None
Available access contexts	HA, FA, UA
Creation state	Root node without an associated data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected; 'previous' gives last node in traversal sequence,'current'and 'next'give an error.
Read whole file	Read in access context FA or UA with FADU identity of 'begin'.
Write whole file (append) Write whole file (replace)	Transfer the series of leaf FADUs which would be generated by reading the whole file in access context FA; perform the transfer with an FADU identity of 'end' and a file access action of 'insert'. Transfer the series of leaf FADUs which would be generated by reading the whole file in access-context HA; perform the transfer with FADU identity 'begin' and file action of 'replace'.

3. Structural constraints

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and may have an associated data unit; all arcs from the root node shall be of length one.

4. Action constraints

Insert: The <Insert> action is allowed only at the end of file. If the FADU identity is 'end' the new node is inserted

following all existing nodes in the file. If the FADU identity is 'node number', the number must be at least one greater than the node number of the last existing node. Any nodes between the last existing node and the new node are empty, i.e., nodes without data. If the FADU identity is a 'node number' not greater than that of the last existing node, an error will occur. The location following <insert> is 'end'.

Erase: The Erase action is only allowed at the root node to empty the file, with FADU identity of 'begin'. The result is a solitary root node without an associated data unit.

Note: It is the intention when using this constraint set to allow for emptying an FADU, i.e., leaving an FADU with a DU of data length 0 (or without a DU); afterwards data may be reinserted into this hole. In order to empty an FADU, the <Replace> operation may be used with new data of length zero (or with an FADU whose <data exists> bit is set to 'false' and no DU). Refilling the hole is accomplished by a <Replace> operation with the new DU (or with the new FADU, whose <data exists> bit is set to 'true' and the new DU).

5. Identity constraints

The FADU identity associated with the file action shall be one of the identities 'begin', 'end', 'first', 'last', 'current', 'next', 'previous' or a 'node number' greater than or equal to one. The actions with which these identities can be used are given in the following table.

Table 9.15. Identity constraints in NBS Ordered flat

Action	Begin	End	First	Last	Current	Next	Previous	Node No.
Locate	valid	valid	valid	valid	valid	valid	valid	valid
Read	whole		leaf	leaf	leaf	leaf	leaf	leaf
Insert		leaf						leaf
Erase	whole							
Replace	whole		leaf	leaf	leaf	leaf	leaf	leaf

APPENDIX C: ABSTRACT SYNTAXES

C.1 Abstract Syntax NBS-AS1

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) abstract-syntax(2) nbs-as1(1)}
was withdrawn on March 16, 1990. It was replaced with the object Abstract syntax NBS-AS1 with the Object Identifier
{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1)}
defined in chapter 9, C.3.

C.2 Abstract Syntax NBS-AS2

This object with Object Identifier
{iso identified-organization icd(9999) organization-code(1) abstract-syntax(2) nbs-as2(2)}
was withdrawn on March 16, 1990. It was replaced with the object Abstract syntax NBS-AS2 with the Object Identifier
{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2)}
defined in chapter 9, C.4.

C.3 Abstract Syntax NBS-AS1 definition

Abstract syntax name: {iso identified-organization oiw(14) ftamsig(5) ied(9999) organization-code(1) abstract-syntax(2) nbs-as1(1)}
"NBS abstract syntax AS1"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 type NBS-AS1.PrimType

```
NBS-AS1 DEFINITIONS ::=
BEGIN
PrimType ::= CHOICE {   INTEGER,
                        BIT STRING,
                        BOOLEAN,
                        IA5String,
                        GraphicString,
                        GeneralString,
                        OCTET STRING,
                        UTCTime,
                        GeneralizedTime,
                        NULL,
                        FloatingPointNumber }

                        --   The support for IA5String is the ISO 646,
IRV                     --      G0 character set and the ISO 646, IRV C0
                                set
                        --   The minimum level of support for
                        --   GraphicString is the ISO 646, IRV G0
                        --   character set and the 8859-1 G0 and G1 sets
                        --   The minimum level of support for
                        --   GeneralString is the ISO 646, IRV G0
                        --   character set and the 8859-1 G0 and G1
                        --   character sets, and ISO 646, IRV C0 set.

FloatingPointNumber ::=     [PRIVATE 0]    CHOICE {
                                finite [0] IMPLICIT SEQUENCE
                                     {    Sign,
                                          mantissa BIT STRING,
                                     --   first bit must be 1
                                          exponent INTEGER},
                                infinity [1] IMPLICIT Sign,
                                signalling-nan [2] IMPLICIT NaN,
                                quiet-nan [3] IMPLICIT NaN,
                                zero [4] IMPLICIT NULL }

Sign    ::= INTEGER { positive (0), negative (1) }
NaN     ::= INTEGER
END
```

For this abstract syntax the following transfer syntax can be used

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

Notes: 1. The mantissa is a number in the range (1/2 <mantissa<1).

2. The value is equal to mantissa * 2 ^{exponent}.

3. The first bit in the mantissa is most significant.

4. See IEEE 754 for definitions of terminology, such as NaN.

5. A minimum length range (in bits) is required for the components of <FloatingPointNumber>, as follows: mantissa 1-23 bits, and exponent 0-8 bits.

C.4 Abstract Syntax NBS-AS2 definition

Abstract syntax name: { iso identified-organization oiw(14) ftamsig(5) ~~ied(9999)-organization-code(1)~~ abstract-syntax(2) nbs-as2(2) }

"NBS file directory entry abstract syntax"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 Type NBS-AS2.FileDirectoryEntry.

```
NBS-AS2 DEFINITIONS ::=

BEGIN

FileDirectoryEntry  ::=[PRIVATE 2] Read-Attributes

Read-Attributes     ::=ISO8571-FTAM.Read-Attributes

END
```

For this abstract syntax the following transfer syntax will be used

{ joint-iso-ccitt asn1(1) basic-encoding(1) }

"Basic Encoding of a single ASN.1 type"

C.5 Abstract Syntax "FTAM unstructured text abstract syntax"

This abstract syntax is defined as DataType1 (File Contents) in table 19 of ISO 8571-2, Annex B.

C.6 Abstract Syntax "FTAM unstructured binary abstract syntax"

This abstract syntax is defined as DataType1 (File Contents) in table 21 of ISO 8571-2, Annex B.

| ~~Part 4: (deleted)~~

| (former Part 4 deleted)

10. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 3

Editor's Note: The "NBS" designation remains in effect for document types, abstract syntaxes, and constraint sets defined in all FTAM agreements up to 1/1/89. After 1/1/89, any new functionality references the "NIST" designation. This is to reflect the change in identifying organization from "NBS" to "NIST."

10.1 INTRODUCTION

This section contains Implementors Agreements based on ISO 8571 File Transfer, Access and Management. These Agreements define enhancements to the Stable FTAM Implementation Agreements for OSI Protocols, Version 1, Edition 1, December 1987 (FTAM Phase 2 Agreements, NBS 500-150), including all their subsequent Errata changes as specified in Version 3, Edition 1 (NIST Special Publication 500-177, this document ch. 9).

Therefore it is assumed that the reader is familiar both with the contents of the base standard ISO 8571 and its underlying layers, and also with the above-mentioned NIST FTAM Phase 2 specifications.

Phase 2 Agreements define six Implementation Profiles which are T1, T2, T3, A1, A2, and M1. In order to avoid ambiguity when referring to these Implementation Profiles the above designations will apply only to Phase 2 functionality, references to Phase 3 enhanced Implementation Profiles will be by the addition of a'.3', i.e., T1.3, T2.3, T3.3, A1.3, A2.3, and M1.3.

The following sections specify the functionality of NIST OIW FTAM Phase 3.

- Sections 10.2 and 10.9 specify the technical details of FTAM Phase 3 which are defined in addition to the functionality of FTAM Phase 2. Included is also a status-overview regarding statements on Phase 2/Phase 3 compatibility and interworking.

- Appendix A is a Profile Requirements List for the Implementation Profiles T1.3, T2.3, A1.3 and M1.3, summarizing all features of FTAM Phase 3, including those of FTAM Phase 2. This Profile Requirements List is fully based on the FTAM PICS Proforma ISO DIS 8571-5.

- Appendix B is an index of Object Identifiers. It is the official NIST OIW REgister of NIST OIW defined FTAM objects. It contains the Object Descriptors and Object Identifiers for these objects, including a reference to the section in the NIST OIW Stable Agreements where the respective object is being defined.

- Appendices C, D, and E provide definitions for additional document types, constraing sets and abstract syntaxes.

10.2 SCOPE AND FIELD OF APPLICATION

These Phase 3 Agreements specify additional functionality to the FTAM Phase 2 Agreements. These additional functions include:

- Further specifications of document types,
- Specification for Restart Data Transfer and Recovery functional units,
- Specification of FADU Locking functional unit, and
- More details on Access Control and Concurrency Control.

All Phase 2 systems are upward compatible to a Phase 3 system and can therefore interwork with it, if the additional functions are negotiated out (e.g., use of Recovery) or not used for the interconnection (e.g., additional features for document types).

10.3 STATUS

These FTAM Phase 3 Agreements were completed December 15, 1989. No further enhancements will be made to this version (see also next section ERRATA).

The following tables summarize the functions and features which are defined for FTAM Phase 3 in addition to the FTAM Phase 2 specifications. They also state the degree of possible interworking and the backward compatibility.

Table 10.0(a). Phase 2/Phase 3 Interworking

Additional Requirements in FTAM Phase 3	Backward Compatibility to FTAM Phase 2
FTAM-1: GraphicString, VisibleString FTAM-2: VisibleString concurrency-control parameter for Initiator create-password parameter for Initiator Profile M1.3: Requires support of (1)-T service class including Limited File Management FU, Enhanced FM FU; TM service class including Enhanced FM FU or (2)-A service class including Limited File Management FU, Enhanced FM FU	full backward compatibility if the additional features of Phase 3 are not being used (character sets in FTAM-1, -2), or not requested by an Initiator (functional units) or not required by a Responder (parameters) not requested by an Initiator (functional units)

Table 10.0(b). Phase 2/Phase 3 Interworking (continued)

Additional Optional Features in FTAM Phase 3	Backward Compatibility to FTAM Phase 2
FTAM-2: GeneralString, IA5String	full backward compatibility if the additional features of Phase 3 are not requested, negotiated out or not being used
FTAM-4	
NBS-8 in T2.3, A1.3	
NBS-9 in A1.3, A2.3	
NBS-10	
NBS-11	
NBS-12	
Recovery functional unit	
Restart-data-transfer functional unit	
FADU-locking functional unit and FADU-lock parameters in A1.3, A2.3	
concurrency-control parameters for Responder	
create-password parameter for Responder	
location-field of access-control element	
suggested-delay term of diagnostic parameter supported conditionally on Recovery ~~or Restart-data-transfer~~ functional units	

Table 10.0(c). Phase 2/Phase 3 Interworking (continued)

Relaxation for FTAM Phase 3	Backward Compatibility to FTAM Phase 2
Profiles A1.3, A2.3 do not require transfer service class	if T service class not being used
no minimum requirement for maximum-string-length parameters for document types	if a Phase 3 system stays below this minimum requirement

10.4 ERRATA

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	SECTION	DESCRIPTION
CP 3/90-1	TECHNICAL	NIST-SP 500-177	A.9, A.13 B.1, B.2.1	NBS-6, NBS-7, NBS-8, NBS-9, NBS-AS1, NBS-AS2, NBS-ordered-flat constraint set with Obj Id. using icd(9999) withdrawn. Above FTAM objects newly defined with Obj. Id. using oiw(14) ftamsig(5)
CP 3/90-3	EDITORIAL		Appendices C, D, E	structuring into subsections, all references to these definitions up-dated accordingly
CP 3/90-5	TECHNICAL		10.8.5.2	support of suggested-delay field of diagnostic not conditional on Restart-data-transfer

10.5 CONFORMANCE

In addition to the specific requirements specified in the following subsections, conformance to this Phase 3 specification requires

- conformance to ISO 8571: 1988
- conformance to Phase 2 FTAM, unless specified otherwise in this chapter 10.

10.5.1 Conformance for Access Profiles

The access Profiles A1.3 and A2.3 do not include the requirement for transferring files using the File Transfer service class.

10.6 ASSUMPTIONS

FTAM Phase 3 Agreements specify additional functionality to the Implementation Profiles T1, T2, T3, A1, A2, and M1 as defined in the FTAM Phase 2 Agreements. So all definitions and requirements for these Implementation Profiles apply also to the Phase 3 Agreements.

10.7 FILESTORE AGREEMENTS

10.7.1 Document Types

In addition to the Phase 2 Document Type Agreements the document types FTAM-4 (see ISO 8571-2, Annex B) and NBS-10, NBS-11, NBS-12 (see Appendix C) are defined for optional support.

Table 10.1 gives the support levels for all document types with respect to the Implementation Profiles.

For FTAM-1, FTAM-2, FTAM-3 and FTAM-4 the supported parameter values for <universal class number> and <string significance>, respectively are listed. Other values are outside the scope of these Agreements. No restriction or minimum requirement is defined for the <maximum string length> parameter of these document types.

Table 10.1. Implementation Profiles and Document Types
(a) FTAM-1 Through FTAM-4

Implementation Profile (Note 1)	Document Type	Universal Class Number (Notes 1, 3, 4, 5)	String Significance
T1.3, T2.3, T3.3, A1.3, A2.3	FTAM-1	Graphic String (25)	'variable' 'fixed'
		VisibleString (26)	'variable' 'fixed'
		GeneralString (27)	'not-significant'
		IA5String (22)	'not-significant'
T2.3, T3.3, A1.3, A2.3	FTAM-2	GraphicString (25)	'not-significant'
		VisibleString (26)	'not-significant'
		[GeneralString (27)]	'not-significant'
		[IA5String (22)]	'not-significant'
T1.3, T2.3, T.3.3, A1.3, A2.3	FTAM-3	-	'not-significant'
[T2.3], [T3.3], [A1.3], [A2.3]	FTAM-4	-	'not-significant'

Table 10.1. Implementation Profiles and Document Types
(b) NBS-6 Through NBS-11

Implementation Profile (Note 1)	Document Type
[T2.3], T3.3, [A1.3], A2.3	NBS-6
[T2.3], T3.3, [A1.3], A2.3	NBS-7
[T2.3], T3.3 [A1.3], A2.3	NBS-8
[T1.3], [T2.3], [T3.3], [A1.3],[A2.3]	NBS-9
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-10
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-11

Table 10.1. Implementation Profiles and Document Types
(c) NBS-12

Implementation Profile (Note 1)	Document Type	Universal Class Number	Character-Set Escape Sequences as defined for Reg. Numbers C0 G0 G1	String-Significance
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-12 See Note 6	IA5String [22]	(parameter absent)	'variable' 'fixed'
		GraphicString[25]	(parameter absent)	'variable' 'fixed'
		GraphicString[25]	- 6 100	'variable' 'fixed'
		VisibleString[26]	(parameter absent)	'variable' 'fixed'
		GeneralString[27]	(parameter absent)	'variable' 'fixed'
		GeneralString[27]	1 6 100	'variable' 'fixed'

Notes:

1. Brackets around a Profile designator or a parameter value indicate that the respective document type or parameter value is optionally supported in this Implementation Profile.

2. The support level for document types in Implementation Profile M1.3 depends on the T- or A-Implementation Profile, in conjunction with which M1.3 is implemented.

3. The support for IA5 String is the ISO 646, IRV G0 character set and the ISO 646, IRV C0 set.

4. The minimum level of support for Graphic String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 sets.

5. The minimum level of support for General String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 sets, and ISO 646, IRV C0 set.

6. If the Character-Set parameter is absent, the following defaults apply:

Universal-Class-Number	Default Registration Numbers		
	C0	G0	G1
IA5String [22]	1	2	-
GraphicString [25]	-	2	-
VisibleString [26]	-	2	-
GeneralString [27]	1	2	-

Character-Sets and Escape Sequences:

Registration Number	Content	Escape Sequence
1	C0 set of ISO 646	ESC 2/1 4/0
2	ISO 646, IRV	-
6	ISO 646, USA Version-X 3.4 - 1968 (Left-hand part of ISO 8859-1)	ESC 2/8 4/2
100	Right-hand part of Latin Alphabet No 1 ISO 8859-1, ECMA-94	ESC 2/13 4/1

10.7.2 FADU Identities

In addition to the Phase 2 FADU Identity Agreements the following is specified:

> For the document type NBS-11 used in conjunction with the Transfer service class or the Transfer and Management service class, the support of the FADU identities of 'current', 'next', 'previous' and 'end' is outside the scope of these Agreements.

10.7.3 Access Control Attribute

The location field of access control element is optionally supported. It is the implementor's choice which combinations of fields in an access control element are supported. The ACE combination should be stated in the PICS.

10.8 PROTOCOL AGREEMENTS

10.8.1 Implementation Profile M1.3

The functions defined for the Implementation Profile M1.3 shall always be implemented in conjunction with one or more of the Implementation Profiles T1.3, T2.3, A1.3, or A2.3. The service classes and functional units that shall be implemented are specified in section 10.10, Appendix A, A.12.4 and A.12.5.

For an implementation supporting the Profile M1.3 in conjunction with T1.3 or T2.3, any of the service classes Transfer, Management or (Transfer, Management, Transfer-and-Management) may be requested and any of the classes Transfer, Management, Transfer-and-Management may be responded on F-INITIALIZE.

For an implementation supporting the Profile M1.3 in conjunction with A1.3 or A2.3, any of the service classes Access or Management may be requested and responded on F-INITIALIZE.

10.8.2 Functional Units

For FTAM Phase 3 implementations Recovery and Restart Data Transfer are optionally supported.

FADU locking is optionally supported for Implementation Profiles A1.3 and A2.3.

10.8.3 Implementation Information Parameter

In addition to the Agreements as specified for FTAM Phase 2, section 9.12 , the following value is defined

NBS-Phase3.

10.8.4 F-Check

In order to maximize interoperability, implementations of FTAM service providers should not restrict the amount of data transmitted between successive F-CHECK requests to a single quantity. Variations in the amount of data transmitted between checkpoints may be required to accommodate differences in real end systems supporting FTAM Virtual Filestores and/or in the communications media underlying FTAM associations. It is required that all FTAM implementations are able to receive at least one PSDU between checkpoints.

10.8.5 Error Recovery

Procedures for Class I, II and III errors are defined and supported for FTAM Phase 3 implementations. It is the implementor's choice whether to handle class I errors using F-RESTART PDUs or whether to use the class II error procedure.

10.8.5.1 Docket Handling

When a class III error occurs, the length of time a docket is maintained is determined by the local system. Recovery from a class III error is only possible as long as both end systems maintain the docket.

It is also a local decision how many dockets can be maintained simultaneously.

10.8.5.2 Parameters for Error Recovery

- The semantics of the <FTAM quality of service> parameter is as defined in ISO 8571, including the local knowledge of FERPM.
- No minimum requirement for the <checkpoint window> parameter or the checkpoint size is defined.
- For the <recovery mode> parameter of F-OPEN, the values 'none' and 'at-start-of-transfer' are supported. The value 'at-any-active-checkpoint' is optionally supported. If recovery mode 'at-start-of-transfer' is negotiated, no F-

CHECK shall be issued. When recovering at the start of the transfer, the <recovery point> value of 0 shall be used.

- It is required that Responders implementing the Restart-data-transfer or the Recovery functional unit must be able to negotiate <recovery mode> parameter to a value other than 'none'.

- For the <diagnostic> parameter of F-INITIALIZE, F-P-ABORT and F-RECOVER PDUs, the term <suggested delay> shall be supported if the Recovery ~~or Restart-data-transfer~~ functional units is are implemented. The Basic FERPM should wait at least the amount of time as given by the <suggested delay> term before attempting to recover.

10.8.6 Concurrency Control

10.8.6.1 Concurrency Control to whole file

The <concurrency control> parameters of F-SELECT, F-CREATE and F-OPEN with or without the <access control> attribute of Security Group are supported for Initiators and optionally supported for Responders.

If supported by a Responder, details of their possible usage is a local matter and shall be specified in the PICS.

Default values for concurrency control are as specified for FTAM Phase 2 Agreements.

No minimum requirement is defined for <concurrency control> parameter values.

For a first accessor either the specified concurrency locks or the default values are assigned. For a subsequent accessor the access to a file is granted only if this concurrency control requirement, as specified in this concurrency control parameter or given by the default values, can be met. Otherwise the subsequent request shall be rejected.

10.8.6.2 FADU Locking

FADU locking functional unit and the respective <FADU lock> parameters are optionally supported for the Implementation Profiles A1.3 and A2.3.

It is understood that ISO 8571-4 Clause 18.4 also applies to FADU locks; that means that as long as a docket is maintained, FADU locks locking any FADUs recorded in that docket should be maintained.

10.8.7 Create Password

The <create password> parameter for an implementation acting as an Initiator is supported. This parameter is optionally supported for an implementation acting as a Responder.

10.8.8 Initiator Identity, Passwords and Account

An Initiator must be capable of sending and not sending the parameters <initiator identity>, <filestore password>, <access passwords> and <create password> to satisfy the requirements of the Responder.

The contents of the <initiator identity>, <filestore password>, <access passwords>, <create password> and <account> parameters shall be in the convention of the responding implementation.

10.9 Range of Values for Integer-Type Parameter

In addition to the parameters specified for FTAM Phase 2 under the same heading, the parameters

```
F-RECOVER request
          bulk-transfer-number
NBS-AS3
     NBS-Node-Name
          starting-fadu
          fadu-count
```

may be encoded so that the length of its contents octets is no more than eight octets.

A P P E N D I C E S

APPENDIX A: PROFILES REQUIREMENTS LIST FOR NIST OIW FTAM PHASE 3

APPENDIX B: NIST OIW REGISTER OF FTAM OBJECTS

APPENDIX C: DOCUMENT TYPES

APPENDIX D: CONSTRAINT SETS

APPENDIX E: ABSTRACT SYNTAXES

APPENDIX A :

PROFILES REQUIREMENTS LIST FOR NIST OIW FTAM PHASE 3

A.0 Introduction

This appendix to NIST FTAM Phase 3 Agreements defines a Profile Requirements List (PRL) for the Implementation Profiles

- T1.3 - Simple File Transfer
- T2.3 - Positional File Transfer
- A1.3 - Simple File Access
- M1.3 - Management

This appendix specifies the constraints and characteristics of NIST OIW FTAM Phase 3 on what shall or may appear in the supplier columns of an FTAM Phase 3 PICS. This appendix is completely based on ISO 8571-5. It uses only a selection of the tables from ISO 8571-5 which are necessary for the specification of the FTAM Phase 3 status, and retains their numbering, in order to facilitate for a supplier to fill in the respective PICS Proforma.

This appendix is a summary of all definitions of FTAM Phase 3 as they appear in the Stable Implementation Agreements for OSI Protocols, Version 3 Edition 1, December 1989, chapters 9 and 10.

A.0.1 Conformance requirement of Base Standards

The D-column of sections A.1 to A.13 specifies the conformance requirement of the base standards ISO 8571, as written in ISO 8571-5. The definitions apply as defined in ISO 8571-5 clause 8.1 :

- m - mandatory support
- o - optional support
- f - full support of attributes
- p - partial support of attributes
- - - not applicable

A single value in the D-column applies to the Initiator role of a system as well as to the Responder role. If two values are specified in the D-column separated by a space, they apply to the Initiator (I) role and to the Responder (R) role, respectively.

A.0.2 Conformance requirement of Profiles

The Conformance requirement of the Implementation Profiles is specified in the 'Profiles' column/columns in sections A.1 to A.13. The following convention is applied for this purpose :

- o a 'PROFILES' column is valid for all Profiles T1.3, T2.3, A1.3 and M1.3
- o if different conformance requirements apply to different Profiles, separate columns are included in the tables, each bearing the corresponding Profile name as its heading, or separate tables for these Profiles are used
- o a single value in these columns applies to the Initiator as well as to the Responder role of an implementation

- o if two values are specified in a column separated by a space, they apply to the Initiator (I) role and to the Responder (R) role, respectively.

For the conformance requirements of the NIST FTAM Phase 3 Profiles the following abbreviations are used.

mandatory; m :
This is a mandatory or optional feature in the base standard. It shall be supported, i.e., its syntax and procedures shall be implemented as specified in the base standard or in FTAM Phase 3 by all implementations claiming conformance to the Profile.

However, it is not a requirement that the feature shall be used in all instances of communication, unless mandated by the base standard or stated otherwise in FTAM Phase 3.

For fully supported attributes, this implies that at least the minimum range of attribute values, as defined in ISO 8571-2, shall be supported unless stated otherwise in FTAM Phase 3.

Also for features which are optional in the base standard, conformant implementations shall be able to interwork with other implementations not supporting this feature.

The support of a feature can be conditional, depending on the support of a class of features to which it belongs, e.g., an attribute in an attribute group, a parameter in a PDU, a PDU in a functional unit.

optional; o :
It is left to the implementation as to whether this feature is implemented or not.

If an attribute group with a support level of 'o' is chosen to be supported, then all the attributes in this group that are classified as 'm' shall be supported.

The support for PDUs is determined by the negotiation of functional units when the connection is established.

If a parameter is optionally supported, then its syntax shall be implemented, but it is left to each implementation whether its procedures are implemented or not.

When receiving an optional parameter which is not subject of negotiation and is not supported by the Receiver, the Receiver shall at least inform the Sender by informative diagnostic and interworking shall not be disrupted.

conditional; c :
This feature shall be supported under the conditions specified in FTAM Phase 3. If these conditions are not met, the feature is outside the scope of the Profile.

excluded; x :
This feature is excluded from the Profile. The implementor's answer in the PICS shall always be 'no'.

outside the scope; i :
This feature is outside the scope of the Profile, i.e., it may be ignored, and will therefore not be subject of a Profile conformance test. However the syntax of all parameters of supported PDUs shall be implemented, even if their procedures are not (i.e., the Receiver shall be able to decode the PDU).

not applicable; - :
This feature is not defined in the context where it is mentioned, e.g., a parameter which is not part of the respective PDU. The occurrence of 'not applicable' features is mainly due to the format of the tables in the Phase 3 Profiles Requirements List.

A.1 (void)

A.2 (void)

Section 2: General ISO 8571 Detail

A.3 ISO 8571 Protocol versions

1	FTAM protocol version number(s)	**One**

A.4 ISO 8571 Addenda

1	ISO 8571-1	—
2	ISO 8571-2	—
3	ISO 8571-3	—
4	ISO 8571-4	—
5	ISO 8571-5	—

A.5 Defect report numbers and amendments

1	ISO 8571-1	—
2	ISO 8571-2	—
3	ISO 8571-3	—
4	ISO 8571-4	—
5	ISO 8571-5	—

A.6 Global statement of conformance

1	Does FTAM Phase 3 conform to ISO 8571 ?	**yes**

A.7 Initiator / Responder capability

	ROLES	D	PROFILES I	R
1	Sender	o	o	o
2	Receiver	o	o	o

NOTE - See section 9.18.1

A.8 Application Context Name details

1	ISO 8571-4 defines a value for a simple transfer mechanism. Other values are not defined for FTAM Phase 3 (see 9.5(9)).

Section 3 : Syntax Detail

A.9 Abstract syntaxes

	Object Descriptor	Object Identifier	D	T1.3	T2.3	A1.3	M1.3
1	**FTAM PCI**	{iso standard 8571 abstract-syntax(2) ftam-pci(1) }	m	**m**	**m**	**m**	**m**
2	**FTAM FADU**	{iso standard 8571 abstract-syntax(2) ftam-fadu(2) }	o	**i**	**m**	**m**	**i**
3		{joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0) version1(1) }	m	**m**	**m**	**m**	**m**
4	**FTAM unstructured text abstract syntax**	{iso standard 8571 abstract-syntax(2) unstructured-text(3) }	o	**m**	**m**	**m**	**-**
5	**FTAM unstructured binary abstract syntax**	{iso standard 8571 abstract-syntax(2) unstructured-binary(4) }	o	**m**	**m**	**m**	**-**
6	**NBS file directory entry abstract syntax**	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2) }	-	**c**	**c**	**c**	**-**
7	**NBS abstract syntax AS1**	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as1(1) }	-	**i**	**c**	**c**	**-**
8	**NBS random access node name abstract syntax**	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-node-name(3) }	-	**i**	**c** see 10.9	**c**	**-**
9	**NBS random binary access file abstract syntax**	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-random-binary(4) }	-	**i**	**c**	**c**	**-**
10	**NBS simple text abstract syntax**	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-simple-text(5) }	-	**i**	**c**	**c**	**-**

NOTES

1 The abstract syntaxes which are supported in the Implementation Profile M1.3 depend on the T-or A-Profile in conjunction with which M1.3 is implemented.

2 The support requirements for the conditional abstract syntaxes depend on the constraint sets and document types which are implemented (see clause A.13).

3 ISO 8571 requires the presence of the transfer syntax derived from the "Basic Encoding of a single ASN.1 type "{joint-iso-ccitt asn1 (1) basic-encoding (1)} encoding rules for transfer of the "FTAM PCI" and the "FTAM FADU" abstract syntaxes. Implementation detail of this transfer syntax, and other transfer syntaxes supported, is specified in the PICS of ISO 8823.

Section 4 : Virtual Filestore Detail

A.10 Virtual filestore

This clause details the conformance to the file model, file attribute support and to file structure support.

A.10.1 File model

	FILE MODEL	D	PROFILES R
1	Hierarchical	o	**m**
	Other models		**i**

A.10.2 Attributes

A.10.2.1 Attribute groups

	ATTRIBUTE GROUP NAME	D	PROFILES
1	Kernel	m	**m**
2	Storage	o	**o**
3	Security	o	**o**
4	Private	o	**i**

A.10.2.2 Attribute values

	KERNEL GROUP (INITIATOR)	D	PROFILES I full	RANGE OF VALUES
1	Filename	f	**m**	see A.10.2.3
2	Permitted Actions	f	**m**	
3	Contents Type	f	**m**	see A.12.7

	KERNEL GROUP (RESPONDER)	D	PROFILES R full	RANGE OF VALUES
4	Filename	f	**m**	see A.10.2.3
5	Permitted Actions	f	**m**	
6	Contents Type	f	**m**	see A.12.7

	STORAGE GROUP (INITIATOR)	D	PROFILES I full	RANGE OF VALUES
7	Storage account	f	m	
8	Date and time of creation	f	m	
9	File availability	f	m	
10	Future filesize	f	m	see 9.17.9

NOTE - An initiator shall not partially support attributes

	STORAGE GROUP (RESPONDER)	D	PROFILES R full	P partial	RANGE OF VALUES
11	Storage account	p	o	o	
12	Date and time of creation	p	o	o	
13	Date and time of last modification	p	o	o	
14	Date and time of last read access	p	o	o	
15	Date and time of last attribute modification	p	o	o	
16	Identity of creator	p	o	o	
17	Identity of last modifier	p	o	o	
18	Identity of last reader	p	o	o	
19	Identity of last attribute modifier	p	o	o	
20	File availability	p	m	x	
21	Filesize	p	m	x	see 9.17.9
22	Future filesize	p	o	o	see 9.17.9

	SECURITY GROUP (INITIATOR)	D	PROFILES I full	RANGE OF VALUES
23	Access control	f	m	see A.12.2
24	Legal qualifications	f	m	

NOTE - An initiator shall not partially support attributes

	SECURITY GROUP (RESPONDER)	D	PROFILES R full	R partial	RANGE OF VALUES
25	Access control	p	m	x	see A.12.2, 9.9.2
26	Legal qualifications	p	o	o	

A.10.2.3 Filename detail

See section 9.9.1

A.10.3 File structures

A.10.3.1 Constraint sets

	CONSTRAINT SET NAME	D	T1.3	T2.3	A1.3	M1.3
1	Unstructured	o	m	m	m	-
2	Sequential Flat	o	i	m	m	-
3	Ordered flat	o	i	o	o	-
4	Ordered flat with unique names	o	i	o	o	-
5	Ordered hierarchical	o	i	i	i	-
6	General hierarchical	o	i	i	i	-
7	General hierarchical with unique names	o	i	i	i	-
8	NBS ordered flat	-	i	o	o	-
9	NBS random access	-	i	o	o	-

A.10.3.2 File and filestore actions

A.10.3.2.1 Filestore Actions

Support for filestore actions is dependent upon the functional units implemented (see A.12.4 and A.12.5)

A.10.3.2.2 File Actions

	RESPONDER	CONSTRAINT SET unstructured	
	ACTION	D	T1.3
1	Locate	———	
2	Read	o	o
3	Insert	———	
4	Replace	o	o
5	Extend	o	o
6	Erase	o	i

	RESPONDER	CONSTRAINT SET: unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACTION	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3
7	Locate	——		o	I	o	I	o	I	-	I	-	I
8	Read	o	o	o	o	o	o	o	o	-	o	-	o
9	Insert	——		o	o	o	o	o	o	-	o	-	o
10	Replace	o	o	——		o	o	o	o	-	o	-	o
11	Extend	o	o	——		o	o	o	o	——		——	
12	Erase	o	I	o	I	o	I	o	I	-	I	-	I

	RESPONDER	CONSTRAINT SET: unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACTION	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3
13	Locate	——		o	o	o	o	o	o	-	o	-	o
14	Read	o	o	o	o	o	o	o	o	-	o	-	o
15	Insert	——		o	o	o	o	o	o	-	o	-	o
16	Replace	o	o	——		o	o	o	o	-	o	-	o
17	Extend	o	o	——		o	o	o	o	——		——	
18	Erase	o	o	o	o	o	o	o	o	-	o	-	o

NOTE - File actions are not defined in Implementation Profile M1.3

A.10.3.2.3 Access contexts supported

	RESPONDER / ACCESS CONTEXT	CONSTRAINT SET: unstructured D	unstructured T1.3
1	US	———	———
2	UA	o	m
3	FS	———	———
4	FL	———	———
5	FA	———	———
6	HN	———	———
7	HA	———	———

	RESPONDER / ACCESS CONTEXT	unstructured D	unstructured T2.3	sequential flat D	sequential flat T2.3	ordered flat D	ordered flat T2.3	ordered flat with unique names D	ordered flat with unique names T2.3	NBS ordered flat D	NBS ordered flat T2.3	NBS random access D	NBS random access T2.3
		CONSTRAINT SET											
8	US	———	———	———	———	———	———	———	———	———	———	———	———
9	UA	o	m	o	m	o	m	o	m	-	m	-	m
10	FS	———	———	———	———	———	———	———	———	———	———	———	———
11	FL	———	———	———	———	———	———	———	———	———	———	———	———
12	FA	———	———	o	m	o	m	o	m	-	m	———	———
13	HN	———	———	———	———	———	———	———	———	———	———	———	———
14	HA	———	———	———	———	o	o	o	o	-	o	———	———

	RESPONDER	CONSTRAINT SET: unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACCESS CONTEXT	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3
15	US	—	—	—	—	—	—	—	—	—	—	—	—
16	UA	o	m	o	m	o	m	o	m	-	m	-	m
17	FS	—	—	—	—	—	—	—	—	—	—	—	—
18	FL	—	—	—	—	—	—	—	—	—	—	—	—
19	FA	—	—	o	m	o	m	o	m	-	m	—	—
20	HN	—	—	—	—	—	—	—	—	—	—	—	—
21	HA	—	—	—	—	o	o	o	o	-	o	—	—

NOTE - The supported access contexts for Impementation Profile M1.3 are defined in the T- or A-Profile in conjunction with which M1.3 is implemented.

A.10.4 Additional information

(Void)

A.10.5 Override

	RESPONDER OVERRIDE	PROFILES D	PROFILES R
1	Create failure	o	m
2	Select old file	o	m
3	Delete and recreate with old attributes	o	o
4	Delete and create with new attributes	o	m

NOTE - The specification of the role of initator is given in section 5 (file protocol detail).

A.11 File protocol

See sections 9.5(1) - (3), 9.17

Clauses A.11.2 to A.11.24 specify an indication of which PDUs are supported. The conformance requirements for PDUs are dependent on the particular functional units implemented. PDUs indicated in clauses A.11.8 to A.11.24 as conditional shall be considered as mandatory when a particular functional unit is implemented, according to the following table.

PDUs	Clause	Functional Units								
		Kernel	Read	Write	Access	LFM	EFM	Grouping	Recovery	Restart
F-CREATE	A.11.8					m				
F-DELETE	A.11.9					m				
F-READ-ATTRIB	A.11.10					m				
F-CHANGE-ATTRIB	A.11.11						m			
F-OPEN	A.11.12		m	m						
F-CLOSE	A.11.13		m	m						
F-BEGIN-GROUP	A.11.14							m		
F-END-GROUP	A.11.15							m		
F-RECOVER	A.11.16								m	
F-LOCATE	A.11.17				m					
F-ERASE	A.11.18				m					
F-READ	A.11.19		m							
F-WRITE	A.11.20			m						
F-DATA-END	A.11.21		m	m						
F-TRANSFER-END	A.11.22		m	m						
F-CANCEL	A.11.23		m	m						
F-RESTART	A.11.24									m

NOTES

1 In order to keep the protocol tables compact some forward references have been introduced to clauses which expand upon the detail of field support.

2 The FTAM protocol will require a number of optional lower layer services to be available (eg Application Entity Titles in ACSE). This requirement is outside the scope of this Profiles Requirements List.

A.11.1 GraphicString support

(Void)

A.11.2 FTAM regime establishment

		D		PROFILES		
		I	R	I	R	
1	F-INITIALIZE PDU	m	m	m	m	
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	State result	-	m	-	m	all values defined in ISO 8571
3	Action result	-	m	-	m	all values defined in ISO 8571
4	Protocol version	m	m	m	m	see Section 2
5	Implementation information	o	o	o	o	see A.12.1
6	Presentation context management	m	m	m	m	see note 1, 9.17.10
7	Service class	m	m	m	m	see A.12.4
8	Functional units	m	m	m	m	see A.12.5
9	Attribute groups	m	m	m	m	see A.10.2
10	Shared ASE information	o	o	I	I	see 9.5(8)
11	FTAM Quality of Service	m	m	m	m	see A.12.8
12	Contents type list	o	o	m	m	see A.12.7.1, 9.18.4
13	Initiator identity	o	-	m	-	see 9.16.1, 9.18.4, 10.8.8
14	Account	o	-	o	-	see 9.18.4, 10.8.8
15	Filestore password	o	-	m	-	see A.12.11, 9.16.1, 10.8.8
16	Diagnostic	-	o	-	m	see A.12.6, 9.13, 10.8.5.2
17	Checkpoint window	m	m	m	m	see note 2, 10.8.5.2

NOTES
1 The values available for the presentation context management field depend upon the functional units implemented in ISO 8823.
2 Checkpoint window field is indicated as mandatory in accordancqwith ISO 8571-4. The field is defaulted to the value 1

A.11.3 FTAM regime termination (orderly)

		D		PROFILES		
		I	R	I	R	
1	F-TERMINATE PDU	m	m	m	m	
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	Shared ASE information	o	o	I	I	see 9.5 (8)
3	Charging	-	o	-	o	see A.12.10

A.11.4 FTAM regime termination (abrupt) by service user

		D	PROFILES	
1	F-U-ABORT PDU	m	m	
	FIELD NAME			RANGE OF VALUES OR REFERENCE
2	Action result	m	m	all values defined in ISO 8571
3	Diagnostic	o	m	see A.12.6, 9.13

A.11.5 FTAM regime termination (abrupt) by service provider

		D	PROFILES	
1	F-P-ABORT PDU	m	m	
	FIELD NAME			RANGE OF VALUES OR REFERENCE
2	Action result	m	m	all values defined in ISO 8571
3	Diagnostic	o	m	see A.12.6, 9.13, 10.8.5.2

A.11.6 File selection

		D		PROFILES		
		I	R	I	R	
1	F-SELECT PDU	m	m	m	m	
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	State result	-	m	-	m	all values defined in ISO 8571
3	Action result	-	m	-	m	all values defined in ISO 8571
4	Attributes	m	m	m	m	see A.10.2, 9.17.9
5	Requested access	m	-	m	-	see A.12.16
6	Access passwords	o	-	m	-	see 9.16.2, 10.8.8
7	Concurrency control	o	-	m	-	see A.12.13, 10.8.6.1
8	Shared ASE information	o	o	I	I	see 9.5(8)
9	Account	o	-	o	-	see 9.18.4, 10.8.8
10	Diagnostic	-	o	-	m	see A.12.6, 9.13

A.11.7 File deselection

		D		PROFILES		
		I	R	I	R	
1	F-DESELECT PDU	m	m	m	m	
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	Action result	-	m	-	m	all values defined in ISO 8571
3	Charging	-	o	-	o	see A.12.10
4	Shared ASE information	o	o	I	I	see 9.5(8)
5	Diagnostic	-	o	-	m	see A.12.6, 9.13

A.11.8 File creation

		D I	D R	PROFILES I	PROFILES R	
1	F-CREATE PDU	c	c	c	c	see A.11, A.12.5
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	State result	-	m	-	m	all values defined in ISO 857
3	Action result	-	m	-	m	all values defined in ISO 8571
4	Override	m	-	m	-	see A.12.15
5	Initial attributes	m	m	m	m	see A.10.2, 9.10.2.2, 9.17.9
6	Create password	o	-	m	-	see A.12.12, 9.16.2, 10.8.7 10.8.8
7	Requested access	m	-	m	-	see A.12.16
8	Access passwords	o	-	m	-	see 9.16.2, 10.8.8
9	Concurrency control	o	-	m	-	see A.12.13, 10.8.6.1
10	Shared ASE information	o	o	i	i	see 9.5(8)
11	Account	o	-	o	-	see 9.18.4, 10.8.8
12	Diagnostic	-	o	-	m	see A.12.6, 9.13

A.11.9 File deletion

		D I	D R	PROFILES I	PROFILES R	
1	F-DELETE PDU	c	c	c	c	see A.11, A.12.5
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	Action result	-	m	-	m	all values defined in ISO 857
3	Shared ASE information	o	o	i	i	
4	Charging	-	o	-	o	see A.12.10
5	Diagnostic	-	o	-	m	see A.12.6, 9.13

A.11.10 Read attributes

		D		PROFILES		
		I	R	I	R	
1	F-READ-ATTRIB PDU	c	c	c	c	see A.11, A.12.5
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	Action result	-	m	-	m	all values defined in ISO 8571
3	Attribute names	m	-	m	-	
4	Attributes	-	o	-	m	see A.10.2, 9.17.9
5	Diagnostic	-	o	-	m	see A12.6, 9.13

A.11.11 Change attributes

		D		T1.3, T2.3, A1.3	M1.3		
		I	R		I	R	
1	F-CHANGE-ATTRIB	c	c	i	m	m	see A.11, A.12.5
	FIELD NAME						RANGE OF VALUES OR REFERENCE
2	Action result	-	m	i	-	m	all values defined in ISO 8571
3	Attributes	m	o	i	m	m	see A.10.2, 9.17.9
4	Diagnostic	-	o	i	-	m	see A.12.6, 9.13

A.11.12 File open

		D		T1.3, T2.3, A1.3		M1.3	
		I	R	I	R		
1	F-OPEN PDU	c	c	m	m	i	see A.11, A.12.5
	FIELD NAME						RANGE OF VALUES OR REFERENCE
2	State result	-	m	-	m	i	all values defined in ISO 8571
3	Action result	-	m	-	m	i	all values defined in ISO 8571
4	Processing mode	m	-	m	-	i	see A.12.17
5	Contents type	m	m	m	m	i	see A.12.7.2
6	Concurrency control	o	o	m	o	i	see A.12.13, 10.8.6.1
7	Shared ASE information	o	o	i	i	i	see 9.5(8)

8	Enable FADU locking	m -	m -	I	'false' for T1.3 and T2.3
9	Activity identifier	o -	o -	I	
10	Diagnostic	- o	- m	I	see A.12.6, 9.13
11	Recovery mode	m m	m m	I	see A. 12.18
12	Remove contexts	o -	I -	I	
13	Define contexts	o -	I -	I	
14	Presentation action	- m	- m	I	see notes

NOTES
1 The values available for the presentation action field depend upon the functional units implemented in ISO 8823.
2 Presentation action field is indicated as mandatory in accordance with ISO 8571-4. The field is defaulted to no action.

A.11.13 File close

		D	T1.3, T2.3, A1.3	M1.3	
1	F-CLOSE PDU	c	m	I	see A.11, A.12.5
	FIELD NAME				RANGE OF VALUES OR REFERENCE
2	Action result	m	m	I	all values defined in ISO 8571
3	Shared ASE information	o	I	I	see 9.5(8)
4	Diagnostic	o	m	I	see A.12.6, 9.13

A.11.14 Beginning of grouping

		D		PROFILES		
		I	R	I	R	
1	F-BEGIN-GROUP PDU	c	c	c	c	see A.11, A.12.5
	FIELD NAME					RANGE OF VALUES OR REFERENCE
2	Threshold	m	-	m	-	

A.11.15 End of grouping

		D	PROFILES	
1	F-END-GROUP PDU	c	c	see A.11, A.12.5
	The F-END-GROUP PDU carries no fields.			

A.11.16 Regime recovery

December 1989

See section 10.8.5

		D I	D R	T1.3, T2.3, A1.3 I	T1.3, T2.3, A1.3 R	M1.3	
1	F-RECOVER PDU	c	c	c	c	i	see A.11, A.12.5
	FIELD NAME						RANGE OF VALUES OR REFERENCE
2	State result	-	m	-	m	i	all values defined in ISO 8571
3	Action result	-	m	-	m	i	all values defined in ISO 8571
4	Activity identifier	m	-	m	-	i	
5	Bulk transfer number	m	-	m	-	i	see 10.9
6	Requested access	m	-	m	-	i	see A.12.16
7	Access passwords	o	-	m	-	i	see 9.16.2, 10.8.8
8	Contents type	-	m	-	m	i	see A.12.7.2
9	Recovery point	m	m	m	m	i	
10	Diagnostic	-	o	-	m	i	see A.12.6, 9.13, 10.8.5.2
11	Remove contexts	o	-	i	-	i	see notes
12	Define contexts	o	-	i	-	i	see notes
13	Presentation action	-	m	-	m	i	see notes

NOTES

1 The values available for the presentation action field depend upon the functional units implemented in ISO 8823.

2 Presentation action field is indicated as mandatory in accordance with ISO 8571-4. The field is defaulted to no action.

A.11.17 Locate file access data unit

		D I	D R	T1.3, T2.3	A1.3 I	A1.3 R	M1.3	
1	F-LOCATE PDU	c	c	i	m	m	i	see A.11, A.12.5
	FIELD NAME							RANGE OF VALUES OR REFERENCE
2	Action result	-	m	i	-	m	i	all values defined in ISO 8571
3	FADU identity	m	o	i	m	o	i	see 9.17.9
4	FADU lock	o	-	i	o	-	i	see A.12.14
5	Diagnostic	-	o	i	-	m	i	see A.12.6, 9.13

A.11.18 Erase file access data unit

		D		T1.3, T2.3	A1.3		M1.3	
		I	R		I	R		
1	F-ERASE PDU	c	c	i	m	m	i	see A.11, A.12.5
	FIELD NAME							RANGE OF VALUES OR REFERENCE
2	Action result	-	m	i	-	m	i	all values defined in ISO 8571
3	FADU identity	m	-	i	m	-	i	see 9.17.9
4	Diagnostic	-	o	i	-	m	i	see A.12.6, 9.13

A.11.19 Read bulk data

		D		T1.3, T2.3		A1.3		M1.3	
		I	R	I	R	I	R		
1	F-READ PDU	c	c	c	c	m	m	i	see A.11, A.12.5
	FIELD NAME								RANGE OF VALUES OR REFERENCE
2	FADU identity	m	-	m	-	m	-	i	see 9.17.9
3	Access context	m	-	m	-	m	-	i	see A.10.3.2.3
4	FADU lock	o	-	i	-	o	-	i	

A.11.20 Write bulk data

		D		T1.3, T2.3		A1.3		M1.3	
		I	R	I	R	I	R		
1	F-WRITE PDU	c	c	c	c	m	m	i	see A.11, A.12.5
	FIELD NAME								RANGE OF VALUES OR REFERENCE
2	FADU operation	m	-	m	-	m	-	i	
3	FADU identity	m	-	m	-	m	-	i	see 9.17.9
4	FADU Lock	o	-	i	-	o	-	i	

A.11.21 End of data transfer

		D	T1.3, T2.3, A1.3	M1.3	
1	F-DATA-END PDU	c	m	i	see A.11, A.12.5
	FIELD NAME				RANGE OF VALUES OR REFERENCE
2	Action result	m	m	i	all values defined in ISO 8571
3	Diagnostic	o	m	i	see A.12.6, 9.13

A.11.22 End of transfer

		D		T1.3, T2.3, A1.3		M1.3	
		I	R	I	R		
1	F-TRANSFER-END PDU	c	c	m	m	i	see A.11, A.12.5
	FIELD NAME						RANGE OF VALUES OR REFERENCE
2	Action result	-	m	-	m	i	all values defined in ISO 8571
3	Shared ASE information	o	o	i	i	i	see 9.5(8)
4	Diagnostic	-	o	-	m	i	see A.12.6, 9.13

A.11.23 Cancel data transfer

See section 9.11

		D	T1.3, T2.3, A1.3	M1.3	
1	F-CANCEL PDU	c	m	i	see A.11, A.12.5
	FIELD NAME				RANGE OF VALUES OR REFERENCE
2	Action result	m	m	i	all values defined in ISO 8571
3	Shared ASE information	o	i	i	see 9.5(8)
4	Diagnostic	o	m	i	see A.12.6, 9.13

A.11.23.1 F-CANCEL mapping

See sections 9.11, 9.17.10

A.11.24 Restart data transfer

		D	T1.3, T2.3, A1.3	M1.3	
1	F-RESTART PDU	c	c	i	see A.11, A.12.5
	FIELD NAME				RANGE OF VALUES OR REFERENCE
2	Checkpoint identifier	m	m	i	

A.12 Expanded PDU field and filestore detail

This clause identifies further PDU field and filestore detail to expand on that given in A.10 and A.11.

A.12.1 Implementation information detail

See sections 9.5(6), 9.12, 10.8.3

A.12.2 Access control detail

See sections 9.9.2, 10.7.3

	Access control element terms	D	PROFILES	RANGE OF VALUES
1	Action list	m	m	
2	Concurrency access	o	o	see A.12.3.3
3	Identity	o	o	
4	Passwords	o	o	see A.12.3.5, A.12.3.6, 10.8.8
5	Location	o	o	

A.12.3 Access control element detail

A.12.3.1 Action list detail (initiator)

(Void)

A.12.3.2 Action list detail (responder)

(Void)

A.12.3.3 Concurrency access term

If the concurrency access term is supported in the access control element the following details of the concurrency control shall be available with each action.

	T1.3 Action	not required D	not required T1.3	shared D	shared T1.3	exclusive D	exclusive T1.3	no access D	no access T1.3
1	Read	o	o	o	o	o	o	o	o
2	Insert	o	i	o	i	o	i	o	i
3	Replace	o	o	o	o	o	o	o	o
4	Extend	o	o	o	o	o	o	o	o
5	Erase	o	i	o	i	o	i	o	i
6	Read attributes	o	o	o	o	o	o	o	o
7	Change attributes	o	i	o	i	o	i	o	i
8	Delete file	o	o	o	o	o	o	o	o

	T2.3 Action	not required D	not required T2.3	shared D	shared T2.3	exclusive D	exclusive T2.3	no access D	no access T2.3
9	Read	o	o	o	o	o	o	o	o
10	Insert	o	o	o	o	o	o	o	o
11	Replace	o	o	o	o	o	o	o	o
12	Extend	o	o	o	o	o	o	o	o
13	Erase	o	i	o	i	o	i	o	i
14	Read attributes	o	o	o	o	o	o	o	o
15	Change attributes	o	i	o	i	o	i	o	i
16	Delete file	o	o	o	o	o	o	o	o

	A1.3 Action	not required D	not required A1.3	shared D	shared A1.3	exclusive D	exclusive A1.3	no access D	no access A1.3
17	Read	o	o	o	o	o	o	o	o
18	Insert	o	o	o	o	o	o	o	o
19	Replace	o	o	o	o	o	o	o	o
20	Extend	o	o	o	o	o	o	o	o
21	Erase	o	o	o	o	o	o	o	o
22	Read attributes	o	o	o	o	o	o	o	o
23	Change attributes	o	I	o	I	o	I	o	I
24	Delete file	o	o	o	o	o	o	o	o

	M1.3 Action	not required D	not required M1.3	shared D	shared M1.3	exclusive D	exclusive M1.3	no access D	no access M1.3
25	Read	o	I	o	I	o	I	o	I
26	Insert	o	I	o	I	o	I	o	I
27	Replace	o	I	o	I	o	I	o	I
28	Extend	o	I	o	I	o	I	o	I
29	Erase	o	I	o	I	o	I	o	I
30	Read attributes	o	o	o	o	o	o	o	o
31	Change attributes	o	o	o	o	o	o	o	o
32	Delete file	o	o	o	o	o	o	o	o

A.12.3.4 Identity term

(Void)

A.12.3.5 Initiator access passwords

If the passwords term of the access control element is implemented the following values shall be supported for the initiator role.

See section 9.16.3

	Initiator Access Passwords	D	PROFILES I
1	OctetString	o	o
2	GraphicString	o	o

A.12.3.6 Responder access passwords

If the passwords term of the access control element is implemented the following values shall be supported for the responder role.

See section 9.16.3

	Responder Access Passwords	D	T1.3 OctetString GraphicString	T2.3 OctetString GraphicString	A1.3 OctetString GraphicString	M1.3 OctetString GraphicString
1	Read-password	o	o	o	o	I
2	Insert-password	o	I	o	o	I
3	Replace-password	o	o	o	o	I
4	Extend-password	o	o	o	o	I
5	Erase-password	o	I	I	o	I
6	Read-attribute-password	o	o	o	o	o
7	Change-attribute-password	o	I	I	I	o
8	Delete-password	o	o	o	o	o

A.12.3.7 Location term

(Void)

A.12.3.7.1 Application Entity Titles detail

See section 9.5(7)

A.12.3.8 Access control element combinations

	Combinations			D	PROFILES R
1	Identity	Password	Location	o	o
2	Identity	Password		o	o
3	Identity		Location	o	o
4		Password	Location	o	o
5	Identity			o	o
6		Password		o	o
7			Location	o	o

NOTE - Implementation of access control without any of the above combinations is valid.

A.12.4 Service class field detail

See table 9.7 and sections 10.5.1, 10.8.1

		D	T1.3, T2.3	A1.3	M1.3 (T)	M1.3 (A)
1	Transfer class	o	m	i	m	i
2	Access class	o	i	m	i	m
3	Management class	o	i	i	m	m
4	Transfer and management class	o	o	i	m	i
5	Unconstrained class	o	i	i	i	i

NOTES
1 The initiator is only permitted to specify those combinations defined in ISO 8571-3
2 The notation M1.3(T) indicates M1.3 combined with a Transfer Profile T1.3 or T2.3. M1.3(A) means M1.3 combined with the Access Profile A1.3.

A.12.5 Functional unit field detail

See table 9.7 and sections 10.8.1, 10.8.2

	T1.3, T2.3	SERVICE CLASSES			
		Transfer		Transfer and Management	
	FUNCTIONAL UNITS	D	T1.3, T2.3	D	T1.3, T2.3
1	Kernel	m	m	m	m
2	Read (see note 2)	c	o	c	o
3	Write (see note 2)	c	o	c	o
4	File Access	———		———	
5	Limited File Management	o	o	m	m
6	Enhanced File Management	o	i	o	i
7	Grouping	m	m	m	m
8	FADU Locking	———		———	
9	Recovery	o	o	o	o
10	Restart	o	o	o	o

NOTES
1 the recovery and the restart functional units are only available at the internal file service interface and should only be explicitly referenced in the protocol.
2 the c indicates that either or both of the read and write functional units shall be implemented in the particular service class.

	A1.3	SERVICE CLASSES		
		Access		
	FUNCTIONAL UNITS	D	A1.3	
11	Kernel	m	m	
12	Read	m	m	
13	Write	m	m	
14	File Access	m	m	
15	Limited File Management	o	o	
16	Enhanced File Management	c	i	
17	Grouping	o	o	
18	FADU Locking	o	o	see 10.8.6.2
19	Recovery	o	o	
20	Restart	o	o	

See 10.8.1

	M1.3(T)	SERVICE CLASSES					
		Transfer		Management		Transfer and Management	
	FUNCTIONAL UNITS	D	M1.3(T)	D	M1.3(T)	D	M1.3(T)
21	Kernel			m	m	m	m
22	Read			———		c	o
23	Write			———		c	o
24	File Access			———		———	
25	Limited File Management	o	m	m	m	m	m
26	Enhanced File Management	o	m	o	m	o	m
27	Grouping			m	m	m	m
28	FADU Locking			———		———	
29	Recovery			———		o	o
30	Restart			———		o	o

NOTE - M1.3(T) indicates M1.3 in conjunction with a Transfer Profile T1.3 or T2.3. This table lists only the additional functionality as defined by M1.3.

See 10.8.1

	M1.3(A) SERVICE CLASSES				
		Access		Management	
	FUNCTIONAL UNITS	D	M1.3(A)	D	M1.3(A)
31	Kernel			m	m
32	Read			———	
33	Write			———	
34	File Access			———	
35	Limited File Management	o	m	m	m
36	Enhanced File Management	o	m	o	m
37	Grouping			m	m
38	FADU Locking			———	
39	Recovery			———	
40	Restart			———	

NOTE - M1.3(A) indicates M1.3 in conjunction with the Access Profile A1.3. This table lists only the additional functionality as defined by M1.3.

A.12.6 Diagnostic field detail

		D	T1.3, T2.3, A1.3	M1.3	
1	Diagnostic type	m	m	m	
2	Error identifier	m	m	m	
3	Error observer	m	m	m	
4	Error source	m	m	m	
5	Suggested delay	o	c	I	see 10.8.5.2
6	Further details	o	m	m	
	For values of the 'further details' term only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required (see sec. 9.13).				

A.12.7 Contents type detail

A.12.7.1 Contents type list parameter

See section 9.10.2.1

		D	PROFILES I	PROFILES R	Maximum number of elements
1	document type specifications	o	o	m	
2	abstract syntax specifications	o	o	m	

A.12.7.2 Contents type parameter

See section 9.10.2.3

		D	PROFILES	REFERENCE
1	document type specifications	o	m	see 9.9.1
2	abstract syntax / constraint set pair specifications	o	I	

NOTE - The detail of document types supported is contained in clause A.13.

A.12.8 FTAM Quality of service details

See section 10.8.5.2

A.12.9 Details of shared ASE information

(Void)

A.12.10 Details of charging

See section 9.5(8), 9.18.4

	Charging	D	PROFILES R
1	Resource identifier term	m	m
2	Charging unit term	m	m
3	Charging value term	m	m

A.12.11 Filestore password detail

	Filestore password detail	D	PROFILES
1	OctetString	o	o
2	GraphicString	o	o

A.12.12 Create password detail

See section 9.16.3

	Create password detail	D	PROFILES
1	OctetString	o	o
2	GraphicString	o	o

A.12.13 Concurrency control

A.12.13.1 Supported values

See section 10.8.6.1

	T1.3	not required		shared		exclusive		no access	
	Action	D	T1.3	D	T1.3	D	T1.3	D	T1.3
1	Read	o	o	o	o	o	o	o	o
2	Insert	o	i	o	i	o	i	o	i
3	Replace	o	o	o	o	o	o	o	o
4	Extend	o	o	o	o	o	o	o	o
5	Erase	o	i	o	i	o	i	o	i
6	Read attrib	o	o	o	o	o	o	o	o
7	Change attrib	o	i	o	i	o	i	o	i
8	Delete file	o	o	o	o	o	o	o	o

	T2.3	not required		shared		exclusive		no access	
	Action	D	T2.3	D	T2.3	D	T2.3	D	T2.3
9	Read	o	o	o	o	o	o	o	o
10	Insert	o	o	o	o	o	o	o	o
11	Replace	o	o	o	o	o	o	o	o
12	Extend	o	o	o	o	o	o	o	o
13	Erase	o	I	o	I	o	I	o	I
14	Read attrib	o	o	o	o	o	o	o	o
25	Change attrib	o	I	o	I	o	I	o	I
16	Delete file	o	o	o	o	o	o	o	o

	A1.3	not required		shared		exclusive		no access	
	Action	D	A1.3	D	A1.3	D	A1.3	D	A1.3
17	Read	o	o	o	o	o	o	o	o
18	Insert	o	o	o	o	o	o	o	o
19	Replace	o	o	o	o	o	o	o	o
20	Extend	o	o	o	o	o	o	o	o
21	Erase	o	o	o	o	o	o	o	o
22	Read attrib	o	o	o	o	o	o	o	o
23	Change attrib	o	I	o	I	o	I	o	I
24	Delete file	o	o	o	o	o	o	o	o

	M1.3	not required		shared		exclusive		no access	
	Action	D	M1.3	D	M1.3	D	M1.3	D	M1.3
25	Read	o	I	o	I	o	I	o	I
26	Insert	o	I	o	I	o	I	o	I
27	Replace	o	I	o	I	o	I	o	I
28	Extend	o	I	o	I	o	I	o	I
29	Erase	o	I	o	I	o	I	o	I
30	Read attrib	o	o	o	o	o	o	o	o
31	Change attrib	o	o	o	o	o	o	o	o
32	Delete file	o	o	o	o	o	o	o	o

A.12.13.2 Responder Default values

See sections 9.14, 10.8.6.1

A.12.14 FADU Locking

	A1.3	FADU Locking Support Values							
		not required		shared		exclusive		no access	
		D	A1.3	D	A1.3	D	A1.3	D	A1.3
1	Read	o	o	o	o	o	o	o	o
2	Insert	o	o	o	o	o	o	o	o
3	Replace	o	o	o	o	o	o	o	o
4	Extend	o	o	o	o	o	o	o	o
5	Erase	o	o	o	o	o	o	o	o

A.12.15 Initiator Override

	Initiator override	D	PROFILES I
1	Create failure	o	o
2	Select old file	o	o
3	Delete and recreate with old attributes	o	o
4	Delete and create with new attributes	o	o

NOTE - The specification of the role of responder is given in the filestore section

A.12.16 Requested Access

See section 9.15

	Action	D	T1.3	T2.3	A1.3	M1.3
1	Read	o	o	o	o	i
2	Insert	o	i	o	o	i
3	Replace	o	o	o	o	i
4	Extend	o	o	o	o	i
5	Erase	o	x	x	o	i
6	Read attribute	o	o	o	o	m
7	Change attribute	o	i	i	i	m
8	Delete file	o	o	o	o	m

A.12.17 Processing mode

	Processing mode	D	T1.3	T2.3	A1.3	M1.3
1	Read	o	o	o	o	i
2	Insert	o	i	o	o	i
3	Replace	o	o	o	o	i
4	Extend	o	o	o	o	i
5	Erase	o	x	x	o	i

A.12.18 Recovery mode

See section 10.8.5.2

	Recovery mode	D	T1.3, T2.3, A1.3	M1.3
1	None	o	m	i
2	At start of transfer	o	m	i
3	Any active checkpoint	o	o	i

Section 6 : Document Type Detail

A.13 Document types

See section 10.7.1

Conformance to document types is given at two levels. The following table indicates which document types have some level of support. The detail of that level of support is stated in the following tables.

	Entry number	FTAM-1	D	T1.3	T2.3	A1.3	M1.3
1	Object descriptor	ISO FTAM unstructured text	o	m	m	m	i
	Object identifier	{iso standard 8571 document-type(5) unstructured-text(1)}			see A.13.1		

	Entry number	FTAM-2	D	T1.3	T2.3	A1.3	M1.3
2	Object descriptor	ISO FTAM sequential text	o	i	m	m	i
	Object identifier	{iso standard 8571 document-type(5) sequential-text(2)}			see A.13.2		

	Entry number	FTAM-3	D	T1.3	T2.3	A1.3	M1.3
3	Object descriptor	ISO FTAM unstructured binary	o	m	m	m	i
	Object identifier	{iso standard 8571 document-type(5) unstructured-binary(3)}			see A.13.3		

	Entry number	FTAM-4	D	T1.3	T2.3	A1.3	M1.3
4	Object descriptor	ISO FTAM sequential binary	o	i	o	o	i
	Object identifier	{iso standard 8571 document-type(5) sequential-binary(4)}			see A.13.4		

	Entry number	NBS-6	D	T1.3	T2.3	A1.3	M1.3
5	Object descriptor	NBS-6 FTAM sequential file	-	i	o	o	i
	Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) sequential(6) }			see A.13.5		

	Entry number	NBS-7	D	T1.3	T2.3	A1.3	M1.3
6	Object descriptor	NBS-7 FTAM random access file	-	i	o	o	i
	Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) random-file(7) }			see A.13.6		

Entry number	NBS-8	D	T1.3	T2.3	A1.3	M1.3
7 Object descriptor	NBS-8 FTAM indexed file	-	i	o	o	i
Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file(8) }			see A.13.7		

Entry number	NBS-9	D	T1.3	T2.3	A1.3	M1.3
8 Object descriptor	NBS-9 FTAM file directory file	-	o	o	o	i
Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) file-directory(9) }			see 9.18.3		

Entry number	NBS-10	D	T1.3	T2.3	A1.3	M1.3
9 Object descriptor	NBS-10 FTAM random binary access file	-	i	o	o	i
Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) random-binary(10) }			see 10.7.1		

Entry number	NBS-11	D	T1.3	T2.3	A1.3	M1.3
10 Object descriptor	NBS-11 FTAM indexed file with unique keys	-	i	o	o	i
Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file-with-unique-keys(11) }			see A.13.8		

Entry number	NBS-12	D	T1.3	T2.3	A1.3	M1.3
11 Object descriptor	NBS-12 NBS FTAM simple text file	-	i	o	o	i
Object identifier	{iso identified-organization oiw(14) ftamsig(5) document-type(5) simple-text-file(12) }			see A.13.9		

Constraint sets and FADU identities for document types

For the constraint set / FADU identity tables the following notation is used:

m	mandatory	in the constraint set definition, or optional in the constraint set definition but shall be implemented by implementations claiming conformance to the Profile. The support of the FADU identity will be dependent on the actions which have been implemented.
o	optional	in the constraint set definition
i	not supported	(outside the scope of this ISP, may be ignored)
-	not applicable	(not defined in the constraint set definition)
x	excluded	(disallowed in the document type definition or in FTAM Phase 3)

Implementation Profile T1.3.

FADU Identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM-1	-	-	m	-	-	-	-	-	-
FTAM-3	-	-	m	-	-	-	-	-	-
NBS-9	-	-	m	-	-	-	-	-	-

Implementation Profile T2.3 (see secs. 9.10, 10.7.2)

FADU Identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM - 1	-	-	m	-	-	-	-	-	-
FTAM - 3	-	-	m	-	-	-	-	-	-
NBS-9	-	-	m	-	-	-	-	-	-
FTAM sequential flat constraint set	o	o	o	o	o	o	o	-	o
FTAM-2	m	m	I	I	I	I	I	-	I
FTAM-4	m	m	I	I	I	I	I	-	I
NBS-6	m	m	I	x	x	I	x	-	x
NBS-12	m	m	x	x	x	x	x	-	x
FTAM ordered flat constraint set	o	o	o	o	o	o	o	o	o
NBS-8	m	I	I	I	I	I	I	m	I
FTAM ordered flat constraint set with unique names	o	o	-	-	o	o	o	o	o
NBS-11	m	I	-	-	I	I	I	m	I
NBS ordered flat constraint set	o	o	o	o	o	o	o	-	o
NBS-7	m	m	m	m	I	I	I	-	m
NBS random access constraint set	o	o	-	-	-	-	-	o	o
NBS-10	m	m	-	-	-	-	-	m	m

Implementation Profile A1.3 (see section 9.10)

FADU identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM-1	-	-	m	-	-	-	-	-	-
FTAM-3	-	-	m	-	-	-	-	-	-
NBS-9	-	-	m	-	-	-	-	-	-
FTAM sequential flat constraint set	o	o	o	o	o	o	o	-	o
FTAM-2	m	m	m	i	i	m	i	-	i
FTAM-4	m	m	m	i	i	m	i	-	i
NBS-6	m	m	m	x	x	m	x	-	x
NBS-12	m	m	m	x	x	m	x	-	x
FTAM ordered flat constraint set	o	o	o	o	o	o	o	o	o
NBS-8	m	m	i	i	m	m	m	m	i
FTAM ordered flat constraint set with unique names	o	o	-	-	o	o	o	o	o
NBS-11	m	m	-	-	m	m	m	m	i
NBS ordered flat constraint set	o	o	o	o	o	o	o	-	o
NBS-7	m	m	m	m	m	m	m	-	m
NBS random access constraint set	o	o	-	-	-	-	-	o	o
NBS-10	m	m	-	-	-	-	-	m	m

A.13.1 FTAM-1 (See section 10.7.1)

A.13.1.1 Universal class number parameter (See section 9.10.1)

		D	T1.3, T2.3, A1.3	
1	Universal class number parameter supported	o	m	
2	PrintableString - Universal class 19	o	I	
3	TeletexString - Universal class 20	o	I	
4	VideotexString - Universal class 21	o	I	
5	IA5String - Universal class 22	o	m	see 9.10.1.1-2
6	GraphicString - Universal class 25	o	m	see A.13.1.3
7	VisibleString - Universal class 26	o	m	
8	GeneralString - Universal class 27	o	m	see A.13.1.4

A.13.1.2 String length parameter and string significance parameter combinations

		D	T1.3, T2.3, A1.3
1	Maximum string length parameter and variable length strings	o	m
2	Maximum string length parameter and fixed length strings	o	m
3	Maximum string length parameter and not significant strings	o	m
4	Unbounded strings and variable length strings	o	m
5	Unbounded strings and not significant strings	o	m

A.13.1.3 G sets supported

G sets which are supported in FTAM-1 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3)**

A.13.1.4 G and C sets supported

G and C sets which are supported in FTAM-1 GeneralString

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets and ISO 646 IRV (C0) control character set is required (see 9.10.1-3)**

A.13.2 FTAM-2 (see section 10.7.1)

A.13.2.1 Universal class number parameter (see section 9.10.1)

		D	T2.3, A1.3	
1	Universal class number parameter supported	o	m	
2	PrintableString - Universal class 19	o	i	
3	TeletexString - Universal class 20	o	i	
4	VideotexString - Universal class 21	o	i	
5	IA5String - Universal class 22	o	o	see 9.10.1.1-2
6	GraphicString - Universal class 25	o	m	see A.13.2.3
7	VisibleString - Universal class 26	o	m	
8	GeneralString - Universal class 27	o	o	see A.13.2.4

A.13.2.2 String length parameter and string significance parameter combinations

		D	T2.3, A1.3
1	Maximum string length parameter and variable length strings	o	i
2	Maximum string length parameter and fixed length strings	o	i
3	Maximum string length parameter and not significant strings	o	m
4	Unbounded strings and variable length strings	o	i
5	Unbounded strings and not significant strings	o	m

A.13.2.3 G sets supported

G sets which are supported in FTAM-2 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3)**

A.13.2.4 G and C sets supported

G and C sets which are supported in FTAM-2 GeneralString

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets and ISO 646 IRV (C0) control character set is required.** **(see 9.10.1.1-3)**

A.13.3 FTAM-3

A.13.3.1 String length parameter and string significance parameter combinations (see section 10.7.1)

		D	T1.3, T2.3, A1.3
1	Maximum string length parameter and variable length strings	o	**I**
2	Maximum string length parameter and fixed length strings	o	**I**
3	Maximum string length parameter and not significant strings	o	**m**
4	Unbounded strings and variable length strings	o	**I**
5	Unbounded strings and not significant strings	o	**m**

A.13.4.1 String length parameter and string significance parameter combinations

		D	T2.3, A1.3
1	Maximum string length parameter and variable length strings	o	i
2	Maximum string length parameter and fixed length strings	o	i
3	Maximum string length parameter and not significant strings	o	m
4	Unbounded strings and variable length strings	o	i
5	Unbounded strings and not significant strings	o	m

See tables 9.2, 9.3

A.13.5.1 Parameter0

				D	T2.3, A1.3
1	Parameter0 supported			-	m
2	Universal-time	-	Universal class 23	-	m
3	Generalized-time	-	Universal class 24	-	m
4	boolean	-	Universal class 1	-	m
5	null	-	Universal class 5	-	m

A.13.5.2 Parameter1 (see section 9.10.1)

				D	T2.3, A1.3
1	Parameter1 supported			-	m
2	integer	-	Universal class 2	-	m
3	bit	-	Universal class 3	-	m
4	IA5	-	Universal class 22	-	m
5	GraphicString	-	Universal class 25	-	m
6	GeneralString	-	Universal class 27	-	m
7	OctetString	-	Universal class 4	-	m

A.13.5.3 Parameter2

		D	T2.3, A1.3
1	Parameter2 supported	-	o

See tables 9.2, 9.3

A.13.6.1 Parameter0

				D	T2.3, A1.3
1	Parameter0 supported			-	m
2	Universal-time	-	Universal class 23	-	m
3	Generalized-time	-	Universal class 24	-	m
4	boolean	-	Universal class 1	-	m
5	null	-	Universal class 5	-	m

A.13.6.2 Parameter1 (see section 9.10.1)

				D	T2.3, A1.3
1	Parameter1 supported			-	m
2	integer	-	Universal class 2	-	m
3	bit	-	Universal class 3	-	m
4	IA5	-	Universal class 22	-	m
5	GraphicString	-	Universal class 25	-	m
6	GeneralString	-	Universal class 27	-	m
7	OctetString	-	Universal class 4	-	m

A.13.6.3 Parameter2

		D	T2.3, A1.3
1	Parameter2 supported	-	o

See tables 9.2, 9.3

A.13.7.1 Parameter0

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter0 supported			-	m	-	m
2	Universal-time	-	Universal class 23	-	m	-	m
3	Generalized-time	-	Universal class 24	-	m	-	m
4	boolean	-	Universal class 1	-	m	-	-
5	null	-	Universal class 5	-	m	-	-

A.13.7.2 Parameter1 (see section 9.10.1)

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter1 supported			-	m	-	m
2	integer	-	Universal class 2	-	m	-	m
3	bit	-	Universal class 3	-	m	-	-
4	IA5	-	Universal class 22	-	m	-	m
5	GraphicString	-	Universal class 25	-	m	-	m
6	GeneralString	-	Universal class 27	-	m	-	m
7	OctetString	-	Universal class 4	-	m	-	m

A.13.7.3 Parameter2

		Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter2 supported	-	o	-	o

See tables 9.2, 9.3

A.13.8.1 Parameter0

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter0 supported			-	m	-	m
2	Universal-time	-	Universal class 23	-	m	-	m
3	Generalized-time	-	Universal class 24	-	m	-	m
4	boolean	-	Universal class 1	-	m	-	-
5	null	-	Universal class 5	-	m	-	-

A.13.8.2 Parameter1 (see section 9.10.1)

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter1 supported			-	m	-	m
2	integer	-	Universal class 2	-	m	-	m
3	bit	-	Universal class 3	-	m	-	-
4	IA5	-	Universal class 22	-	m	-	m
5	GraphicString	-	Universal class 25	-	m	-	m
6	GeneralString	-	Universal class 27	-	m	-	m
7	OctetString	-	Universal class 4	-	m	-	m

A.13.8.3 Parameter2

		Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter2 supported	-	o	-	o

A.13.9 NBS-12 (see section 10.7.1)

A.13.9.1 Universal class number parameter (see section 9.10.1)

				D	T2.3, A1.3	
1	Universal class number parameter supported			-	m	
2	PrintableString	-	Universal class 19	-	I	
3	TeletexString	-	Universal class 20	-	I	
4	VideotexString	-	Universal class 21	-	I	
5	IA5String	-	Universal class 22	-	m	
6	GraphicString	-	Universal class 25	-	m	see A.13.9.5
7	VisibleString	-	Universal class 26	-	m	
8	GeneralString	-	Universal class 27	-	m	see A.13.9.6

A.13.9.2 String length parameter

		D	T2.3, A1.3
1	Maximum string length parameter supported	-	m

A.13.9.3 String significance parameter

		D	T2.3, A1.3
1	String significance parameter supported	-	m
2	Variable length strings supported	-	m
3	Fixed length strings supported	-	m

A.13.9.4 Character set parameter

		D	T2.3, A1.3
1	Character set parameter supported	-	m

A.13.9.5 G sets supported

G sets which are supported in NBS-12 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3)**

A.13.9.6 G and C sets supported

G and C sets which are supported in NBS-12 GeneralString.

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character set and ISO 646 IRV (C0) control character sets is required.** **(see 9.10.1.1-3)**

- END OF FTAM PHASE 3 PROFILES REQUIREMENTS LIST -

APPENDIX B : NIST OIW REGISTER OF FTAM OBJECTS

B.1 Introduction

This Index, the NIST OIW Register of OIW FTAM objects, contains a complete list of all FTAM objects as defined by NIST OIW.

NIST OIW was authorized by BSI in its letter dated August 09, 1989, as Registration Authority for NIST OIW defined objects. The Object Identifier Prefix for OIW is

{ iso(1) identified-organization(3) oiw(14) }

NIST OIW Plenary has delegated the authority and the task for maintenance of the NIST OIW Register of FTAM objects to its FTAM Special Interest Group (SIG) at its meeting on September 15, 1989. The Object Identifier Prefix for the FTAM SIG is

{ iso(1) identified-organization(3) oiw(14) ftamsig(5) }

~~For the FTAM Phase 2 defined objects, the FTAM SIG has decided to keep the ad hoc Object Identifier values as designated in 1987 and 1988 with the not necessarily unique Object Identifier Prefix~~

~~{ iso(1) identified-organization(3) icd(9999) organization-code(1) }~~

~~The reason is that FTAM Phase 2 products were already completed and released to users, so that changing these values would result in serious interworking problems.~~

For each new OIW FTAM object to be registered, a complete technical definition, that describes the purpose, scope and the unique characteristics of the object, must be prepared and presented to OIW FTAM SIG for technical discussion and acceptance, and for final approval by OIW Plenary.

B.2 INDEX of OIW FTAM Objects

B.2.1 FTAM Phase 2 Defined Objects

Object Identifier Prefix : **nist-adhoc** := iso(1) identified-organization(3) icd(9999) organization-code(1)

Object	Object Descriptor	Object Identifier	Date of Registration	Reference to Definition
NBS-6	NBS-6 FTAM sequential file	{nist-adhoc document-type(5) sequential(6) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.1
NBS-7	NBS-7 FTAM random access file	{nist-adhoc document-type(5) random-file(7) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.2
NBS-8	NBS-8 FTAM indexed file	{nist-adhoc document-type(5) indexed-file(8) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.3
NBS-9	NBS-9 FTAM file directory file	{nist-adhoc document-type(5) file-directory(9) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.4
	NBS ordered flat constraint set	{nist-adhoc constraint-set(4) nbs-ordered-flat(1) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix B.1
NBS-AS1	NBS abstract syntax AS1	{nist-adhoc abstract-syntax(2) nbs-as1(1) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix C.1
NBS-AS2	NBS file directory entry abstract syntax	{nist-adhoc abstract-syntax(2) nbs-as2(2) }	Dec 15, '89 Withdrawn March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix C.2
AP-Title		{nist-adhoc ftam-nil-ap-title(7) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 section 5.12.1.1.1

Object Identifier Prefix : **nist-oiw-ftam** := iso(1) identified-organization(3) oiw(14) ftamsig(5)

Object	Object Descriptor	Object Identifier	Date of Registration	Reference to Definition
NBS-6	NBS-6 FTAM sequential file	{nist-oiw-ftam document-type(5) sequential(6) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.5
NBS-7	NBS-7 FTAM random access file	{nist-oiw-ftam document-type(5) random-file(7) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.6
NBS-8	NBS-8 FTAM indexed file	{nist-oiw-ftam document-type(5) indexed-file(8) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.7
NBS-9	NBS-9 FTAM file directory file	{nist-oiw-ftam document-type(5) file-directory(9) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix A.8
	NBS ordered flat constraint set	{nist-oiw-ftam constraint-set(4) nbs-ordered-flat(1) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix B.2
NBS-AS1	NBS abstract syntax AS1	{nist-oiw-ftam abstract-syntax(2) nbs-as1(1) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix C.3
NBS-AS2	NBS file directory entry abstract syntax	{nist-oiw-ftam abstract-syntax(2) nbs-as2(2) }	March 16, '90	Stable Agreements Vers. 3, Ed. 2, March '90 NIST SP 500-177 chapter 9, appendix C.4

B.2.2 FTAM Phase 3 Defined Objects

Object Identifier Prefix : **nist-oiw-ftam** := iso(1) identified-organization(3) oiw(14) ftamsig(5)

Object	Object Descriptor	Object Identifier	Date of Registration	Reference to Definition
NBS-10	NBS-10 random binary access file	{nist-oiw-ftam document-type(5) random-binary(10) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix C.1
NBS-11	NBS-11 FTAM indexed file with unique keys	{nist-oiw-ftam document-type(5) indexed-file-with-unique-keys(11) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix C.2
NBS-12	NBS-12 FTAM simple text file	{nist-oiw-ftam document-type(5) simple-text-file(12) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix C.3
	NBS Random Access	{nist-oiw-ftam constraint-set(4) nbs-random-access(2) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix D.1
NBS-AS3	NBS random access node name abstract syntax	{nist-oiw-ftam abstract-syntax(2) nbs-node-name(3) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix E.1
NBS-AS4	NBS random binary access file abstract syntax	{nist-oiw-ftam abstract-syntax(2) nbs-random-binary(4) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix E.2
NBS-AS5	NBS simple text abstract syntax	{nist-oiw-ftam abstract-syntax(2) nbs-simple-text(5) }	Dec 15, '89	Stable Agreements Vers. 3, Ed. 1, Dec '89 NIST SP 500-177 chapter 10, appendix E.3

~~10.12~~ APPENDIX C: DOCUMENT TYPES

C.1 NBS-10 Random Binary Access Document Type

1. Entry Number: NBS-10

2. Information objects

Table 10.2. Information objects in NBS-10

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) random-binary(10)} "NBS-10 random binary access file"
abstract syntax names: a) name of asname1 b) name of asname2 c) name of asname3	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-random-binary(4)} "NBS random binary access file abstract syntax" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU" {iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-node-name(3)} "NBS random access node name abstract syntax"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1)} "Basic encoding of a single ASN.1 type"
file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-random-access(2)} "NBS random access constraint set"

File contents:

```
Datatype1 ::= a single octet

Datatype2 ::= Node-Name
 --The type to be used for Node-Name is defined in
 --ISO 8571-FADU
 --The only Choice for Node-Name is user-coded
Datatype3 ::= NBS-Node-Name
 --As defined by the NBS Node Name Abstract Syntax
```

3. Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units each of which consists of one data element. The data element is made up of one octet. The order of these elements is significant. The semantics of the data elements is not specified by this document type.

The document structure takes the form allowed by the FTAM hierarchical file model as constrained by the NBS random access constraint set. The definition for FTAM hierarchical file model appears in 8571-2.

There are no size or length limitations imposed by this definition.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a series of octets.

9. Definition of transfer

9.1. Datatype definition

The presentation data value used for transfer is an ASN.1 OCTET STRING.

Datatype2 is used to specify the FADU-Identity of "name-list" in the FTAM PDUs specifying FADU-Identity, where "name-list" is defined as a SEQUENCE of EXTERNAL. The EXTERNAL is defined as Node-Name in the FTAM FADU abstract syntax. The use of Datatype2 is defined in "NBS random access constraint set."

Datatype3 specifies the "user-coded" form of the Node-Name in the FTAM FADU abstract syntax, where "user-coded" is defined as an

EXTERNAL. That EXTERNAL is defined by Datatype3. The use of Datatype3 is defined in "NBS random access constraint set."

9.2 Presentation data values

The document is transmitted as a series of presentation data values. Each presentation data value shall consist of the "data" from one or more FADUs concatenated together. The result is one value of the ASN.1 data type OCTET STRING. The "fadu_count" field supplied in the Node-Name specifies the number of FADUs to transfer during a Read operation. The requested FADUs may be transferred as one or more presentation data values.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 10.2.

Note: Specific carrier standards may impose additional constraints on the presentation context to be used, when the above permits a choice.

Boundaries between P-DATA primitives and between presentation data values are chosen locally by the sending entity at the time of transmission. The boundaries are not preserved when the file is stored and they carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of Data Units within the file.

10. Transfer syntax

An implementation supporting these document types shall support the transfer syntax generation rules named in table 10.2 for all presentation data values transferred.

Implementations may optionally support other transfer syntaxes.

11. ASE specific specifications

11.1 Simplification and relaxation

The document type NBS-10 may be simplified to the document type FTAM-3. The resultant document contains the same sequence of data values as would result from accessing the file as an NBS-10 file.

11.2 The READ operation

A READ operation may be applied to a range of FADUs via the FADU Identity of "NodeSeq." The "starting-fadu" part of the node name specifies the node number of the first FADU; the "fadu-count" specifies the number of consecutive FADUs to be transferred.

A READ operation applied to a range of FADUs that spans beyond the end of file is valid. All available data in the range is transferred. An informative diagnostic (5005) is returned on the F-Data-End Request indicating that the end of file was reached and a portion of the request was satisfied.

11.3 The REPLACE operation

When the REPLACE operation is applied to the root FADU of an NBS-10 document, the transferred data shall be any NBS-10 document.

The REPLACE operation applied to a FADU identity of "node number" is used to replace a series of FADUs, starting at the specified position in the file, by the new FADUs being transferred. The number of replaced FADUs is determined by the number of transferred FADUs.

If the replacement spans beyond the end of the existing file, then the additional FADUs are inserted at the end of the file.

11.4 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred data shall be a series of FADUs which would be generated by reading any NBS-10 document type in access context UA.

C.2 NBS-11 Indexed Sequential File With Unique Keys

1. **Entry Number: NBS-11**
2. **Information objects**

Table 10.3. Information objects in NBS-11

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file-with-unique-keys(11)} "NBS-11 FTAM indexed file with unique keys"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization oiw(14) ftamsig(5) ~~organization-code(1)~~ abstract syntax(2) nbs-as1(1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE            {DataTypes, KeyType, KeyPosition}

  DataTypes  ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}

  KeyType    ::= CHOICE {Parameter0, Parameter1, Parameter2}

             --  Parameter0, Parameter1, Parameter2, as defined for the
             --  document types NBS-6, NBS-7, NBS-8

  KeyPosition::= INTEGER
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set(4) ordered-flat-unique-names(4)} "FTAM ordered flat constraint set with unique names"

```
file contents:
               Datatype1 ::= PrimType -- as defined in Appendix 9C, C.3
                                      ~~Annex 9 A, Part 3~~
                                      of NIST SP 500-~~162~~ 177
               Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                        Enter-Subtree-Data-Element }
                                        Exit-Subtree-Data-Element }
```

3. **Scope and field of application**

The document type defines the contents of a file for storage, for transfer and access using FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. **References**

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see table 10.3). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set with unique names:

- The FADU identity 'node number' is not required for conformant implementations
- The identities 'next' and 'previous' are allowed for all FADUs

Each data element is a data type from the set of primitive data types defined in ~~Appendix 9A, Part 3 of NIST 500-162~~ Appendix 9C, C.3 of NIST SP 500-177. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the <DataTypes> parameter.

NBS-11 Indexed Sequential File With Unique Keys

1. **Entry Number: NBS-11**
2. **Information objects**

Table 10.3. Information objects in NBS-11

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file-with-unique-keys(11)} "NBS-11 FTAM indexed file with unique keys"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization icd(9999) organization-code(1) abstract-syntax(2) nbs-as1(1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding(1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE            {DataTypes, KeyType, KeyPosition}

  DataTypes  ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}

  KeyType    ::= CHOICE {Parameter0, Parameter1, Parameter2}

             --  Parameter0, Parameter1, Parameter2, as defined for the
             --  document types NBS-6, NBS-7, NBS-8

  KeyPosition::= INTEGER
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set(4) ordered-flat-unique-names(4)} "FTAM ordered flat constraint set with unique names"

```
file contents:
               Datatype1 ::= PrimType -- as defined in Annex 9 A, Part 3
                                      of NIST SP 500-162
               Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                        Enter-Subtree-Data-Element }
                                        Exit-Subtree-Data-Element }
```

3. **Scope and field of application**

The document type defines the contents of a file for storage, for transfer and access using FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. **References**

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see table 10.3). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set with unique names:

- The FADU identity 'node number' is not required for conformant implementations
- The identities 'next' and 'previous' are allowed for all FADUs

Each data element is a data type from the set of primitive data types defined in Appendix 9A, Part 3 of NIST 500-162. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the <DataTypes> parameter.

The string-length field of Parameter 1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

Each data unit in the file has a key associated with it, which is the user-coded form of Node Name. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in Appendix 9A, Part 3 of NIST 500-162. Appendix 9 C, C.3 of NIST SP 500-177.

The type and length of the key are defined by the <KeyType> parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in the following table 10.4.

Table 10.4. Datatypes for keys

Key Type	Minimum Range (octets)	Order
ASN.1 INTEGER	(1-2)	increasing numeric value
ASN.1 IA5String	(1-16)	lexical order
ASN.1 GraphicString	(1-16)	lexical order
ASN.1 GeneralString	(1-16)	lexical order
ASN.1 OCTET STRING	(1-16)	increasing value
ASN.1 GeneralizedTime		increasing time value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing numeric value

The position of the key in the data unit is specified by the <KeyPosition> parameter.
KeyPosition = 0 implies the key is not part of the data
KeyPosition > 0 specifies the actual data element in the data unit.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 10.3, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 10.3, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1," carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2."

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g., document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 10.2 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-11 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

A document of type NBS-11 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>. The traversal order of the FADUs must be maintained.

Note: The traversal order is as reading the file as NBS-11 in key order.

A document of type NBS-11 may be accessed as a document of type NBS-8 (allowed only when reading the file) by specifying document type NBS-8 in the <contents type> parameter in the <F-OPEN REQUEST>.

11.2 Access context selection

A document of type NBS-11 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set

with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied the transferred material shall be the series of FADU which would be generated by reading any NBS-11 document with the same parameter values in access context FA.

A transferred FADU whose name duplicates that of an already existing FADU will cause the <INSERT> operation to fail. The failure shall be signalled by issuing an F-CANCEL Request with a corresponding diagnostic.

11.4 The EXTEND operation

This operation is excluded for the use with this document type.

11.5 The REPLACE operation

When the <REPLACE> operation is applied with FADU Identity 'begin', a transferred FADU whose name duplicates that of a previously transferred FADU will cause the <REPLACE> operation to fail. The failure shall be signalled by issuing an F-CANCEL Request with a corresponding diagnostic.

C.3 NBS-12 Simple Text File Document Type

1. **Entry Number: NBS-12**
2. **Information objects**

Table 10.5. Information objects in NBS-12

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) simple-text-file(12)} "NBS-12 FTAM simple text file"
abstract syntax names: a) name for asname1 b) name for asname2	 {iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-simple-text(5)} "NBS simple text abstract syntax" {iso standard 8571 abstract-syntax(2) ftam-fadu(2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1)} "Basic Encoding of a single ASN.1 type"

Parameter Syntax

```
PARAMETERS ::= SEQUENCE{

    universal-class-number [0] IMPLICIT INTEGER,
    maximum-string-length  [1] IMPLICIT INTEGER,
    string-significance    [2] IMPLICIT INTEGER {variable (0), fixed (1)},
    character-set          [3] IMPLICIT OctetString OPTIONAL}
```

file model	{iso standard 8571 file-model(3) hierarchical(1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set(4) sequential flat(2)} "FTAM sequential flat constraint set"

File contents

```
              Datatype1 ::= NBS Text
                                  --as defined in the NBS Simple Text
                                  --Abstract Syntax registration entry


              Datatype2 ::= Node-Descriptor-Data-Element
```

3. **Scope and field of application**

The document type defines the contents of a file for storage, and for transfer and access by FTAM.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

ISO 8824, Information Processing Systems-Open Systems Interconnection-Specification of Abstract Syntax Notation 1 (ASN.1).

ISO 8825, Information Processing Systems-Open Systems Interconnection-Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

ISO 6429, Information Processing-ISO 7-bit and 8-bit coded character sets-Additional control functions for character imaging devices.

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1. In addition, it makes use of the terms character string, graphics character, and format effector as defined in document type registration entry "FTAM-2" ir ISO 8571-2.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

This document consists of zero, one or more file access data units, each of which consists of one character string. The order of each of these elements is significant. The semantics of the character strings is not specified by this document type.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the sequential flat constraint set. These definitions appear in ISO 8571-2. As additional constraints FADU identity will be limited to the following values:

a) 'begin' and 'end' when using the Transfer or Transfer and Management service classes.

b) 'begin', 'end', 'first', and 'next' when using the Access service class.

Each character string consists of characters from the character set defined by the ASN.1 (ISO 8824) character set type whose universal class number is given by the "universal-class-number" parameter and by the escape sequences contained in the optional "character-set" parameter. If the character set type allows explicit escape sequences, the "character-set" parameter, if present, contains escape sequences which designate and invoke specific

character sets. If the "character-set" parameter is not present, character sets are assumed to be designated and invoked as specified in table 2 in ISO 8825. Character strings shall not contain escape sequences.

There are no size or length limitations imposed by this definition, except those specified here. Each character string is of a length determined by the number of characters given by the "maximum-string-length" parameter.

Note: The length restriction refers to the number of characters from the applicable character set, not to the number of octets in the encoding, nor to the line length in any rendition of the document, where these are different.

The exact significance of the character strings is determined by the "string-significance" parameter. If its value is "variable," the length of the character strings is less than or equal to the length given. If the value is "fixed," the length of each character string is exactly equal to the length given.

If the document is interpreted on a character imaging device (outside the scope of ISO 8571), the interpretation depends on the character set in use.

a) If the character set contains format effectors, they shall be interpreted as defined in ISO 6429; end of string and end of file access data unit are given no formatting significance, and do not contribute to the document semantics;

b) If the character set does not contain format effectors, the end of each character string is interpreted as implying carriage return and line feed formatting actions in any rendition. The end of file access data unit is given no formatting significance beyond that attached to the end of the string in it.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 modules ISO8571-FADU and ISO 8571-CONTENTS in ISO 8571, in which each of the file contents data elements has the abstract syntactic structure of "NBS Simple Text."

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 10.5, the ASN.1 datatype declared as "NBS-Text" in the NBS Simple Text Abstract Syntax definition. The choice in "NBS-Text" is determined by the universal-class-number parameter; or

b) Datatype2 defined in table 10.5, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO 8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1," carrying one of the character strings of the document. Each character shall be transmitted using one of the character sets identified by the universal-class-number parameter. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in table 10.5, or

b) one value of the ASN.1 datatype "Datatype2." All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2" declared in table 10.5.

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between P-DATA primitives are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

9.3 Sequence of presentation data values

The sequence of presentation data values of type (a) and the sequence of presentation data values of types (a) and (b) is the same as the sequence of character strings within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 10.5 for all presentation data values transferred.

11. ASE specific specifications

11.1 Simplification and relaxation

11.1.1 Simplification to FTAM-1

This simplification loses information.

The document type NBS-12 may be accessed as a document type FTAM-1. The resultant document contains the same sequence of data values as would result from accessing the structured text file in access context UA. That is, only the presentation data values in the abstract syntax "asname1" are present. If the "character-set" parameter was present before the simplification, its contents will be added to the beginning of each string.

Note: The boundary between file access data units remains a boundary between strings, but any special significance given to it is lost.

11.1.2 Relaxation to FTAM-2

The document type NBS-12 may be relaxed to the document type FTAM-2. If the "character-set" parameter was present before the relaxation, its contents will be added to the beginning of each string.

11.1.3 Character set relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to a different document of type NBS-12 with

- a different "universal-class-number" parameter value,
- a different "character-set" parameter value,
- different values for both of these parameters,
- a different "universal-class-number" parameter value and no "character-set" parameter value, or
- no "character-set" parameter value,

if the resultant document type permits all characters from the original document type. If this relaxation involves including format effectors and none were present before the simplification, the characters "carriage return" and "line-feed" shall be added to the end of each string.

Note: If the characters "carriage return" and "line feed" are not part of the format effectors, the formatting action may be represented by "newline," or some other implementation specific choice if there is no representation of "newline" defined.

11.1.4 String length relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to another document type NBS-12 with a larger "maximum-string-length" parameter.

11.2 Access context selection

A document of type NBS-12 may be accessed in any one of the access contexts defined in the sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any NBS-12 document type with the same parameter values in access context FA.

10-13 APPENDIX D: CONSTRAINT SETS

D.1 NBS Random Access Constraint Set

Table 10.6. Basic constraints in the NBS Random Access Constraint Set

Constraint set descriptor	NBS Random Access
Constraint set identifier	{iso identified-organization oiw(14) ftamsig(5) constraint-set(4) nbs-random-access(2)}
Node names	All names shall be of the same type; the type of the names and an ordering of the names shall be defined when reference is made to the constraint set.
File access actions	Locate, Read, Insert, Erase, Replace
Qualified actions	None
Available access context	UA
Creation state	Root node without an associate data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected
Read whole file	Read in access context UA with FADU-Identity of"begin"
Write whole file	Transfer a series of leaf FADUs which would be generated by reading the whole file in access context UA; Perform the transfer with an FADU Identity of "end" and a file access action of "insert," or with an FADU Identity of "begin" and an action of "replace," or with an FADU Identity of "node-number" and an action of "replace." Here "node number" identifies the first FADU in the preorder traversal sequence.

Table 10.7. Identity constraints in the NBS Random Access Constraint Set

Action	Begin	End	NodeSeq	Node Number
Locate				leaf
Read	whole		leaf	
Insert		leaf		
Erase	whole			leaf
Replace	whole			leaf

Note: NodeSeq = A sequence of node names with a single member

1. **Field of application**

The NBS Random Access constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made randomly by NodeSeq. The structuring of the file into individual FADUs is determined by the NodeName.

2. **Basic constraints**

The basic constraints in the NBS Random Access constraint set are given in table 10.6.

3. **Structural constraints**

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and shall have an associated data unit; all arcs from the root node shall be of length one.

4. **Action constraints**

Insert: the insert action is allowed only at the end of the file, with FADU-Identity of "end"; the new node is inserted following all existing nodes in the file. The location following the insert is "end."

Erase: the erase action is allowed at the root node to empty the file, with FADU-Identity of "begin." The result is a solitary root node without an associated data unit. Erase with the FADU-Identity of "node number" means truncation of the file.

Replace whole file: the FADU-Identity is "begin" and the complete series of new FADU contents is sent.

Replace new leaves: the FADU-Identity is "node number" and the number of FADUs being replaced is given by the number of FADUs sent.

5. **Identity constraints**

The FADU-Identity associated with the file action shall be one of the identities: begin, end, Node Number and NodeSeq. The actions with which

these identities can be used are given in table 10.7.

| ~~10-14~~ APPENDIX E: ABSTRACT SYNTAXES

E.1 NBS Node Name Abstract Syntax

Abstract Syntax Name
{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-node-name(3) }

"NBS random access node name abstract syntax"

This is an abstract syntax for the user-coded Node-Name in the FTAM FADU abstract syntax.

```
NBS-AS3 DEFINITIONS::=

BEGIN

     NBS-Node-Name::= SEQUENCE

          {    starting-fadu [0] IMPLICIT INTEGER,
               fadu-count [1] IMPLICIT INTEGER }
                    --a "fadu-count" of 0 specifies the
                    --range of FADUs
                    --beginning at "starting-fadu" and
                    --ending at "end of file"

END
```

For this abstract syntax the following transfer syntax will be used.

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

E.2 NBS Random Binary Access File Abstract Syntax

Abstract Syntax Name
{ iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-random-binary(4) }

"NBS random binary access file abstract syntax"

This is an abstract syntax for the transfer of the file contents for NBS Random binary files.

```
NBS-AS4 DEFINITIONS::=

BEGIN
     NBS-Random Binary ::= OCTET STRING
         --contains one or more presentation data values
         --concatenated together.
         --Each presentation data value is defined as
```

```
                    --Datatype1 in table 10.2.
          END
```

For this abstract syntax the following transfer syntax will be used.

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

E.3 NBS Simple Text Abstract Syntax

```
Abstract Syntax Name
          {iso identified-organization oiw(14) ftamsig(5)
          abstract-syntax(2) nbs-simple-text(5) }
          "NBS simple text abstract syntax"

          NBS-AS5 DEFINITIONS::=

          BEGIN

          NBS-Text::= CHOICE {

                              IA5String,--Universal Class 22
                              GraphicString,--Universal Class 25
                              VisibleString,--Universal Class 26
                              GeneralString--Universal Class 27
                                   }
          END
```

For this abstract syntax, the following transfer syntax will be used:

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

11 DIRECTORY SERVICES PROTOCOLS

11.1 INTRODUCTION

This is an Implementation Agreement developed by the Implementor's Workshop sponsored by the National Institute of Standards and Technology to promote the useful exchange of data between devices manufactured by different vendors. This agreement is based on and employs protocols developed in accord with the OSI Reference Model. While this agreement introduces no new protocols, it eliminates ambiguities in interpretations.

This is an Implementation Agreement for the OSI Directory based on the ISO and CCITT documents cited in the Reference section 19.2 (hereafter referenced as Directory Documents). This agreement is aligned with the UNOFFICIAL 'FINAL' version of the X.500 Series of Recommendations, December 1988. Where technical differences between the ISO and CCITT versions of these documents exist (e.g., Transport Requirements) the ISO versions are given precedence. figure 11.1 displays the structure of this Implementation Agreement. References to corresponding CCITT documents are included for information.

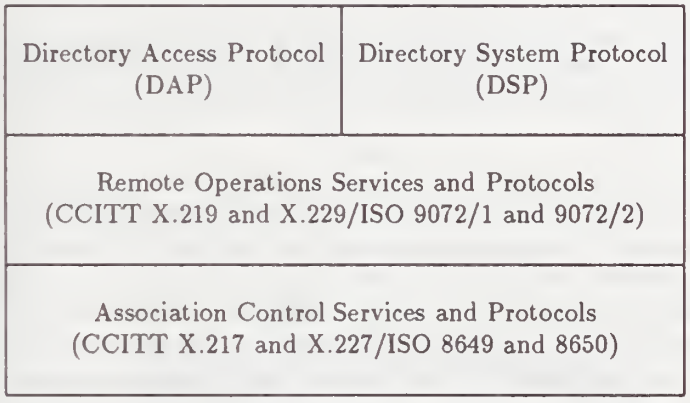


Figure 11.1. Structure of this Implementation Agreement.

The Directory User Agents (DUAs) and Directory System Agents (DSAs) provide access to The Directory on behalf of humans and applications such as Message Handling and File Transfer, Access, and Management. See the Scope and Field of Application section for more information on the model used in the Directory.

This document covers both the Directory Access Protocol (DAP) and the Directory System Protocol(DSP) defined in the Directory Documents. A good working knowledge of the Directory Documents is assumed by this chapter. All terminology and abbreviations used but not defined in this text may be found in those documents.

11.2 SCOPE AND FIELD OF APPLICATION

Centralized and distributed directories can both be accommodated in this Agreement by the appropriate choice of protocols and pragmatic constraints from those specified. Figure 11.2 illustrates a centralized directory and figure 11.3 illustrates a distributed directory.

This agreement does not cover interaction between co-located entities, such as a co-resident DUA and DSA. It also does not specify the interface between a user (person or application) and a DUA. Bilateral agreements between a DUA and DSA or DSA and DSA may be implemented in addition to the requirements stated in this document. Conformance to this agreement requires the ability to interact without the use of bilateral agreements other than those required in the Directory Documents.

The logical structure of the Directory Information Base (DIB) is described in the Directory Documents. The manner in which a local portion of the DIB is organized and accessed by its DSA is not in the scope of this agreement.

11.3 STATUS

This version was completed in December 1989.

11.4 USE OF THE DIRECTORY

Given the rapid multiplication and expansion of OSI applications, telecommunication systems and services, there is growing need for users of OSI applications, as well as the applications themselves, to communicate with each other. In order to facilitate their communications, a Directory protocol, as referenced in these agreements, has been tailored to meet their respective needs.

In one instance, The Directory will be used as a service to provide humans, in an on-line fashion, rapid and easy retrieval of information useful for determining what telecommunications services are available, and/or how to access, and address their correspondents. Further, service providers offering such a Public Directory may also use this service internally with other various telecommunications services (e.g., MHS) for the proper addressing of calls or messages. Likewise, this does not preclude the usage of these agreements to similarly generate a privately operated Directory that supports both human and application information exchanges.

In another instance, The Directory, will be used as a service by computer applications without direct human involvement. One important service is to provide Presentation Address resolution for named objects, on behalf of OSI applications. The Directory may be used by applications to search for objects (i.e., Application Entities), without direct human involvement, by the use of the "search" or "list" operations.

To support the many possible usages, The Directory is a general purpose system. It is capable of storing data of many different forms as attributes within entries, and is also capable of supporting simple or complex hierarchical structures, with variations in structure possibly occurring between one part of The Directory and another.

Compliant DSA implementations should safeguard this generality, where possible, by placing the minimum of restrictions in "hard-wired" form.

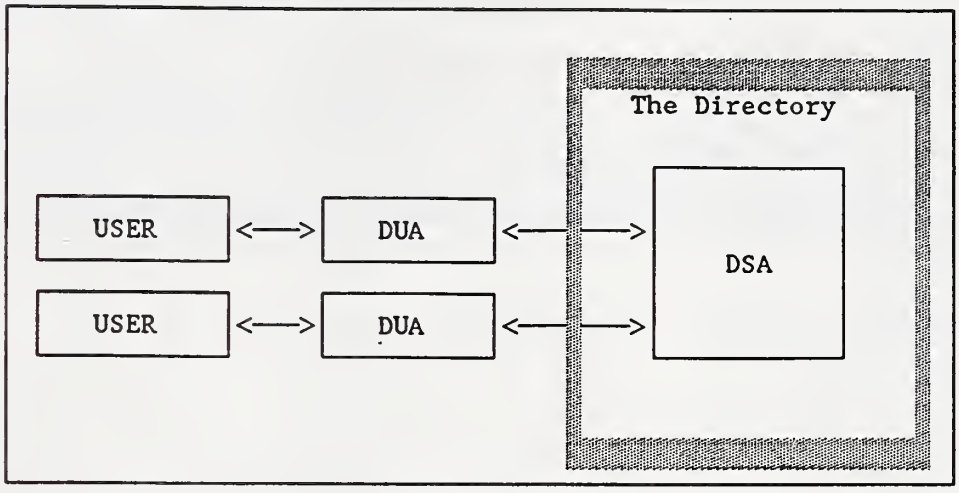


Figure 11.2. Centralized Directory Model.

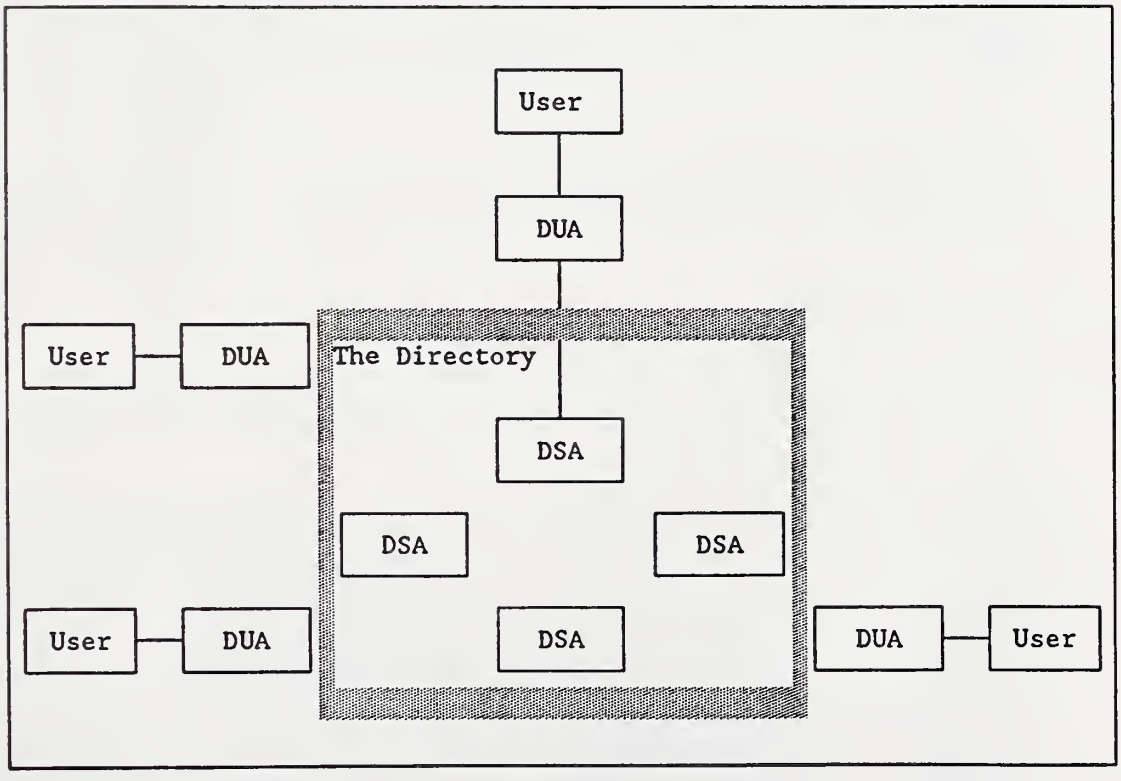


Figure 11.3. Distributed Directory Model.

11.5 DIRECTORY ASEs AND APPLICATION CONTEXTS

This section highlights the ASEs (Application Service Elements) and Application Contexts defined in the Directory Documents and of concern in these Agreements. The functionality of the Directory AEs (DUAs and DSAs) is defined by a set of ASEs, each Directory ASE specifying a set of Directory operations.

The interaction between these AEs is described in terms of their use of ASEs. This specific combination of a set of ASEs and the rules for their usage defines an application context.

The following ASEs are described in the Directory Documents:

- Read ASE
- Chained Read ASE
- Search ASE
- Chained Search ASE
- Modify ASE
- Chained Modify ASE

ROSE and ACSE also form part of the Directory Application Contexts.

The following Application Contexts are described in the Directory Document:

- Directory Access Application Context
- Directory System Application Context

11.6 SCHEMA

There are seven (7) major topics that relate to schema:

11.6.1 Support of Structures and Naming Rules

DSAs shall be capable of supporting (subject to refinements laid down in these Agreements) the structure and naming rules defined in the Directory Documents, Part 7, Annex B.

Part 7, Annex B of the Directory Documents provides a framework for the basic use of the Directory in terms of the objects defined in Part 7. It does not, however, form part of the standard and, in any case, permits structures and practices which may be undesirable. The guidelines below provide tighter control within the Annex B framework.

It is recommended that only an entry subordinate to Root or Country may use a State-OrProvinceName AVA as an RDN.

11.6.2 Support of Object Classes and Subclasses

The DSAs shall be able to support all superclasses of the supported object classes (e.g., Top, Person).

Use of an object class in this profile or the standard (or a subclass derived from one or more of these object classes) is recommended wherever the semantics are appropriate for the application. The derivation of a new object class as an immediate subclass of Top should be avoided. For example, to represent printers in the Directory, one can derive a subclass of Device.

An entry of a particular object class may contain any optional attribute listed for it in the Directory Documents; a conformant DSA shall be able to support all these optional attributes.

In addition, a DSA may permit any locally registered attribute, or a subset of these, by providing the local extension facilities permitted by unregistered object classes (viz. Directory Documents, Part 2, clause 9.4.1 (a) and Note).

11.6.3 Support of Attribute Types

DSAs shall be able to support the storage and use of attribute type information, as defined in the Directory Documents, Part 6, including their use in naming and access to entries; they shall also support the definition of new attribute types, making use of pre-existing attribute syntaxes.

DSAs shall support the encoding, decoding, and matching of all the attributes in the Naming Prefixes of every naming context they hold (ref Directory Documents, Part 4, clause 9). These attributes may include attributes that are not permitted to appear in entries in those naming contexts.

11.6.4 Support of Attribute Syntaxes

Suggested methods for the interpretation of selected Attribute Syntaxes are defined in Appendix A.

11.6.5 Naming Contexts

The root of a naming context shall not be an alias entry.

11.6.6 Common Profiles

This section identifies profiles that are commonly useful for various applications while an application–specific profile(s) is identified by the application.

11.6.6.1 OIW Directory Common Application Directory Profile

11.6.6.1.1 Standard Application Specific Attributes and Attribute Sets

The attributes and attribute sets in the Directory Document, Part 6, associated with the object classes listed below are required.

11.6.6.1.2 Standard Application Specific Object Classes

DSAs shall be able to support storage and use of the object classes below, as defined in the Directory Documents, Part 7, and these object classes are expected to be useful for a range of applications.

The following object classes are mandated by the standard:

Top	Alias
DSA	

The following object classes are expected to be generally useful in the creation of the upper portion of the DIT:

Country	Organization
Locality	Organizational Unit
Application Process	

The following object classes are expected to be generally useful in the creation of DIT leaf entries:

Alias	Group of Names
Application Process	Organizational Person
Application Entity	Organizational Role
DSA	Residential Person
Device	

11.6.6.2 OIW Directory Strong Authentication Directory Profile

11.6.6.2.1 Other Profiles Supported

This profile is used in conjunction with the OIW Directory Common Application Directory Profile.

11.6.6.2.2 Standard Application Specific Object Classes

The following object classes are expected to be generally useful for applications to support strong authentication:

- Strong Authentication User
- Certification Authority

11.6.7 Restrictions on Object Class Definitions

An object class may not be defined as a subclass of itself, as the chain of superclasses of such an object class would be a closed loop, isolated from all other object classes, specifically Top. Such isolation is clearly illegal.

11.7 PRAGMATIC CONSTRAINTS

This section describes pragmatic constraints to which a conformant implementation shall adhere in addition to those specified in the Directory Documents. The pragmatic constraints can be divided into two major areas. The first includes those aspects of pragmatic constraints which apply to scope of service (see sec. 11.7.1 and sec. 11.7.2). The second includes those aspects of pragmatic constraints which are specific to particular attribute types (see sec. 11.7.3).

11.7.1 General Constraints

11.7.1.1 Character Sets

It is a requirement to support all character sets and other name forms defined in the Directory Documents, Part 6. Those character sets include:

- T.61
- PrintableString
- NumericString

11.7.1.2 APDU Size Considerations

In the process of chaining requests it is possible that a chaining DSA may receive, invoke or return APDUs that exceed its capacity. It is a minimum requirement that invoke APDUs and return result APDUs shall be accepted unless they exceed 32767 octets in size; in this case they may be discarded as illustrated in the right side of figure 11.4 (page 11 – 8), and an "unwillingToPerform" error reporting service shall be used.

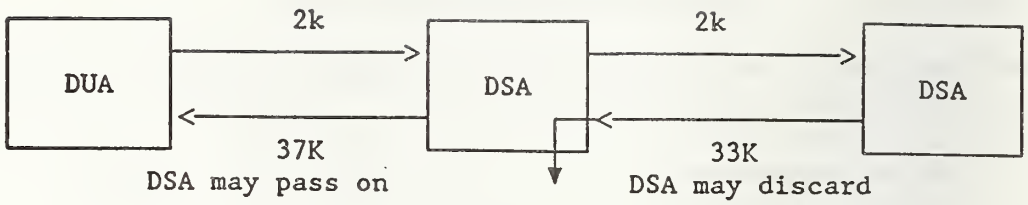


Figure 11.4. APDU Exchange.

11.7.1.3 Service Control (SC) Considerations

This agreement recognizes that DUAs may automatically supply defaults for any SC parameter. The choice of default values selected (if any) is seen to be a matter of local policy and consumer needs.

11.7.1.4 Priority Service Control

Priority is specified as a service control argument in the Directory Documents. The following statements represent a clarification of the semantics that may be used by a DSA in interpreting and operating on this parameter.

The logical model in figure 11.5 (page 11 – 9) may be considered as an example by DSAs that implement this Service Control. In figure 11.5, note that:

- the DSA maintains three logical queues corresponding to the three priority levels;
- the DSA Scheduler is separate and distinct from any scheduling function provided by the underlying operating system or control program services;
- the DSA Scheduler presents jobs to the Underlying Operating Services for execution and always presents jobs of a higher priority before those of a lower priority;
- the DSA Scheduler will not preempt a request once it has been passed to the underlying operating system service.

11.7.2 Constraints on Operations

There are no overall constraints upon service arguments or results except those implied in section 11.7.1.2 of this document.

11.7.2.1 Filters

It is required that DSAs, at a minimum, support 8 nested "Filter" parameters, and a total limit of 32 Filter Items. If these limits are exceeded, the recipient of that SearchArgument may return the ServiceProblem "unwillingToPerform".

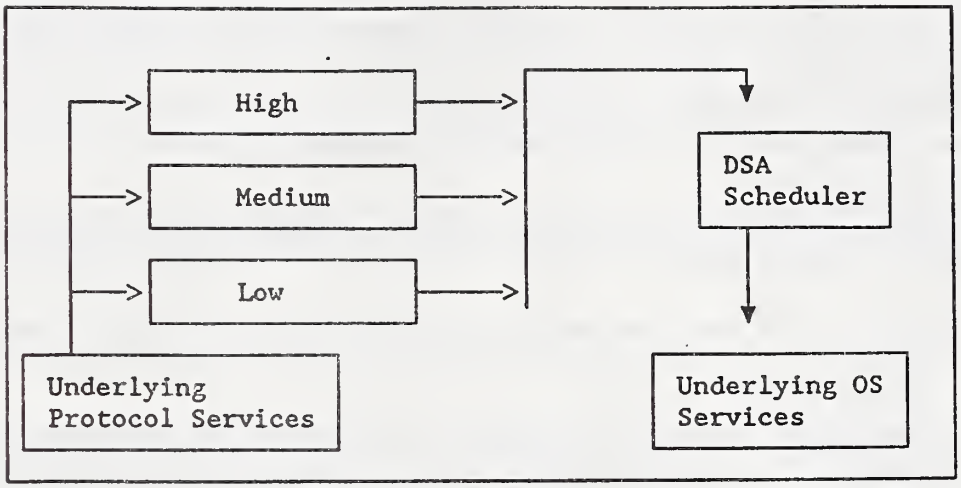


Figure 11.5. Logical DSA Application Environment.

11.7.2.2 Errors

There are no constraints upon any Error service except the APDU size limit as defined in Section 11.7.1.

11.7.2.3 Error Reporting – Detection of Search Loop

A search operation may encounter a looping situation when the search encompasses "whole-subtree", and an alias is encountered which is a superior to some other subtree that has been encountered during the search.

DSAs should be able to detect this situation. One possible method is by:

1. Maintaining a list of the base objects of searches initiated as a consequence of Step 5 of Part 4, clause 18.7.2.2.1 of the Directory Documents (this may require an analysis of the TraceInformation field).

2. Determining whether a new base object is superior to any base object on this list.

A new base object which would cause a loop in this way should be discarded (i.e. should not cause a new search), but no error should be reported by an error–reporting service. The circumstances should be logged so that it may be reported to an appropriate Administrative Authority for rectification.

11.7.3 Constraints Relevant to Specific Attribute Types

Table 11.1 (pages 11 – 36 and 11 – 37) gives pragmatic constraints associated with selected attribute types specified in the Directory Documents; many of these constraints also appear and are the same in the CCITT version of the Directory Documents. Each constraint in Table 11.1 is given in terms of a length constraint. The length constraint for a given attribute

value is the number of units which a sending entity shall not exceed and which a receiving entity shall accept and process. A sending entity need not be capable of sending attribute values as large as the length constraints.

Note that in table 11.1 the length constraint for strings is expressed as the number of allowable characters.

In addition to the constraints given in table 11.1, the following constraints apply to alphabets and integer values.

- Alphabets: T.61 Strings used as attribute values shall only encode graphic characters and spaces. They shall not contain formatting characters (such as subscript) or other control characters.
- Integer Values: DSAs shall be required to "pass through" encoded integer attribute values of arbitrary length (e.g., when chaining a Directory operation). No Directory component (i.e., DUA or DSA) shall be deemed non–conformant if it encodes integer attribute values of arbitrary length.

  Components of the Directory are required to support (for storage and processing), as a minimum, integer attribute values encoded in 4 octets.

11.8 CONFORMANCE

The following sections will describe various aspects of Directory conformance. It should be noted that conformance to the various ASEs and conformance to the Authentication Framework are viewed as separate issues and are presented in that context.

11.8.1 DUA Conformance

Conformance requirements for DUAs are adequately specified in the Directory Documents, Part 5, clause 9.1 and the Directory Access Profile (see sec. 11.8.6). It should be noted that the DUA conformance is based on DAP Protocol and not the User Interface. Not all options available in the standard need to be made available to the user of the DUA.

It is recognized that DUAs will be widely differing in nature:

- Some are intended to support human users, some application users
- Particular DUAs may not support particular operations because the application that they support has no requirement; others will be general purpose, and will support all operations.
- Some DUAs will have a fixed view of the Directory content and structure, reflecting the usage of The Directory by a particular application; others will have a more flexible view which can be adapted to new usages.
- Some DUAs will provide automatic referral services with automatic establishment and release of associations; others will place the burden on the user.

- Some DUAs will provide a variety of authentication means; others will support no authentication
- Some DUAs will handle operations synchronously; others will have the capability of maintaining several identifiable dialogues with The Directory at one time.

In the next section, different types of DSAs are discussed. The DUA is independent of the type of DSA it is communicating with and does not need to know what type of DSA it is communicating with.

11.8.2 DSA Conformance

Basic conformance requirements for a DSA are defined in the Directory Documents, Part 5, clause 9.2. Some of the terms used to describe DSA conformance are summarized below.

- *Centralized*: A centralized DSA is defined as one that contains its entire relevant DIT; it follows that it will not make use of the DSP or generate referral responses. Since this model only contains a single DSA it is not subject to DSA interworking issues and will always provide a consistent level of service and results. A centralized DSA shall be fully "protocol" conformant to the DAP.
- *Cooperating*: In a distributed directory, responsibility for various portions of the DIT may be "distributed" among multiple DSAs. On a per operation basis we define a DSA to be *holding* when it is responsible for the fragment of the DIB in which a given entry will appear if it exists; we define a DSA to be *propagating* when it is unable to complete the name resolution process.

All DSAs shall be capable of acting as a holder and a propagator.

11.8.3 DSA Conformance Classes

A DSA implementation shall satisfy the conformance requirements as defined in the Directory Documents, Part 5, section 9.2, and shall support the "Versions" argument of "Bind".

Per the conformance section of the Directory Documents, a DSA shall conform to the abstract syntax of the attribute types for which conformance is claimed. These attribute types shall include those required by section 11.6.3 of this Implementor's Agreement.

Additionally, an implementation conformant to these agreements shall state which of the following conformance classes it implements:

- **Conformance Class 0 – Centralized DSA**

  A DSA conformant to this class only supports the DirectoryAccessAC.

  As the performance of Search and List operations can consume significant resources, the policies of some centralized DSAs may be such that these operations will not be performed. For these cases, the reply to requests for such operations would be a ServiceError with the "unwillingToPerform" ServiceProblem.

- **Conformance Class 1 – Distributed DSA**

  A DSA implementation conformant to this class shall implement all the operations in the ASEs that are part of the Application context for which it claims conformance. It shall support the DirectoryAccessAC and it may optionally support the DirectorySystemAC.

DSAs conformant to these Agreements shall support the OIW Directory Common Application Directory Profile. In addition, DSAs may optionally conform to the OIW Directory Strong Authentication Directory Profile. Future versions of these Agreements may allow additional possibilities for minimal profile conformance.

11.8.4 Authentication Conformance

A Directory System may choose to implement various levels of authentication (Directory Documents, Part 8). We define the following levels of authentication in the DS:

- No authentication at all; (**None**)
- **Simple Uncorroborated:** identification without verification
- **Simple Uncorroborated** authentication **with** verification: verified identification without a password.
- **Simple Corroborated** authentication: verified identification with a password; intended to make masquerading difficult.
- **Strong** authentication: identification with verification using cryptographic techniques intended to make masquerading, in practical terms, nearly impossible.

The "Authentication Framework" document describes the specific goal of each authentication level; listed below are several practical uses of the various levels. [1]

> **Simple Uncorroborated** authentication may be desired to maintain access statistics or in a private network where the initiator is implicitly trusted and there is no need to incur the additional overhead of more sophisticated authentication methods.
>
> **Simple Corroborated** authentication may be necessary in situations where strong authentication is not practical, (i.e., international connection, no knowledge of algorithms in use, etc).
>
> **Strong** authentication will be required for secure environments.

A DSA that implements Simple Corroborated authentication will check the user password by means of a compare operation on the user's entry. If no user password is supplied (Simple Uncorroborated authentication) the DSA will validate the presence of the entry for

[1] It is the case that some DSAs containing public information may not require authentication.

the user, by a read operation or otherwise. The authentication will fail if the password is incorrect or if the user's entry does not exist.

A DSA that implements Simple Uncorroborated authentication without verification will accept simple credentials without validating them.

Implementations claiming conformance shall, as a minimum, implement **None** and **Simple Uncorroborated** authentication without verification.

11.8.5 Directory Service Conformance

The following sections will describe various aspects of Directory conformance. Conformance to the Authentication Framework is viewed as a separate issue from conformance to the rest of the Directory document and is presented in that context.

Directory Profiles are broken into two sections. Service support specifies the level of support for operations and errors. Protocol support specifies the protocol elements required for implementations which claim conformance to specified operations.

To specify the support for operations and errors, two classifications are used as follows.

1. r: required

   The operation shall be implemented and the respective error shall be handled for conformance to these agreements.

   For DUAs, *required* means:

   - For ARGUMENT parameters, create the DAP protocol elements to convey the service request to the DSA.
   - For RESULT and ERROR parameters, accept the DAP protocol elements.

   For DSAs, *required* means:

   - For ARGUMENT parameters, accept the protocol elements when received and create the protocol elements when acting as a requesting DSA.
   - For RESULT and ERROR parameters, be able to convey all possible results when responding in either the DAP or DSP protocols and when receiving results, perform additional processing as defined for cooperating DSAs.

2. n: not required

   It is left to implementations as to whether the operation or error is implemented or not.

To specify the support for protocol elements, four classifications are used as follows.

1. M: mandatory

   Generation of element is a mandatory static conformance requirement (i.e., a conformant implementation shall be capable of generating the element).

   Generation of element is a mandatory dynamic conformance requirement (i.e., the element shall be present in all instances of communication which use the element).

The terms *static conformance* and *dynamic conformance* are defined in ISO 9646–1, "OSI Conformance Testing Methodology and Framework, Part 1: General Concepts."

2. G: generate

   Generation of element is a mandatory static conformance requirement.

   Generation of element is a conditional dynamic conformance requirement; the condition is:

   > Where a DSA is a propagating DSA, it shall be capable of generating the protocol element as received in related APDUs received from other DSAs. Where the DSA is a holding DSA, it shall be capable of creating all possible values of a protocol element unless otherwise noted in the "Comments" line.

3. S: support

   When receiving protocol elements, implementations of these agreements shall be capable of accepting these elements without error. Actions specified in the Directory documents and in these agreements shall be taken.

4. O: optional

   When generating protocol elements:

   - Generation of element is an optional static conformance requirement. If the implementor claims support for the corresponding Directory capability, then the implementation shall be capable of generating the element.
   - Generation of element is an optional dynamic conformance requirement. If the implementor claims support for the corresponding Directory capability, then the element shall be present in instances of communication which use the element (except where defaults allow otherwise).

   When receiving protocol elements, implementations of these agreements shall be capable of accepting these elements without error. However, actions specified in the base standard and in these agreements may be taken but are not required.

Where protocol elements are nested, the classification of the nested protocol elements is of relevance only when the immediately containing protocol element is generated. The classification of the protocol elements at the highest level is relative with respect to support of the operation.

Also note that in table 11.3, some rows contain two support classifications in the DSA column. In such cases, the support classification in parentheses applies to centralized DSA's only. When there is only one support classification given, it applies equally to centralized and non-centralized DSA's.

11.8.6 The Directory Access Profile

This agreement requires implementations of the DUA to provide access to the Directory Services as defined in the DUA column in table 11.2. For the services in table 11.2 which are supported, these agreements further require DUAs to support the protocol elements as defined in the DUA column in table 11.3 (parts 1 – 7).

These agreements require implementations of the DSA to support the Directory Services as defined in the DSA column in table 11.2 (page 11 – 40). These agreements further require DSAs to support the protocol elements as defined in the DSA column in table 11.3. Table 11.3 is listed in seven parts (page 11 – 41 through page 11 – 47). Note that the requirements for a centralized DSA and a cooperating DSA are different.

11.8.7 The Directory System Profile

These agreements require implementations of distributed DSAs which provide DSP to support the responder role for services as defined in table 11.4 (page 11 – 48). Further, these agreements require DSAs to support the protocol elements as specified in table 11.5. Table 11.5 is listed in nine parts (page 11 – 49 through page 11 – 57).

DSAs are required to support the requestor role for all the services as defined in table 11.4 if conforming to the chained mode of interaction.

11.8.8 Digital Signature Protocol Conformance Profile

Table 11.6 on page 11 – 58 and table 11.7 on page 11 – 58 provide information on the digital signature protocol conformance profile.

Note that elements in CommonArguments and CommonResults SecurityParameters that are not specified in table 11.6 and table 11.7 are covered in the Directory Service Protocol Support (table 11.5) and Directory Access Protocol Support (table 11.3).

11.8.9 Strong Authentication Protocol Conformance Profile

table 11.8 on page 11 – 59 and table 11.9 on page 11 – 60 provide information on the strong authentication protocol conformance profile.

11.9 DISTRIBUTED OPERATIONS

The following requirements apply to DSAs supporting distributed operations:

> DSAs supporting authentication (e.g., simple authentication by name and password) shall be able to invoke DSP operations to carry out authentication by reference to other DSAs. Thus all such DSAs shall support the DSP protocol. The requirement is implied by the Directory Documents.

11.9.1 Referrals and Chaining

It is recommended that a DSA which has chained a request act upon any referrals it receives rather than returning them to the requestor if the "preferChaining" service control is present.

11.9.2 Trace Information

A TraceInformation value carries forward a record of the DSAs which have been involved in the performance of an operation. It is used to detect the existence of, or avoid, loops which might arise from inconsistent knowledge or from the presence of alias loops in the DIT.

Each DSA which is propagating an operation to another, adds a new item to the trace information. If the propagation of a Search operation involves the creation of a new Search (cf Directory Documents, Part 4, clause 18.7.2.2.2), the trace information shall not be re–set, but the full trace information for the overall Search operation to the point where the new Search was generated shall be included in the new Search.

11.10 UNDERLYING SERVICES

This section specifies requirements over and above those given in the Directory Documents.

11.10.1 ROSE

It should be noted that support of "abandon" implies support of operation class 2.

11.10.2 Session

All directory implementations are required to support Session Version 2.

11.10.3 ACSE

The A-ABORT service is required by association–accepting DSAs to escape unwanted associations, which, under the ROSE protocol, they cannot release. In all other cases (association-initiating DSAs and DUAs) it may be preferable (though not required) to escape associations using UNBIND rather than abort.

The aborting DUA or DSA may optionally use the user information field of the A–ABORT. Such information, however, is only meaningful for diagnostic purposes and its use is not covered by these Agreements.

11.11 ACCESS CONTROL

Guidelines relating to access control can be found in Annex F of the Directory Documents, Part 2.

11.12 TEST CONSIDERATIONS

This section outlines some items that implementors may wish to consider in terms of testing expectations; additionally, future conformance testers may wish to consider these items when developing tests.

11.12.1 Major Elements of Architecture

One important aspect of testing is to confirm the correct behavior of DSAs and DUAs with respect to major elements of the directory architecture.

Such major elements include:

- Conformance Statement
- Distinguished names (e.g., name resolution, equivalence of various forms)
- Entries and Attributes (e.g., accessibility by operations, compliance with rules)
- Handling of distributed operations (e.g., naming contexts and knowledge)
- Schemas
  - Structure rules (e.g., storage and maintenance of structure and of naming rules)
  - Object classes and sub-classes (e.g., storage and extension of rules for object attributes)
  - Attribute types (e.g., storage and maintenance of syntax classes and rules for multi or single valued attributes)
  - Attribute syntax (e.g., maintenance and support for attribute value testing and matching, to specification for a defined set of attribute types)
- Operations
  - all operations
  - correct function
  - correct result
  - correct responses
- Aliases (e.g., correct resolution, error responses)
- Authentication and Access Control (e.g., limitation of modify access)
- ROSE (e.g., correct handling of invokes, results, rejects, and invoke ids)
- ACSE (e.g., association establishment / refusal for invalid application contexts, etc.)

11.12.2 Search Operation

Testing of support for filter items should be reasonable. It is not expected that DSAs will be able to handle worst case testing in this area.

11.13 ERRORS

This section provides clarification of the semantics of various operation errors and implementation guidelines on their usage.

11.13.1 Permanent vs. Temporary Service Errors

This section provides some clarification regarding the usage of the Service Errors *busy*, *unavailable*, and *unwillingToPerform*.

The error *busy* is particularly transient. It is returned when one or more of The Directory's internal resources are being used to their capacity and, hence, the requested operation cannot, for the moment, be performed. The Directory should be able to recover from this type of resource depletion after a short while.

The error *unavailable* is also temporary but somewhat less transient. It indicates that The Directory (or some part of it) is currently unavailable and may continue to be unavailable for a reasonably long period of time. For example, this error is returned when a given DSA is functionally disabled, or when a specific part of the DIB is undergoing reconfiguration.

The error *unwillingToPerform* has a permanent connotation. It indicates that The Directory cannot perform the requested operation because it would require resources beyond its capacity. For example, this error may be returned by a DSA if satisfying a request would result in the generation of an APDU in excess of 32767 octets.

11.13.2 Guidelines for Error Handling

11.13.2.1 Introduction

This subsection provides a recommended mapping of error situations which may be encountered to ROSE Rejects or to the errors provided in the DAP and DSP protocols of the Directory Documents.

The Directory Documents are not adequately definitive about the handling of errors. In this document, more explicit guidelines are given.

Error situations are defined by:

- Symptom (i.e., the manner in which the error was detected).
- Situation (i.e., the circumstance or phase during which the error was detected. For each possible situation, the error-handling procedure needs to be defined).

11.13.2.2 Symptoms

Table 11.10 (page 11 – 61 to page 11 – 63) describes a set of symptoms; the set is not necessarily exhaustive. Each is identified by a title which is used later in describing error actions. The title used for each symptom is not intended to imply any particular usage in a particular implementation.

11.13.2.3 Situations

Table 11.11 (page 11 – 64) identifies recognized situations within which particular symptoms may give rise to distinct error actions.

11.13.2.4 Error Actions

Table 11.13 (page 11 – 66 to page 11 – 71) summarizes specific error actions for each possible combination of symptom and situation. Symptoms are described in section 11.13.2.2 and situations are described in section 11.13.2.3.

Each entry in table 11.13 corresponds to the symptom in the left–most column and the situation given in the column header. Each entry may specify:

- a specific error action. The error action is described using the notation shown in table 11.12.
- a specific error action and a relevant note. The note will be indicated by a number enclosed in parentheses. The notes can be found on page 11 – 72.
- only a relevant note.
- a blank (which indicates the corresponding combination of symptom and situation is not meaningful in the context of these Agreements).

The entries in table 11.13 which specify a specific error action will do so using the notation shown in table 11.12 (page 11 – 65).

11.13.2.5 Reporting

In addition to the use of error–reporting services, DSAs should implement logging services to assist in management of the Directory. The list below describes classes of error which should be logged. Note that the list is not necessarily complete.

1. Errors indicating attempted breaches of security.
2. Errors indicating local software or hardware malfunction.
3. Errors indicating malfunction or other unacceptable behavior on the part of the invoker of an operation.
4. Errors indicating loss of chaining service by another DSA.
5. Error conditions that would be difficult to diagnose with the level of detail supplied over the protocol.
6. Aborts and other exceptional communications events.

The form and accessibility of any such logs is for further study.

11.14 Specific Authentication Schemes

This section describes identified authentication algorithms. Use of algorithms in this section is not mandatory. Use of algorithms other than those described in this section or described in the Directory Documents is by bilateral agreement.

11.14.1 Specific Strong Authentication Schemes

This section provides information on one alternative to the RSA digital signature scheme. The alternative is identified as the "ElGamal" digital signature scheme. Future contributions may result in other alternatives being added to this section.

Implementors may choose to provide digital signature capability based on RSA, ElGamal, or some other scheme appropriate for use in the OSI Directory environment.

It should be noted that use of RSA is governed by U.S.A. patent law.

11.14.1.1 ElGamal

The information in this section includes a tutorial description of the ElGamal scheme for digital signature using the notation defined in the Directory Documents, Part 8. It is intended that much of the tutorial information provided in this section will be moved to the security agreements sometime in the future.

11.14.1.1.1 References

[ELGA85] ElGamal T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. on Inform. Theory*, vol. IT-31, No. 4, July 1985.

[DIFF76] Diffie W., Hellman M., "New Directions in Cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, Nov. 1976

[COPP86] Coppersmith, D., Odlyzko, A., Schroeppel, R., "Discrete Logarithms in $GF(p)$," *Algorithmica*, vol. 1, 1986.

[McCl79] McClellan, J., Rader, C., *Number Theory in Digital Signal Processing*, Prentice–Hall, 1979.

[PATT87] Patterson, W., *Mathematical Cryptology for Computer Scientists and Mathematicians*, Rowman & Littlefield, 1987.

[ODLY] Odlyzko, A., "On the Complexity of Computing Discrete Logarithms and Factoring Integers," to appear in *Fundamental Problems in Communication and Computation*, B. Gopinath and T.Loven, Eds., New York, NY:Springer.

[ODLY84] Odlyzko, A., "Discrete Logarithms in Finite Fields and Their Cryptographic Significance," in *Advances in Cryptology, Proceedings of EUROCRYPT 84*. New York, NY:Springer–Verlag, pp. 224–314.

[ELGA85b] ElGamal, T., "A Subexponential–time Algorithm for Computing Discrete Logarithms over $GF(p^2)$," *IEEE Trans. Inform. Theory*, vol. IT-31, July 1985.

[SIER88] Sierpinski, W., *Elementary Theory of Numbers*, North–Holland 1988.

[RFC1115] Linn, J., *Privacy Enhancement for Internet Electronic Mail: Part III – Algorithms, Modes, and Identifiers*, RFC–1115, August 1989, IAB Privacy Task Force.

11.14.1.1.2 Background

The ElGamal digital signature scheme is based on earlier work done by Diffie and Hellman [DIFF76] in which it was suggested that a likely candidate for a one–way function is the *discrete exponential function*

$$f(x) \equiv \alpha^x \pmod{p} \tag{1}$$

where x is an integer between 1 and $p-1$ inclusive, where p is a very large prime number, and where α is an integer such that $1 \leq \alpha \leq p$ and $\{\alpha \bmod p, \alpha^2 \bmod p, \cdots, \alpha^{p-1} \bmod p\}$ is equal to the set $\{1, 2, \cdots, p-1\}$. In algebraic terminology, such an α is called a *primitive element.* References on the topic of primitive roots and elements are [McCl79] and [PATT87].

Now, in the real number system, if $y = \alpha^x$, then by definition of the logarithm we can solve for x using $x = \log_\alpha(y)$. The same idea extends to solving eq (1) for x so that inverting $f(x)$ requires calculating *discrete logarithms.* The reason Diffie and Hellman suspected eq (1) is one–way is that for suitable p, it is computationally difficult to invert $f(x)$. According to the current state of the art, computing discrete logs for suitable p has been found to require a number of operations roughly equivalent to

$$\exp(\sqrt{cb \ln b}) \tag{2}$$

where b is the number of bits in p, and c is estimated at $c = .69$ according to [ODLY]. This can be compared to only about $2 \log_2 p$ multiplications for discrete exponentiation. If in fact the best known algorithm for computing discrete logs is near optimal then Expression (2) is a good measure of the problem's complexity (for a properly chosen p) and the discrete exponential function has all the qualities of a one–way function as described by Diffie and Hellman.

11.14.1.1.3 Digital Signature

- Private Key: X_S denotes the private key for user X. X_S is a randomly chosen integer which user X keeps secret.
- Public Key : X_p denotes the public key for user X and is calculated using the corresponding private key such that

$$X_p \equiv \alpha^{X_S} \pmod{p} \tag{3}$$

  where

  - p is a prime satisfying the requirements listed in section 11.14.1.1.5.
  - α is a primitive element mod p.
  - Note that p and α could be used globally, but because they should be easily changeable (see sec. 11.14.1.1.5 for information about why these two parameters should be easily changeable) it would probably be preferable for each user to choose his/her own p and α. If users choose their own, then p and α must be made available to the recipient for use in the signature verification process.

- Signing Procedure: Suppose user A wants to sign a message intended for recipient B. The basic idea is to compute a two part signature (r, s) for the message m such that

$$\alpha^{h(m)} \equiv (A_p)^r r^s \pmod{p} \tag{4}$$

where h is a one–way hash function.

Compute the signature (r, s) as follows.

1. Choose a random number k, uniformly between 0 and $p-1$ such that k and $p-1$ have no common divisor except 1 (i.e., $\gcd(k, p-1) = 1$).
2. Compute r such that
$$r \equiv \alpha^k \pmod{p} \tag{5}$$
3. Use r to solve for the corresponding s as follows.
   (a) rewrite eq (4) using eq (5) and the definition of the public key to get
   $$\alpha^{h(m)} \equiv \alpha^{(A_S)r}\alpha^{ks} \pmod{p} \tag{6}$$
   Combining exponents, get
   $$\alpha^{h(m)} \equiv \alpha^{(A_S)r+ks} \pmod{p} \tag{7}$$
   eq (7) implies that
   $$h(m) \equiv (A_S)r + ks \pmod{p-1} \tag{8}$$
   Note that eq (8) has a single solution for s because k was chosen such that $\gcd(k, p-1) = 1$. See [SIER88] for supporting theorem.
   (b) now solve for s and get
   $$s \equiv I(h(m) - (A_S)r) \pmod{p-1} \tag{9}$$
   where I is computed such that $k * I \equiv 1 \pmod{p-1}$.

   The ElGamal signature is comparable in size to the corresponding RSA signature.

11.14.1.1.4 Verification

The recipient receives A_P, m, r, s, α, and p and computes both sides of eq (4) and then compares the results.

11.14.1.1.5 Known Constraints on Parameters

The following list of constraints is the result of a search of current literature and may not be complete.

1. p must be prime
2. p must be large.

   Note that Expression (2) can be used to speculate on the level of security afforded by cryptosystems based on the discrete log problem. Breaking the ElGamal scheme has not been proven to be equivalent to finding discrete logs, but if we **assume** equivalence then we can estimate how large p should be for a desired level of security.

   For instance, suppose we wanted to use Expression (2) to decide how large p should be so that we can be reasonably sure the system cannot be broken (using the best

known algorithm) in a practical amount of time. To be on the conservative side, we decide we want to protect against a special purpose machine that can perform 10^{15} operations per second. Specifically, we want to know how large p should be so that such a machine would take at least one year to break the system.

In one year, the hypothetical machine can perform 3×10^{22} operations. To find the size of the desired p, solve the following equation for b.

$$\exp(\sqrt{cb \ln b}) = 3 \times 10^{22} \tag{10}$$

We get $b \approx 606$. This is the number of bits in the desired p. So, the magnitude of the desired p is about 2^{606} which is roughly 266×10^{180}.

Hence, to be reasonably sure of attaining the desired level of security, we find a prime number greater than 266×10^{180} which satisfies all the other criteria listed in this section. Our confidence, however, is strictly based on the assumption that breaking ElGamal is as difficult as finding discrete logs and the assumption that the best known algorithm for finding discrete logs is near optimal.

3. p should occasionally be changed. This requirement is discussed in [ODLY84] and is related to the discovery of new algorithms for computing discrete logarithms in $GF(p)$.

4. $p - 1$ must have at least one large prime factor. This requirement is discussed in [ODLY84] and is imposed by the Silverman–Pohlig–Hellman algorithm which computes discrete logarithms in $GF(p)$ using on the order $\sqrt{r}$ operations and a comparable amount of storage, where r is the largest prime factor in $p - 1$.

5. p should not be the square of any prime. A subexponential–time algorithm for computing discrete logarithms in $GF(p^2)$ has been found. See [ELGA85b] for details.

11.14.1.1.6 Note on subjectPublicKey

The ASN.1 data element **subjectPublicKey**, defined as **BIT STRING** in Annex (G) of Directory Documents, Part 8, should be interpreted in the case of ElGamal as being of type:

```
SEQUENCE {INTEGER, INTEGER}
```

where the first integer is the Arithmetic Modulus and the second is the primitive element for the finite field. The sequence is represented by the ASN.1 Basic Encoding Rules.

Implementors should take note that the size of the integers used for these parameters is expected to exceed the pragmatic constraints specified for integers by the upper layers SIG.

11.14.1.2 One–Way Hash Functions

11.14.1.2.1 SQUARE–MOD–N Algorithm

Recent research regarding the square–mod–n one-way hash function described in Annex D of the Directory Documents, Part 8, has revealed that the function is not secure. Its use, therefore, is discouraged.

11.14.1.2.2 MD2 Algorithm

MD2 is a one–way hash function and is described in [RFC1115]. Implementors should note that the use of MD2 may be subject to license agreements.

11.14.1.2.3 Study of Other One–Way Hash Functions

The Directory SIG is studying the applicability of alternative one–way hash functions. Currently, this work includes review of SNEFRU; the Working Implementation Agreements report the status of the work.

11.14.1.2.4 Use of One–Way Hash Functions in Forming Signatures

MD2 may be used to form digital signatures in conjunction with RSA or ElGamal.

11.14.1.3 ASN.1 for Strong Authentication Algorithms

This section defines object identifiers assigned to authentication algorithms. The definitions take the form of the ASN.1 module, "OIWAlgorithmObjectIdentifiers".

```
OIWAlgorithmObjectIdentifiers {iso(1) identified-organization(3)
                               oiw(14) dssig(7)
                               oIWAlgorithmObjectIdentifiers(1)}
DEFINITIONS ::=
BEGIN

EXPORTS
    md2, md2WithRSA, elGamal, md2WithElGamal;

IMPORTS
authenticationFramework
          FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                                  usefulDefinitions(0)}

              ALGORITHM FROM AuthenticationFramework
                                                  authenticationFramework;

-- categories of object identifiers

algorithm OBJECT IDENTIFIER ::= {iso(1) identified-organization(3)
                                 oiw(14) dssig(7) algorithm(2)}

encriptionAlgorithm OBJECT IDENTIFIER ::= {algorithm 1}

hashAlgorthm OBJECT IDENTIFIER        ::= {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER  ::= {algorithm 3}
```

```
-- algorithms

md2 ALGORITHM
    PARAMETER NULL
   ::= {hashAlgorithm 1}

md2WithRsa ALGORITHM
    PARAMETER NULL
    ::= {signatureAlgorithm 1}

elGamal ALGORITHM
    PARAMETER NULL
    ::= {encryptionAlgorithm 1}
```

Editor's Note: Refer to the June 1990 Working Agreements for information regarding why PARAMETER NULL is specified above for the elGamal encryption algorithm.

```
md2WithElGamal ALGORITHM
    PARAMETER NULL
    ::= {signatureAlgorithm 2}

END -- of Algorithm Object Identifer Definitions
```

11.14.1.4 Note on the ENCRYPTED MACRO

The value associated with the ENCRYPTED MACRO, as defined in Directory Documents, part 8, clause 8.4 shall be interpreted in the case of ElGamal as being type:

```
SEQUENCE{ INTEGER, INTEGER }
```

The first integer in the sequence is r (see eq (5), sec. 11.14.1.1.3). The second integer is s (see eq (9), sec. 11.14.1.1.3).

11.14.2 Protected Simple Authentication

Protecting the user's distinguished name and password provides greater degrees of security than where passwords are not protected.

The procedure for achieving this protection, referred to as protected simple authentication, is outlined in the Directory Documents, Part 8, clause 5.3. The approach by which protected identifying information may be generated is outlined in the Directory Documents, Part 8, clause 5.4. For the purpose of these agreements, $f1$ and $f2$ as specified in the Directory Documents, Part 8, clause 5.4 are identical MD2 one–way functions. The algorithms for implementation of the MD2 one–way function are described in [RFC1115] (see

sec. 11.14.1.1.1). Note that the use of MD2 may be subject to licensing agreement. Use of other algorithms for other one–way functions is by bilateral agreement.

User *A* generates Protected2 as specified in the Directory Documents, Part 8, clause 5.4. Authenticator2 is then conveyed to *B* in the form of SimpleCredentials. Table 11.14 on page 11 – 73 shows the relationship between SimpleCredential fields and the elements of protected simple authentication as shown in figure 2 of the Directory Documents, Part 8.

11.14.3 Simple Authentication

There are two major classes of authentication supported by the Directory (i.e., simple and strong authentication). Simple authentication is based on a password being passed between the two associated entities (e.g., between a Directory User and a DUA, or between two DSAs). In the case of interaction between a Directory User and a DUA, the password is compared in some way with the password attribute in the user's entry in the Directory. In the case of interaction between two DSAs, this cannot be done since the DSA object class, as defined in the Directory Documents (Part 7, clause 6.14) does not contain a password attribute.

To facilitate simple authentication between DSAs, it is recommended that a DSA have local access to a list of one or more known DSAs, with a copy of each known DSA's password. Maintenance of that information is done through the use of bilateral agreements between DSA administrators.

11.15 APPENDIX A: MAINTENANCE OF ATTRIBUTE SYNTAXES

11.15.1 Introduction

The attribute types defined in the Directory Documents, Part 6, and listed in table 11.1 (page 11 – 38) have requirements, in DSAs which support them, for underlying algorithms that:

- check attribute values for syntactical correctness and compliance with pragmatic constraints;
- match attribute values (comparing for equality, for matching substrings, and for relative ordering).

11.15.2 General Rules

A DSA may receive a legitimately encoded attribute or AVA that is unsupported by the DSA. If the DSA is not required to act on it, or to store it within an entry, it may handle it by passing it on without error. Such attributes may also be used in search filter–item definitions: in this case, no error is reported, but the filter–item shall be deemed to be undefined for all entries in the DSA. This rule applies to occurrences of attributes in both operation arguments and results.

Conversely, a DSA must return a suitable error if an operation requires it to act on or store an attribute or AVA of type unsupported by the DSA. This constraint applies even for AVAs that are contained in attributes that take names as values, since the DSA will be unable correctly to match the attribute values without this attribute information.

11.15.3 Checking Algorithms

The subsections below give additional checks (beyond those directly implied by the Directory Documents) which shall be applied to attributes before they are stored in the DSA.

11.15.3.1 distinguishedNameSyntax

Each component AVA must be checked, unregistered attribute types comprising an error; check also that no two AVAs in the same RDN have the same attribute type.

11.15.3.2 integerSyntax

Local implementations may apply local limitations.

11.15.3.3 telephoneNumberSyntax

The value of policing further rules is for further study (this applies also to telexNumber, teletexTerminalIdentifier, facsimileTelephoneNumber, G3FacsimileNonBasicParameters, x121Address, and iSDNAddress).

11.15.3.4 countryName

The value must be checked for compliance with ISO3166: 1981 (E/F). (Note that from time to time further codes may be allocated.)

11.15.3.5 preferredDeliveryMethod

The values of the integer elements should not be restricted.

11.15.3.6 presentationAddress

No further checks should be applied.

11.15.4 Matching Algorithms

Matching algorithms are conveniently defined in terms of a two-step process:

1. Take the checked reference value, and the value to be matched, and, if necessary, reduce them to a canonical (i.e., standard) form (normalization) appropriate to each attribute syntax.
2. Carry out the comparison in the specified way (e.g., equality, substrings or ordering) using the appropriate rules for the value - character string, integer, boolean, etc.

Note that the lexical ordering of character strings (when supported) may be subject to local rules.

IMPORTANT NOTE: The combination of normalization and comparison may be replaced, in a particular implementation, by equivalent procedures. Additional notes on normalization are given below.

11.15.4.1 UTCTimeSyntax

If the "seconds" field is absent, it shall be inserted, and set to "00", and the form converted to the "Z" form. Note. The normalization strategy does not match times where the stored form omits the seconds field, and the compared form contains it, e.g.,

```
8804261919Z
    880426191926Z
```

(It might have been expected that these two forms, which coincide in time to within a few seconds, would be considered identical.)

11.15.4.2 distinguishedNameSyntax

For each attribute value, carry out normalization in accordance with the normalization rules defined for the type (if registered); values corresponding to unregistered attribute types are left unchanged at this stage.

11.15.4.3 caseIgnoreListSyntax

To facilitate matching, particularly for substrings, normalization may be considered in terms of a representation which replaces the separate ASN.1 elements by a single string with a delimiter.

11.16 APPENDIX B: GLOSSARY

The following abbreviations may be useful; not all are used within these agreements.

ACL	Access Control List
ACSE	Association Control Service Element
ADDMD	Administration Directory Management Domain
AETitle	Application Entity Title
APDU	Application Protocol Data Unit
ASE	Application Service Element
ASN.1	Abstract Syntax Notation - 1
AVA	Attribute Value Assertion
BRM	Basic Reference Model
CA	Certification Authority
CCITT	The International Telegraph and Telephone Consultative Committee
CEN	Committee for European Normalization
CENELEC	Committee for European Normalization Electronique
CEPT	(Committee of European Posts and Telephones)
COS	Corporation for Open Systems
DAP	Directory Access Protocol
DIB	Directory Information Base
DIT	Directory Information Tree
DMD	Directory Management Domains
DSA	Directory System Agent
DSP	Directory System Protocol
DUA	Directory User Agent
EWOS	European Workshop for Open Systems
FTAM	File Transfer, Access & Management
INTAP	Interoperability Technical Association for Information Processing, Japan
ISDN	Integrated Services Digital Network
ISO/IEC	International Organization for Standardization
KT	Knowledge Tree
LL	Lower layers of OSI model (layers 1-4)
MAP	Manufacturing Automation Protocol
MHS	Message Handling Systems
NIST	National Institute of Standards and Technology
NSAP	Network Services Access Point
OSI	Open Systems Interconnection
PKCS	Public Key Crypto System
POSI	Promotion for Open System Interconnection
PRDMD	Private Directory Management Domain
PSAP	Presentation Service Access Point
RDN	Relative Distinguished Name
ROSE	Remote Operations Service Element
SSAP	Session Service Access Point
SIG	Special Interest Group
SPAG	Standards Promotion & Application Group
TOP	Technical and Office Protocols
TSAP	Transport Service Access Point

UL	Upper layers of OSI model (layers 5-7)
UPU	Universal Postal Union

11.17 APPENDIX C: REQUIREMENTS FOR DISTRIBUTED OPERATIONS

The following material is included for tutorial purposes, and does not represent material additional to the Directory Documents. It is also not intended as a complete statement of requirements (the Distributed Operations part of the Directory Documents should be referred to for a complete treatment).

11.17.1 General Requirements

DSAs supporting distributed operations and claiming support of chaining must fully support DSP, as defined by the Directory Documents. DSAs supporting distributed operations must always be able to accept incoming DSP associations and invocations. DSAs claiming support of chaining must support:

- Loop detection
- Loop avoidance

In passing on operations (when chaining or multi-casting), the original DAP-supplied invocation must be passed on without change of content. In particular, there must be no alteration in any way of any primitive content.

The support of a facility for returning cross-references (Directory Documents, Part 4, clause 10.4.1) is optional.

To ensure that traceInformation can be analyzed properly, DSAs shall only possess names that are compliant with the recommendations of the Directory Documents, Part 7 (including Annex B).

11.17.2 Protocol Support

11.17.2.1 Usage of ChainingArguments

When using ChainingArguments: [2]

- *originator* need not be used if *requestor* in CommonArguments is used;
- *targetObject* shall not be used unless the target object differs from object/base object (if it is present, object/base object are ignored for purposes of name resolution);
- *operationProgress*, *traceInformation*, *aliasDereferenced*, *aliasedRDNs*, *referenceType*, and *timeLimit* shall be generated, accepted, and used in accordance with the Directory Documents;
- *returnCrossReferences* and info may optionally be generated, and shall always be accepted.

[2] In this section, the names of protocol elements (within ChainingArguments) are italicized.

11.17.2.2 Usage of ChainingResults

When using ChainingResults: [3]

- *crossReferences* and *info* may optionally be generated, and shall always be accepted.

[3] In this section, the names of protocol elements (within ChainingResults) are italicized.

11.18 APPENDIX D: GUIDELINES FOR APPLICATIONS USING THE DIRECTORY

11.18.1 Tutorial

11.18.1.1 Overview

Applications may have a requirement for Directory functionality. This tutorial provides assistance to those groups intending to specify Directory usage for a specific application (e.g., Message Handling Systems).

11.18.1.2 Use of the Directory Schema

11.18.1.2.1 Use of Existing Object Classes

Applications wishing to use the Directory should have determined within a standard, Implementor's Agreements, or on a propriety basis, the relevant Directory schema for their objects. Consider the following two examples:

1. Network management applications may with to define a SMAE object class.
2. File transfer applications may with to define a File Store object class.

Groups should examine relevant standards to determine if application-specific object classes or attributes have been already defined before considering any additional definition. These object classes and attributes may be found in a variety of places including a specific application standard (e.g., [Recommendation CCITT '88 X.402 | ISO 10021-2] and the Directory Documents.). Standardized object classes and attributes should be strongly considered before additional schema elements are created.

11.18.1.2.2 Kinds of Object Classes

There are effectively two kinds of object classes permitted within the Directory Documents: structural and auxiliary. The terms structural and auxiliary are used here for convenience when referring to particular kinds of object classes. The terms themselves are not defined in the Directory Documents.

Structural object classes have associated DIT structure rules (which control naming). Entries of this object class type are intended to be instantiated in Directory entries. A structural object class provides information on the base mandatory and optional content of a DIT entry.

An auxiliary object class provides information to enhance the mandatory and optional contents of entries. It is always used in conjunction with a structural object class.

The object class hierarchy is formed as a result of the definition of structural object classes, and the addition of auxiliary object classes.

For example, all object classes in the Directory Documents, Part 7, are structural except for strongAuthentication User and certificationAuthority. These two object classes should be considered auxiliary and used in conjunction with other, structural object classes.

11.18.1.2.3 Use of Unregistered Object Classes

The Directory Documents, Part 2, clause 9.4.1 provides a "special" form of object class called "unregistered." An unregistered object class is not assigned an object identifier. One of the uses for unregistered object classes is to provide a means of creating a single Directory entry which logically represents a variety of object classes. Uses for unregistered object classes include:

- Locally adding attributes to a predefined superclass;
- Locally making optional attribute types in a predefined superclass mandatory;
- Creating an object class derived from multiple superclasses, without needless proliferation of registered object classes.

For example, it may be advantageous to provide an entry which represents a person who is both a MHS and a FTAM user.

Unregistered object classes may best be illustrated by example. Consider an entry which represents a collection of company entries for Fizzy Company whose users have MHS O/R addresses. Using the guidelines above, the Fizzy Company defines an unregistered object class using the structural object class organizationalPerson from the Directory Documents, Part 7, and the auxiliary object class mhs-user from the MHS standards [Recommendation X.402 | ISO 10021-2] as follows:

```
fizzyCompanyPerson ::= OBJECT-CLASS
 SUBCLASS OF organizationalPerson, mhs-user
      MUST CONTAIN {}
      MAY CONTAIN {}
```

Note that no object identifier is assigned.

Also note that since there are not MUST or MAY CONTAIN's in the fizzyCompanyPerson Object Class, the last two lines of the object class assignment (i.e., "MUST CONTAIN MAY CONTAIN ") are optional. As with the registered form of object classes, an unregistered object class always inherits all the attributes in any of its superclasses. There is no mechanism defined whereby a subclass may selectively inherit attributes from its superclasses.

An unregistered object class always appears as a leaf in the Object Class tree. (i.e., An unregistered object class may not be a superclass of some other object class).

Using unregistered object classes in conjunction with multiple inheritance is useful as shown by figure 11.6 in which three ways of creating the same two object classes are shown. Either three, four, or five registered object classes are used.

Examples (a) and (c) in figure 11.6 are both better ways of defining the object classes than that in example (b), even though example (c) needs to use one more registered object

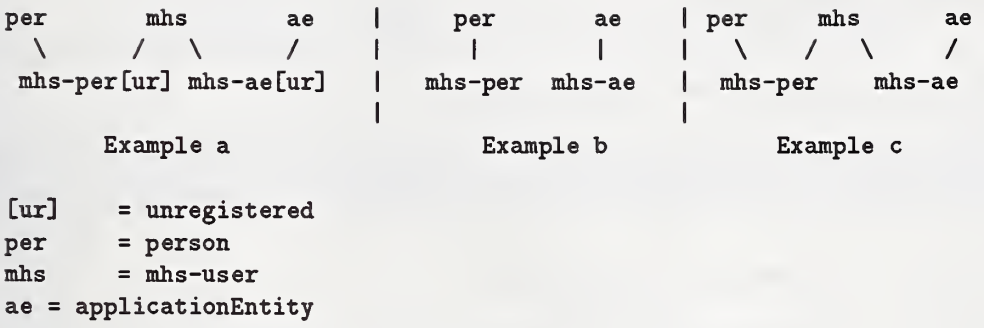


Figure 11.6. Three Ways of Creating Two Object Classes.

class than example (b). This is because the multiple inheritance technique, used in examples (a) and (c), enables a Directory User searching the Directory to easily create a filter to find all entries that contain mhs-user attributes, based on a value in the object class attribute (Each Directory entry contains a list of the object identifiers of the object classes it has inherited from, so the filter would just have to find all entries that held the object identifier value of mhs-user).

Example (a), which uses three registered object classes, is better than example (c), which uses five, because registering the extra two object classes does not provide any advantage over not registering them, and the first method avoids needless proliferation of registered object classes.

11.18.1.2.4 Side Effects of Creating Unregistered Object Classes

This section discusses two side effects of creating unregistered object classes.

1. When an unregistered object class is defined from a single superclass, there is no means available to distinguish between the two. Within the local scope for which the unregistered class is defined, all relevant entries are considered to belong to the unregistered class.

   The following is an example of this problem:

   > An object class of oC1(reg) has attribute type at1 mandatory and at2 optional. An unregistered form of this, oC1(unreg) is created, which makes at2 mandatory. When an Add Entry operation is received with both attributes present, the entry could belong to either form of oC1; it is indeterminate. After the entry is added a Modify Entry operation is received which requests the removal of attribute type at2. It is not clear if this operation should succeed, or whether an object class violation should be reported. If the attribute may be removed, then the entry belonged to the oC1(reg) object class and the unregistered form never existed, otherwise if the attribute may not be removed, then the entry belonged to oC1(unreg) and the registered form no longer exists.

2. More than one unregistered object class cannot be defined from the same superclass(es) for use within the same local scope, as there is no means available to distinguish the

classes from one another.

11.18.2 Creation of New Object Classes

If no appropriate object class is available, a new object class may be defined. This should only be done if no standardized object classes and attributes can fulfill the requirements.

11.18.2.1 Creation of New Subclasses

Generally, an application-specific object class is defined as a subclass of a pre-existing Directory object class. These object classes are specified in the Directory Documents, Part 7. The subclass may be structural or auxiliary. Optional attributes of the superclass may be made mandatory. New attributes may also be added.

For example, MHS has used the Directory structural object class applicationEntity to derive the object class for their MHS-specific application entity MTAs.

If absolutely no relevant object class is available, an object class may be defined as a subclass of the basic object class called "Top".

If no appropriate object class is available, a new object class may be defined. This should only be undertaken if no standardized object class can fulfill the requirements. When defining new object classes the object–class macro, as defined in the Directory Documents, Part 2, clause 9.4.6, should be used.

If new subclasses are defined, suggested or required name forms may also be specified in text.

11.18.2.2 Creation of New Attributes

If no appropriate attributes are available, a new attribute type may be defined. This should only be undertaken if no standardized attributes can fulfill the requirements. When defining new attributes the attribute macro, as defined in the Directory Documents, Part 2, clause 9.5.3, should be used.

11.18.3 DIT Structure Rules

Applications may desire to provide guidance on DIT structure rules and naming. As with object classes, standardized or suggested structure (including naming) rules from the Directory Documents part 7, Annex B and application-specific standards should be consulted before providing new structure rules. Annex B in the Directory Documents, Part 7, provides guidelines on how to specify this information. Structure rules associated with superclasses should be adopted wherever suitable.

11.19 APPENDIX E: TEMPLATE FOR AN APPLICATION SPECIFIC PROFILE FOR USE OF THE DIRECTORY

The template defined below should be used by OIW SIGs intending to specify Directory usage. Such application specific profiles shall be contained in application specific chapters of the OIW agreements. The information under each heading should be filled in (the text under each heading provides guidance on the intent of the section and should not be included in the profile).

- PROFILE TITLE

  Application specific profiles are named in the following way:

  `OIW <SIG-NAME> <DESCRIPTOR> DIRECTORY PROFILE`

  `(e.g., OIW DIRECTORY STRONG AUTHENTICATION DIRECTORY PROFILE )`

- OTHER PROFILES SUPPORTED

  Other OIW Directory profiles which are to be used by this specific application are listed here. Attributes, attribute sets, object classes and structure rules that are referenced in these profiles need not be enumerated below.

- STANDARD APPLICATION SPECIFIC ATTRIBUTES AND ATTRIBUTE SETS

  Any attributes supported from the relevant standards. For example, the MHS SIG might include mhs-or-address here.

- STANDARD APPLICATION SPECIFIC OBJECT CLASSES

  Any object classes supported from the relevant standards. For example, the MHS SIG might include mhs-user here.

- OIW APPLICATION SPECIFIC ATTRIBUTES AND ATTRIBUTE SETS

  This, optional, component of this profile allows for the specification of OIW application specific attributes and attribute sets. This section of this template should be used rarely and with consideration that no standard profile or attribute/attribute set exists which can be used.

- OIW APPLICATION SPECIFIC OBJECT CLASSES

  This, optional, component of this profile allows for the specification of OIW application specific object classes. This section of this template should be used rarely and with consideration that no standard profile or object class exists which can be used.

- STRUCTURE RULES

  Guidance for DIT structural rules, provided only when structure rules associated with superclasses are not adopted. The Directory Documents, Part 7, Annex B provide an example and guideline to use in specifying this information.

Table 11.1. Pragmatic Constraints for Selected Attributes (Part 1 of 2)

Attribute Type	Content	Constraints	Primary Source	Notes
Aliased Object Name	Distinguished Name			Note 3 (page 11 – 39)
Business Category	T.61 or Printable String	ub-business-category 128	CCITT X.520	
Common Name	T.61 or Printable String	ub-common-name 64	CCITT X.520	
Country Name	Printable String	2	ISO 3166	
Description	T.61 or Printable String	ub-description 1024	CCITT X.520	About 1 screen full
Facsimile Telephone Number	Facsimile Telephone Number	ub-telephone-number 32	CCITT X.520	Optionally includes G3 non–basic parameters (Upper bounds ffs)
International ISDN Number	Numeric String	ub-isdn-address 16	CCITT X.520	E.164 Internat'l ISDN Number
Knowledge Information	T.61 or Printable String	1024	NIST	About 1 screen full
Locality Name	T.61 or Printable String	ub-locality-name 128	CCITT X.520	
Member	Distinguished Name			Note 3 (page 11 – 39)
Object Class	Object Identifier	256 octets	NIST	
Organization Name	T.61 or Printable String	ub-organization-name 64	CCITT X.520	
Organizational Unit Name	T.61 or Printable String	ub-organizational-unit-name 64	CCITT X.520	
Owner	Distinguished Name			Note 3 (page 11 – 39)
Physical Delivery Office Name	T.61 or Printable String	ub-physical-office-name 128	CCITT X.520	
Post Office Box	T.61 or Printable String	ub-post-office-box 40	CCITT X.520	
Postal Address	Postal Address	ub-postal-line 6 ub-postal-string 30	CCITT X.520	UPU
Presentation Address	Presentation Address	224 octets	NIST	Note 2 (page 11 – 39), ISO 7498.3 & X.200
Role Occupant	Distinguished Name			Note 3 (page 11 – 39)
See Also	Distinguished Name			Note 3 (page 11 – 39)

Table 11.1. Pragmatic Constraints for Selected Attributes (Part 2 of 2)

Attribute Type	Content	Constraints	Primary Source	Notes
Serial Number	Printable String	ub-serial-number 64	CCITT X.520	
State or Province Name	T.61 or Printable String	ub-state-name 128	CCITT X.520	
Street Address	T.61 or Printable String	ub-street-address 128	CCITT X.520	
Surname	T.61 or Printable String	ub-surname 64	CCITT X.520	
Telephone Number	Printable String	ub-telephone-number 32	CCITT X.520	E.123
Teletex Terminal Identifier	Teletex Terminal Identifier	ub-teletex-terminal-id 1024	CCITT X.520	Optionally includes Teletex non–basic parameters (upper bound ffs)
Telex Number	Telex Number	ub-telex-number 14 ub-country-code 4 ub-answerback 8	CCITT X.520	Contains sequence of telex number, country code, and answerback
Title	T.61 or Printable String	ub-title 64	CCITT X.520	
User Password	Octet String	ub-user-password 128	CCITT X.520	Allow long passwords generated by machine
X.121 Address	Numeric String	ub-x121-address 15	CCITT X.520	X.121

Notes for table 11.1

1. The pragmatic constraints of these parameters are defined in other standards. We will accommodate these values in our pragmatic constraints.

2. Presentation address is composed of "X" NSAP addresses, and three selectors, $(20X + 32 + 16 + 16)$, e.g., if $X = 1$, this would be 84. These numbers are based on the most recent implementors' agreements. With 8 NSAP addresses this value is 224.

3. Pragmatic constraints are only applied to the individual components of Distinguished-Name as defined in the Directory Documents, Part 2. Not all components of a DN will necessarily be understood by an implementation.

4. Implementors should be aware that constraints on Postal Address may not be sufficient for some markets.

Table 11.2. Directory Access Service Support.

Operations and Errors	Support Classification DUA	DSA	Comments
– BIND and UNBIND –			
DirectoryBind	r	r	
DirectoryUnbind	r	r	
– OPERATIONS –			
– READ OPERATIONS –			
Read	n	r	
Compare	n	r	
Abandon	n	r (note 2)	
– SEARCH OPERATIONS –			
List	n	r (note 1)	
Search	n	r (note 1)	
– MODIFY OPERATIONS –			
AddEntry	n	r	
RemoveEntry	n	r	
ModifyEntry	n	r	
ModifyRDN	n	r	
– ERRORS –			
Abandoned	(note 4)	r	
AbandonedFailed	(note 4)	r	
AttributeError	(note 4)	r	
NameError	(note 4)	r	
Referral	(note 4)	r (note 3)	
SecurityError	(note 4)	r	
ServiceError	(note 4)	r	
UpdateError	(note 4)	r	

Notes for table 11.2

1. As performance of Search and List operations can consume significant resources, the policies of some centralized DSAs may be that such operations will not be performed. For these cases, the reply to the requests for such operations would be ServiceError with the "unwillingToPerform" Service Problem.

2. Reference Directory Documents, Part3, clause 9.3.6

3. Centralized DSAs would not generate referrals.

4. See EntryInformationSelection information under Common Data Types (table 11.3, Part 6)

Table 11.3. DAP Protocol Support (Part 1 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
– BIND and UNBIND –			
DirectoryBind			
DirectoryBindArgument	M	S	
credentials	O	S	
simple	O	S	
name	G	S	
validity	O	O	
password	G	S	
strong	O	O	See Strong Authentication Protocol Conformance Profile for requirements when strong authentication is supported.
externalProcedure	O	O	
versions	O	S	Supported value: v1988
DirectoryBindResult	S	G	
credentials	O	G	Shall be the same CHOICE as in DirectoryBindArgument.
simple	O	G	
name	S	G	
validity	O	O	
password	O	O	
strong	O	O	See Strong Authentication Protocol Conformance Profile for requirements when strong authentication is supported.
externalProcedure	O	O	
versions	S	O	Supported value: v1988
DirectoryBindError	S	G	
versions	S	O	Supported value: v1988
ServiceProblem	S	G	Supported value: unavailable
SecurityProblem	S	G	Supported values: inappropriateAuthentication, invalidCredentials
DirectoryUnbind			The DirectoryUnbind has no arguments.

Table 11.3. DAP Protocol Support (Part 2 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
– OPERATIONS, ARGUMENTS AND RESULTS –			
– READ OPERATIONS –			
Read			
ReadArgument	M	S	
object	M	S	
selection	O	S	See note 2 on page 11 – 47.
CommonArguments	O	S	
ReadResult	S	G	
entry	S	M	
CommonResults	S	G	
Compare			
CompareArgument	M	S	
object	M	S	
purported	M	S	
CommonArguments	O	S	
CompareResult	S	G	
DistinguishedName	S	G	
matched	S	M	
fromEntry	S	G	
commonResults	S	G	
Abandon			
AbandonArgument	M	S	
invokeId	M	S	
AbandonResult	S	G	
– SEARCH OPERATIONS –			
List			
ListArgument	M	S	
object	M	S	
CommonArguments	O	S	
ListResult	S	G	
listInfo	S	G	
DistinguishedName	S	G	
subordinates	S	M	
Rel.DistinguishedName	S	M	For the case where subordinates is empty set, RDN is absent.
aliasEntry	S	G	
fromEntry	S	G	
partialOutcomeQualifier	S	G	
CommonResults	S	G	
UncorrelatedListInfo	S	G (O)	
ListResult	S	G	See note 1 on page 11 – 47 for additional information related to the DSA support classification.

Table 11.3. DAP Protocol Support (Part 3 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
Search			
SearchArgument	M	S	
baseObject	M	S	
subset	O	S	
filter	O	S	
searchAliases	O	S	
selection	O	S	
CommonArguments	O	S	
SearchResult	S	G	
searchinfo	S	G	
DistinguishedName	S	G	
entries	S	M	
partialOutcomeQualifier	S	G	
CommonResults	S	G	
uncorrelatedSearchinfo	S	G (O)	
SearchResult	S	G	
partialOutcomeQualifier	S	G	
limitProblem	S	G	
unexplored	S	G	
unavailableCriticalExt	S	O	
– MODIFY OPERATIONS –			
AddEntry			
AddEntryArgument	M	S	
object	M	S	
entry	M	S	
CommonArgument	O	S	
AddEntryResult	S	G	
RemoveEntry			
RemoveEntryArgument	M	S	
object	M	S	
CommonArguments	O	S	
RemoveEntryResult	S	G	
ModifyEntry			
ModifyEntryArgument	M	S	
object	M	S	
changes	M	S	At least one entry modification must be supported.
addAttribute	O	S	
removeAttribute	O	S	
addValues	O	S	
removeValues	O	S	
CommonArguments	O	S	
ModifyEntryResult	S	G	

Table 11.3. DAP Protocol Support (Part 4 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
ModifyRDN			
ModifyRDNArgument	M	S	
object	M	S	
newRDN	M	S	
deleteOldRDN	O	S	
CommonArguments	O	G	
ModifyRDNResult	S	G	
- ERRORS AND PARAMETERS -			
Abandoned			
AbandonFailed			
problem	S	M	
operation	S	M	
AttributeError			
object	S	M	
problems	S	M	Min. 1 error (See Directory Documents, Part 3, clause 12.4.2.2)
type	S	M	
value	S	G	
NameError			
problem	S	M	
matched	S	M	
Referral			
candidate	S	G	

Table 11.3. DAP Protocol Support (Part 5 of 7)

Protocol Element	Support Classification		Comments
	DUA	DSA	
SecurityError			
problem	S	M	
ServiceError			
problem	S	M	
UpdateError			
problem	S	M	
– COMMON ARGUMENTS / RESULTS –			
CommonArguments			
ServiceControls	O	S	
SecurityParameters	O	S	See Section 11.8.8.
certification-path	O	S	
name	O	S	
time	O	S	
random	O	S	
target	O	S	
requestor	O	S	
OperationProgress	O	S (O)	
nameResolutionPhase	M	S	
nextRDNToBeResolved	O	S	
aliasedRDNs	O	S (O)	
extensions	O	S	
identifier	M	S	
critical	O	S	
item	M	S	
CommonResults			
SecurityParameters	O	G (O)	See Section 11.8.8.
certification-path	O	G	
name	O	G	
time	O	G	
random	O	G	
target	O	G	
performer	O	G (O)	
aliasDereferenced	O	G	

Table 11.3. DAP Protocol Support (Part 6 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
– COMMON DATA TYPES –			
ServiceControls			
options	O	S	
priority	O	S	
timeLimit	O	S	
sizeLimit	O	S	
scopeOfReferral	O	S	
EntryInformationSelection			
attributeTypes	O	S	
allAttributes	O	S	Must support at least one of the CHOICE.
select	O	S	
infoTypes	O	S	
EntryInformation			
DistinguishedName	S	M	
fromEntry	S	G	
SET OF CHOICE	S	G	
AttributeType	S	G	
Attribute	S	G	
Filter			Must support at least one of the CHOICE.
item	O	S	
and	O	S	
or	O	S	
not	O	S	
FilterItem			
equality	O	S	
substrings	O	S	
type	M	S	
strings	M	S	
initial	O	S	Must support at least one of the CHOICE.
any	O	S	
final	O	S	
greaterOrEqual	O	S	
lessOrEqual	O	S	
present	O	S	
approximateMatch	O	S	

Table 11.3. DAP Protocol Support (Part 7 of 7)

Protocol Element	Support Classification DUA	DSA	Comments
SecurityParameters	O	O	See Section 11.8.8.
certification-path	O	S	
name	O	S	
time	O	S	
random	O	S	
target	O	S	
ContinuationReference			
targetObject	O	M	
aliasedRDNs	O	G	
OperationProgress	O	M	
nameResolutionPhase	O	M	
nextRDNToBeResolved	O	G	
rdnsResolved	O	G	
AccessPoint	O	M	
AccessPoint			
Name	O	M	
PresentationAddress	O	M	
pSelector	O	G	
sSelector	O	G	
tSelector	O	G	
nAddress	O	M	

Notes for table 11.3

1. As performance of Search and List operations can consume significant resources, the policies of some centralized DSAs may be that such operations will not be performed. For these cases, the reply to the requests for such operations would be ServiceError with the "unwillingToPerform" Service Problem.

2. See EntryInformationSelection information under Common Data Types (table 11.3, part 6)

Table 11.4. Directory System Service Support.

Operations and Errors	Support Classification		Comments
	Request	Response	
– BIND and UNBIND –			
DSABind	n (notes 1,2)	r	
DSAUnbind	n (notes 1,2)	r	
– OPERATIONS –			
– CHAINED READ OPERATIONS –			
ChainedRead	n (notes 1,2)	r	
ChainedCompare	n (notes 1,2)	r	
chainedAbandon	n (note 1)	r	
– CHAINED SEARCH OPERATIONS –			
ChainedList	n (note 1)	r	
ChainedSearch	n (note 1)	r	
– CHAINED MODIFY OPERATIONS –			
ChainedAddEntry	n (note 1)	r	
ChainedRemoveEntry	n (note 1)	r	
ChainedEntry	n (note 1)	r	
ChainedModifyRDN	n (note 1)	r	
– ERRORS –			
Abandoned	n (note 1)	r	
Abandonfailed	n (note 1)	r	
AttributeError	n (note 1)	r	
NameError	n (note 1)	r	
DSARefferal	n (note 1)	r	
SecurityError	n (note 1)	r	
SeviceError	n (note 1)	r	
UpdateError	n (note 1)	r	

Notes for table 11.4

1. Necessary when supporting the chained mode of interaction.

2. Some of these operations may be necessary to support distributed authentication. This requirement is distinct from support for chained mode of interaction.

Table 11.5. DSP Protocol Support (Part 1 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– BIND and UNBIND –			
DSABind			
DirectoryBindArgument	M	S	
credentials	G	S	
simple	G	S	
name	G	S	
validity	O	O	
password	G	S	
strong	O	O	See Strong Authentication Protocol Conformance Profile for requirements when strong authentication is supported.
externalProcedure	O	O	
versions	G	S	Supported value: v1988
DSABindResult	S	G	
credentials	S	G	Shall be the same CHOICE as in DirectoryBindArgument.
simple	S	G	
name	S	G	
validity	O	O	
password	S	G	
strong	O	O	See Strong Authentication Protocol Conformance Profile for requirements when strong authentication is supported.
externalProcedure	O	O	
versions	S	G	Supported value: v1988
DirectoryBindError	S	G	
versions	S	G	Supported value: v1988
ServiceProblem	S	G	Supported values: busy and unavailable.
SecurityProblem	S	G	Supported values: inappropriateAuthentication, invalidCredentials.
DSAUnbind			The DSAUnbind has no arguments.

Table 11.5. DSP Protocol Support (Part 2 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– OPERATIONS, ARGUMENTS AND RESULTS –			
– CHAINED READ OPERATIONS –			
ChainedRead			
ChainingArgument	M	S	
ReadArgument	M	S	
object	M	S	
selection	G	S	
CommonArguments	G	S	
ChainingResult	S	M	
ReadResult	S	M	
entry	S	M	
CommonResults	S	G	
ChainedCompare			
ChainingArgument	M	S	
CompareArgument	M	S	
object	M	S	
purported	M	S	
CommonArguments	G	S	
ChainingResult	S	M	
CompareResult	S	M	
DistinguishedName	S	G	
matched	S	M	
fromEntry	S	G	
CommonResults	S	G	
ChainedAbandon			
AbandonArgument	M	S	
invokeId	M	S	
AbandonResult	S	G	

Table 11.5. DSP Protocol Support (Part 3 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– OPERATIONS, ARGUMENTS AND RESULTS –			
– CHAINED SEARCH OPERATIONS –			
ChainedList			
ChainingArguments	M	S	
ListArgument	M	S	
object	M	S	
CommonArguments	G	S	
ChainingResults	S	M	
ListResult	S	M	
listInfo	S	G	
DistinguishedName	S	G	
subordinates	S	M	
Rel.DistinguishedName	S	M	
aliasEntry	S	G	
fromEntry	S	G	
partialOutcomeQualifier	S	G	
CommonResults	S	G	
uncorrelatedListInfo	S	G	
ListResult	S	G	
ChainedSearch			
SearchArgument	M	S	
baseObject	M	S	
sugset	G	S	
filter	G	S	
searchAliases	G	S	
selection	G	S	
CommonArguments	G	S	
ChainingResults	S	M	
SearchResult	S	M	
Searchinfo	S	M	
DistinguishedName	S	G	
entries	S	M	
partialOutcomeQualifier	S	G	
CommonResults	S	G	
uncorrelatedSearchinfo	S	G	
SearchResult	S	G	
partialOutcomeQualifier	S	G	
limitProblem	S	G	
unexplored	S	G	
unavailableCriticalExt	S	G	

Table 11.5. DSP Protocol Support (Part 4 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– CHAINED MODIFY OPERATIONS –			
ChainedAddEntry			
ChainingArguments	M	S	
AddEntryArgument	M	S	
object	M	S	
entry	M	S	
CommonArguments	G	S	
ChainingResults	S	M	
AddEntryResults	S	M	
ChainedRemoveEntry			
ChainingArguments	M	S	
RemoveEntryArgument	M	S	
object	M	S	
CommonArguments	G	S	
ChainingResults	S	M	
RemoveEntryResult	S	M	
ChainedModifyEntry			
ChainingArguments	M	S	
ModifyEntryArgument	M	S	
object	M	S	
changes	M	S	
addAttribute	G	S	
removeAttribute	G	S	
addValues	G	S	
removeValues	G	S	
CommonArguments	G	S	
ChainingResults	S	M	
ModifyEntryResult	S	M	
ChainedModifyRDN			
ChainingArguments	M	S	
ModifyRDNArgument	M	S	
object	M	S	
newRDN	M	S	
deleteOldRDN	G	S	
CommonArguments	G	S	
ChainingResults	S	M	
ModifyRDNResult	S	M	

Table 11.5. DSP Protocol Support (Part 5 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– ERRORS and PARAMETERS –			
Abandoned			
AbandonFailed			
problem	S	M	
operation	S	M	
AttributeError			Min. 1 error (see Directory Documents, part3, clause 12.4.2.2)
object	S	M	
problems	S	M	
problem	S	M	
type	S	M	
value	S	G	
NameError			
problem	S	M	
matched	S	M	
DSARefferal			
ContinuationReference	S	M	
contextPrefix	S	G	
SecurityError			
problem	S	M	

Table 11.5. DSP Protocol Support (Part 6 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
ServiceError	S	G	For Directory operations
problem	S	M	
UpdateError	S	G	
problem	S	M	
- COMMON ARGUMENTS / RESULTS -			
CommonArguments			
ServiceControls	G	S	
SecurityParameters	O	S	see Section 11.8.8.
requestor	G	S	
OperationProgress	G	S	
nameResolutionPhase	M	S	
nextRDNToBeResolved	G	S	
aliasedRDNs	G	S	
extensions	G	S	
identifier	M	S	
critical	G	S	
item	M	S	
CommonResults			
SecurityParameters	S	O	See Section 11.8.8.
requestor	S	G	
aliasDereferenced	S	G	

Table 11.5. DSP Protocol Support (Part 7 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– COMMON DATA TYPES –			
ServiceControls			
options	G	S	
priority	G	S	
timeLimit	G	S	
sizeLimit	G	S	
scopeOfReferral	G	S	
EntryInformationSelection			
attributeTypes	G	S	
allAttributes	G	S	
select	G	S	
infoTypes	G	S	
EntryInformation			
DistinguishedName	S	M	
fromEntry	S	G	
SET OF CHOICE	S	G	
AttributeType	S	G	
Attribute	S	G	
Filter			
item	G	S	
and	G	S	
or	G	S	
not	G	S	
FilterItem			
equality	G	S	
substrings	G	S	
type	G	S	
strings	G	S	
initial	G	S	
any	G	S	
final	G	S	
greaterOrEqual	G	S	
lessOrEqual	G	S	
present	G	S	
approximateMatch	G	S	

Table 11.5. DSP Protocol Support (Part 8 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
– COMMON DATA TYPES FOR DISTRIBUTED OPERATION –			
ChainingArguments			
originator	G	S	
targetObject	G	S	
operationProgress	G	S	
nameResolutionPhase	M	S	
nextRDNToBeResolved	G	S	
traceInformation	M	S	
aliasDereferenced	G	S	
aliasedRDNs	G	S	
returnCrossRefs	G	S	See Directory Documents, Part 4, clause 10.4.1
referenceType	G	S	
DomainInfo	O	O	
timeLimit	G	S	
SecurityParameters	O	S	See note 1 (page 11 – 57) regarding the support classification for Request. Also see Section 11.8.8
ChainingResults			
Info	O	O	
crossReferences	S	G	
SecurityParameters	S	O	See note 1 (page 11 – 57) regarding the support classification for Response. Also see Section 11.8.8
CrossReference			
contextPrefix	S	M	See Directory Documents, Part 4, clause 12.4.2.2
accessPoint	S	M	
TraceInformation			
TraceItem	M	S	
TraceItem			
dsa	M	S	
targetObject	G	S	
operationProgress	M	S	
nameResolutionPhase	M	S	
nextRDNToBeResolved	G	S	

Table 11.5. DSP Protocol Support (Part 9 of 9)

Protocol Element	Support Classification		Comments
	Request	Response	
ContinuationReference			
targetObject	S	M	
aliasedRDNs	S	G	
operationProgress	S	M	
nameResolutionPhase	S	M	
nextRDNToBeResolved	S	G	
rdnsResolved	S	G	
referenceType	S	G	
AccessPoint	S	M	
AccessPoint			
Name	S	M	
PresentationAddress	S	M	
pSelector	S	G	
sSelector	S	G	
tSelector	S	G	
nAddress	S	M	

Notes for table 11.5

1. The support classification is G when supporting the chained mode of interaction.

2. Some of these operations may be necessary to support distributed authentication. This requirement is distinct from support for chained mode of interaction.

Table 11.6. DAP Support for Digital Signature Protocol Conformance Profile.

Protocol Element	Support Classification		Comments
	DUA	DSA	
– COMMON ARGUMENTS / RESULTS –			
CommonArguments			
SecurityParameters			
certification-path	G	S	
name	G	S	
time	G	S	
random	G	S	
target	G	S	
requestor	G	S	
CommonResults			
SecurityParameters	S	G	
performer	S	G	

Table 11.7. DSP Support for Digital Signature Protocol Conformance Profile.

Protocol Element	Support Classification		Comments
	DUA	DSA	
– COMMON ARGUMENTS / RESULTS –			
CommonArguments			
SecurityParameters			
certification-path	G	S	
name	G	S	
time	G	S	
random	G	S	
target	G	S	
requestor	G	S	
CommonResults			
SecurityParameters	G	S	
performer	O	G	

Table 11.8. DAP Support for Strong Authentication Protocol Conformance Profile.

Protocol Element	Support Classification		Comments
	DUA	DSA	
DirectoryBindArgument	M	S	
credentials	G	S	
simple	G	S	
name	G	S	
validity	G	S	
password	G	S	
strong			
certification-path	G	S	
bind-token	G	S	
externalProcedure	O	O	
versions	O	S	
DirectoryBindResult	S	G	
credentials	S	G	
simple	S	G	
name	S	G	
validity	S	G	
password	S	G	
strong	S	G	
certification-path	S	G	
bind-token	S	G	
externalProcedure	O	O	
versions	S	O	

Table 11.9. DSP Support for Strong Authentication Protocol Conformance Profile.

Protocol Element	Support Classification		Comments
	DUA	DSA	
DirectoryBindArgument	M	S	
credentials	G	S	
simple	G	S	
name	G	S	
validity	G	S	
password	G	S	
strong			
certification-path	G	S	
bind-token	G	S	
externalProcedure	O	O	
versions	O	S	
DirectoryBindResult	S	G	
credentials	S	G	
simple	S	G	
name	S	G	
validity	S	G	
password	S	G	
strong	S	G	
certification-path	S	G	
bind-token	S	G	
externalProcedure	O	O	
versions	S	O	

Table 11.10. Error Symptoms (Part 1 of 3)

Symptom	Descripton
E_ACCESS	The initiator has insufficient access rights to carry out this operation.
E_ADMIN_LIMIT	The Directory has reached some limit set by an administrative authority, and no partial results are available to return to the user.
E_ALIAS_DEREF	One of three situations exists: 1. An alias has been encountered while a previous alias was being dereferenced, or 2. a name contained an alias plus one or more additional RDNs when the dontDereferenceAliases service control was being used, or 3. the name, supplied in an operation that precludes alias dereferencing, contained an alias plus one or more additional RDNs.
E_ALIAS_LOOP	During a whole–subtree search operation, an alias has been encountered which would lead to a loop (i.e., the alias points to an entry which is superior to entries which have already been evaluated in carring out the search).
E_ALIAS_PROBLEM	An alias has been encountered, but the entry to which it points does not exist.
E_ARG_BOUNDS	The argument does not comply with pragmatic constraints (defined locally or by functional standards).
E_ARG_SYNTAX	An operation argument either has incorrect ASN.1 encoding or it has correct ASN.1 encoding, but does not comply to the syntax as defined in the Directory Documents. Note: 1. Within BindArgument, additional elements are permitted, to allow future extensions, and do not create an error situation. 2. Errors within attribute values are not included in this codification (see E_ATT_SYNTAX).
E_ARG_VIOL	An operation argument has correct syntax, but it violates additional rules and constraints levied by the Directory Documents (e.g., use of a Priority integer value whose meaning is undefined). Note: 1. Within a Relative Distinguished Name, having two AVAs of the same attribute type is an error which is covered by E_DN, and not by E_ARG_VIOL. 2. Errors within attribute values are not included in this codification (see E_ATT_SYNTAX).
E_ATT_BOUNDS	An attribute value does not comply with bounds specified either by the Directory Documents or by functional standards.
E_ATT_OR_VALUE_EXISTS	Within an entry, an attribute or attribute value already exists, causing an error situation.

Table 11.10. Error Symptoms (Part 2 of 3)

Symptom	Descripton
E_ATT_SYNTAX	An attribute value either has incorrect ASN.1 encoding or it has correct ASN.1 encoding but does not comply witht the ASN.1 encoding defined by the attribute type.
E_ATT_VALUE	An attribute value, although of correct ASN.1 encoding, and conformant with the syntax defined for the attribute type, is not compliant with other rules (e.g., a non–ISO 3166 country name encoding).
E_ACCESS	The initiator has insufficient access rights to carry out this operation.
E_AUTHENTICATION	The authentication offered does not match that required by the object being authenticated.
E_BUSY	The DSA is unable to handle this operation at this time (but it may be able to do so after a short while).
E_CHAIN	The DSA needs to use chaining to carry out this operation, but is prohibited from doing so by Service Controls.
E_CREDENTIALS	The credentials offered do not match those of the object with which authentication is taking place.
E_DBE	An inconsistency has been detected in the DSA's data base, which may be localized to a particular entry or set of entries.
E_DIT_STRUCTURE	An attempt was made via an add operation to to place an entry in the DIB whose object class would violate the DIT structure rules.
E_DN	A DN contains an RDN with two AVAs of the same attribute type.
E_DSA	A DSA to which chaining is taking place is unable to respond.
E_ENTRY_EXISTS	An entry of the given name already exists, causing an error.
E_EXTENSION	A DSA was unable to satisfy a request because one or more critical extensions were not available.
E_ILLEGAL_ROOT_OBJ	Root's DN has been supplied as the object of a Read, Compare, AddEntry, RemoveEntry, ModifyEntry, ModifyRDN, or as the Base Object of a single level search.
E_ILLEGAL_ROOT_VAL	Root's DN has been supplied illegally as an attribute value (e.g., as an Aliased Object Name).
E_LOOP	A loop has been detected in the knowledge information within the system.
E_MATCH	The attribute specified does not support the required matching capability.
E_MISSING_AVA	When creating, or after modifying, an entry, an AVA in the entry's RDN is not represented within the entry's set of attributes.
E_MISSING_OBJECT_CLASS	When creating an entry, the entry does not possess an object class.
E_MULTI_DSA	The operation is an update operation which affects other DSAs.
E_NAMING_VIOLATION	The name of the new or modified entry is incompatible with its object class.
E_NON_LEAF_OPERATION	The operation being attempted is illegal except on a leaf.
E_NONNAMING_ATTRIBUTE	In either an add or ModifyRDN operation, an attribute is included in the last RDN that is not a valid naming attribute according to the DIT structure rules.

Table 11.10. Error Symptoms (Part 3 of 3)

Symptom	Descripton
E_NOT_SINGLE_VALUED	An attribute, registered as single–valued, has been found with more than one value.
E_NO_SUCH_ATT	The specified attribute has not been found.
E_NO_SUCH_OBJECT	The specified entry has not been found.
E_NO_SUCH_VALUE	The specified attribute value has not been found.
E_OBJECT_CLASS_MOD	An (illegal) attempt has been made to alter or remove an object class attribute.
E_OBJECT_CLASS_VIOL	There is a schema violation (e.g., missing mandatory attribute, or non–allowed attribute present).
E_REFERENCE	An erroneous reference has been detected (e.g., DSA cannot handle name even as far as the number of RDNs that have already been resolved).
E_SCOPE	No referrals were available within the requested scope.
E_SYSTEM_PERM	A serious and permanent software or system error has been detected which prevents completion of the operation.
E_SYSTEM_TEMP	A serious but temporary software or system error has been detected which prevents completion of the operation.
E_TIMEOUT	The operation has not completed within the allotted time.
E_UNABLE_TO_COMPLETE	The DSA is unable to complete this operation, or others like it (this applies particularly to search).
E_UNABLE_TO_PROCEED	The DSA cannot satisfy the operation after receiving it on the basis of a valid non–specific subordinate reference.
E_UNDEFINED_ATT	An unregistered attribute has been encountered.
E_UNSUPPORTED_OC	The object class of the entry is not supported as a valid object class for entries within this DSA.
E_VERSION	An unexpected version has been found in Bind.
E_ZERO_VALUES	An attribute has been found (e.g., as a result of a modify–entry operation) with no values.

Table 11.11. Error Situations

Situation	Description
BIND-LOCAL	A bind is being attempted; either the entry named is (or should be) within a local naming context, or name resolution is being carried out on the part of the name that is known locally.
BIND-REMOTE	A bind is being attempted, and the entry named is not within a local naming context; remote validation of credentials is being carried out.
NAME-RESOLUTION	Name resolution is being carried out.
ADD-ENTRY-NAME-RESOLUTION	During an add entry operation, name resolution has been successfully accomplished on the superior object, and is not being carried out to determine whether the new entry already exists.
ADD-ENTRY	The entry is being generated.
MODIFY-ENTRY	The entry is being modified.
MODIFY-RDN	The RDN is being modified.
REMOVE-ENTRY	The entry is being removed.
READ	The entry is being read.
COMPARE	A Compare operation is being carried out on the entry.
LIST	A List operation is being carried out on the entry.
SEARCH-FILTER	A Search operation is being carried out; the filter is being evaluated or acted upon.
SEARCH-ENTRY	A Search operation is being carried out; the required entry information is being evaluated or acted upon.
ABANDON	An Abandon operation is being carried out.
TRACE-EVALUATION	The trace element is being evaluated for loops.

Table 11.12. Notation Used to Describe Error Actions

Error Action Notation	Meaning
Rej	A reject operation is generated, with problem mistyped–argument.
Ab(<*qualifier*>)	Abandon Failed Error is generated. The *qualifier* may take on values codified as follows: CA – Cannot abandon NSO – No such operation TL – Too late
A(<*qualifier*>)	Attribute Error is generated. The *qualifier* may take on values codified as follows: AVE – Attribute or value already exists CV – Constraint violation IAS – Invalid attribute syntax IM – Inappropriate matching NSA – No such attribute UAT – Undefined attribute type
N(<*qualifier*>)	NameError is generated. The *qualifier* may take on values codified as follows: ADP – Alias dereferencing problem AP – Alias problem IAS – Invalid attribute syntax NSO – No such object
SC(<*qualifier*>)	Security Error is generated. The *qualifier* may take on values codified as follows: IA – Inappropriate authentication IAR – Insufficient access rights IC – Invalid credentials IS – Invalid signature NI – No information PR – Protection required
S(<*qualifier*>)	Service Error is generated. The *qualifier* may take on values codified as follows: ALE – Administrative limit exceeded B – Busy CR – Chaining required DE – Dit Error IR – Invalid reference LD – Loop detected OOS – Out of Scope TLE – Time limit exceeded UA – Unavailable UAP – Unable to proceed UCE – Unavailable critical extension UWP – Unwilling to perform
U(<*qualifier*>)	Update Error is generated. The *qualifier* may take on values codified as follows: AMD – Affects multiple DSA EAE – Entry already exist NAN – Not allowed on non–leaf NAR – Not allowed on RDN NV – Naming violation OCV – Object class violation OMP – Object class modification prohibited

Table 11.13. Error Actions (Part 1 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)					
	Bind-Local	Bind-Remote-Resolution	Name-Resolution	Add-Entry-Name-Resolution	Add-Entry	Modify-Entry
E_ACCESS			SC(IAR)(14)	SC(IAR)(14)	SC(IAR)(14)	SC(IAR)(14)
E_ADMIN_LIMIT	S(UA)	S(UA)	S(ALE)	S(ALE)	S(ALE)	S(ALE)
E_ALIAS_DEREF	S(IC)	S(IC)	N(ADP)			
E_ALIAS_LOOP						
E_ALIAS_PROBLEM	S(IC)	S(IC)	N(AP)			
E_ARG_BOUNDS	(8)	(7)	S(UWP)(12)	S(UWP)(12)	S(UWP)(12)	S(UWP)(12)
E_ARG_SYNTAX	(1)	(1)	Rej	Rej	Rej	Rej
E_ARG_VIOL	(1)	(1)	Rej	Rej	Rej	Rej
E_ATT_BOUNDS	SC(IC)	(7)	N(IAS)	N(IAS)	A(CV)	A(CV)
E_ATT_OR_VALUE_EXISTS					A(AVE)	A(AVE)
E_ATT_SYNTAX	SC(IC)	(7)	N(IAS)	N(IAS)	A(IAS)	A(IAS)
E_ATT_VALUE	SC(IC)	(7)	N(IAS)	N(IAS)	A(IAS)	A(IAS)
E_AUTHENTICATION	SC(IA)	SC(IA)				
E_BUSY	S(UA)	S(UA)	S(B)	S(B)	S(B)	S(B)
E_CHAIN				S(CR)		
E_CREDENTIALS	SC(IC)	SC(IC)				
E_DBE	S(UA)	S(UA)	S(DE)	S(DE)	S(DE)	S(DE)
E_DIT_STRUCTURE					U(NV)	
E_DN	SC(IC)	SC(IC)	N(NSO)	C(NV)		
E_DSA		S(UA)	S(UA)	S(UA)		
E_ENTRY_EXISTS				U(EAE)		
E_EXTENSION			S(UWP)	S(UCE)	S(UCE)	S(UCE)
E_ILLEGAL_ROOT_OBJ	SC(IC)	SC(IC)		N(NSO)	N(NSO)	N(NSO)
E_ILLEGAL_ROOT_VAL	SC(IC)	(7)	N(IAS)	N(IAS)	A(IAS)	A(IAS)
E_LOOP		S(UA)	S(LD)			

Table 11.13. Error Actions (Part 2 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)					
	Bind-Local	Bind-Remote-Resolution	Name-Resolution	Add-Entry-Name-Resolution	Add-Entry	Modify-Entry
E_MATCH	SC(IC)	SC(IC)	A(IM)	A(IM)		A(IM)
E_MISSING_AVA					U(NAR)	U(NAR)
E_MISSING_OBJECT_CLASS					U(OCV)	U(OMP)
E_MULTI_DSA				S(AMD)		
E_NAMING_VIOLATION				U(NV)		
E_NON_LEAF_OPERATION						
E_NONNAMING_ATTRIBUTE					U(NV)	
E_NOT_SINGLE_VALUED					A(CV)	A(CV)
E_NO_SUCH_ATT						A(NSA)
E_NO_SUCH_OBJECT	SC(IC)	SC(IC)	N(NSO)			
E_NO_SUCH_VALUE						A(NSA)
E_OBJECT_CLASS_MOD						U(OMP)
E_OBJECT_CLASS_VIOL					U(OCV)	U(OCV)
E_REFERENCE		S(UA)	S(IR)			
E_SCOPE			S(OOS)			
E_SYSTEM_PERM	S(UA)		S(UWP)	S(UWP)	S(UWP)	S(UWP)
E_SYSTEM_TEMP	S(UA)		S(UA)	S(UA)	S(UA)	S(UA)
E_TIMEOUT	S(UA)	(9)	S(TLE)	S(TLE)	S(TLE)	S(TLE)
E_UNABLE_TO_COMPLETE						
E_UNABLE_TO_PROCEED		(2)	(2)			
E_UNDEFINED_ATT	SC(IC)		(3)	U(NV)	A(UAT)	A(UAT)
E_UNSUPPORTED_OC					U(OCV)	
E_VERSION	S(UA)					
E_ZERO_VALUES					A(CV)	A(CV)

Table 11.13. Error Actions (Part 3 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)				
	Modify-RDN	Remove-Entry	Read	Compare	Trace-Evaluation
E_ACCESS	SC(IAR)(14)	SC(IAR)(14)	SC(IAR)(14)	SC(IAR)(14)	
E_ADMIN_LIMIT	S(ALE)		S(ALE)	S(ALE)	
E_ALIAS_DEREF					
E_ALIAS_LOOP					
E_ALIAS_PROBLEM					
E_ARG_BOUNDS	S(UWP)(12)		S(UWP)(12)	S(UWP)(12)	
E_ARG_SYNTAX	Rej	Rej	Rej	Rej	Rej
E_ARG_VIOL	Rej	Rej	Rej	Rej	Rej
E_ATT_BOUNDS	N(IAS)			A(CV)	(7)
E_ATT_OR_VALUE_EXISTS					
E_ATT_SYNTAX	N(IAS)			A(IAS)	(7)
E_ATT_VALUE	N(IAS)			A(IAS)	(7)
E_AUTHENTICATION					
E_BUSY	S(B)	S(B)	S(B)	S(B)	
E_CHAIN					
E_CREDENTIALS					
E_DBE	S(DE)	S(DE)	S(DE)	S(DE)	
E_DIT_STRUCTURE					
E_DN	A(CV)			A(IAS)	
E_DSA					
E_ENTRY_EXISTS	U(EAE)				
E_EXTENSION	S(UCE)	S(UCE)	S(UCE)	S(UCE)	
E_ILLEGAL_ROOT_OBJ	N(NSO)	N(NSO)	N(NSO)	N(NSO)	
E_ILLEGAL_ROOT_VAL	N(IAS)			A(IAS)	(7)
E_LOOP					

Table 11.13. Error Actions (Part 4 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)				
	Modify-RDN	Remove-Entry	Read	Compare	Trace-Evaluation
E_MATCH	A(IM)			A(IM)	(7)
E_MISSING_AVA					
E_MISSING_OBJECT_CLASS					
E_MULTI_DSA	S(AMD)	S(AMD)			
E_NAMING_VIOLATION	U(NV)				
E_NON_LEAF_OPERATION	U(NAN)	U(NAN)			
E_NONNAMING_ATTRIBUTE					
E_NOT_SINGLE_VALUED	A(CV)				
E_NO_SUCH_ATT			A(NSA)(4)	A(NSA)(4)	
E_NO_SUCH_OBJECT					
E_NO_SUCH_VALUE					
E_OBJECT_CLASS_MOD					
E_OBJECT_CLASS_VIOL	U(OCV)				
E_REFERENCE					
E_SCOPE					
E_SYSTEM_PERM	S(UWP)	S(UWP)	S(UWP)	S(UWP)	S(UWP)
E_SYSTEM_TEMP	S(UA)	S(UA)	S(UA)	S(UA)	S(UA)
E_TIMEOUT	S(TLE)	S(TLE)	S(TLE)	S(TLE)	
E_UNABLE_TO_COMPLETE					
E_UNABLE_TO_PROCEED					
E_UNDEFINED_ATT	A(UAT)		A(NSA)(4)	A(NSA)	(7)
E_UNSUPPORTED_OC					
E_VERSION					
E_ZERO_VALUES					(11)

Table 11.13. Error Actions (Part 5 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)			
	List (Filter)	Search (Filter)	Search Entry	Abandon
E_ACCESS	SC(IAR)(14)	SC(IAR)(14)	SC(IAR)(14)	
E_ADMIN_LIMIT	S(ALE)(13)	S(ALE)(13)	S(ALE)(13)	
E_ALIAS_DEREF		(5)		
E_ALIAS_LOOP		(5)		
E_ALIAS_PROBLEM		(5)		
E_ARG_BOUNDS	S(UWP)(12)	S(UWP)(12)	S(UWP)(12)	
E_ARG_SYNTAX	Rej	Rej	Rej	Rej
E_ARG_VIOL	Rej	Rej	Rej	
E_ATT_BOUNDS		A(CV)		
E_ATT_OR_VALUE_EXISTS				
E_ATT_SYNTAX		A(IAS)		
E_ATT_VALUE		A(IAS)		
E_AUTHENTICATION				
E_BUSY	S(B)	S(B)	S(B)	
E_CHAIN				
E_CREDENTIALS				
E_DBE	S(DE)	S(DE)	S(DE)	
E_DIT_STRUCTURE				
E_DN		A(IAS)		
E_DSA		(5)		
E_ENTRY_EXISTS				
E_EXTENSION	S(UCE)(13)	S(UCE)(13)	S(UCE)(13)	
E_ILLEGAL_ROOT_OBJ		(10)		
E_ILLEGAL_ROOT_VAL		A(IAS)		
E_LOOP		(5)		

Table 11.13. Error Actions (Part 6 of 6)

Symptom (See Table 11.10)	Situation (See Table 11.11)			
	List (Filter)	Search (Filter)	Search Entry	Abandon
E_MATCH		A(IM)		
E_MISSING_AVA				
E_MISSING_OBJECT_CLASS				
E_MULTI_DSA				
E_NAMING_VIOLATION				
E_NON_LEAF_OPERATION				
E_NONNAMING_ATTRIBUTE				
E_NOT_SINGLE_VALUED				
E_NO_SUCH_ATT				
E_NO_SUCH_OBJECT				
E_NO_SUCH_VALUE				
E_OBJECT_CLASS_MOD				
E_OBJECT_CLASS_VIOL				
E_REFERENCE				
E_SCOPE				
E_SYSTEM_PERM	S(UWP)	S(UWP)	S(UWP)	Ab(CA)
E_SYSTEM_TEMP	S(UA)	S(UA)	S(UA)	Ab(CA)
E_TIMEOUT	S(TLE)(13)	S(TLE)(13)	S(TLE)(13)	
E_UNABLE_TO_COMPLETE	S(B)	S(B)	S(B)	Ab(CA)
E_UNABLE_TO_PROCEED				
E_UNDEFINED_ATT		(6)	(6)	
E_UNSUPPORTED_OC				
E_VERSION				
E_ZERO_VALUES				

Notes for table 11.13

1. Use A-U-ABORT. Note, however, that extra elements are permitted here.

2. An "unable-to-proceed" error becomes SC(IC) for bind and N(NSO) for operations if no DSA contacted can located the object.

3. An undefined attributed encountered during name resolution is only an error – N(NSO) – if the entry is identified as local. See also Note 10 below.

4. The A(NSA) condition is reserved in the case of "read" for the situation when no attribute of the specific list provided can be returned (for reasons that include security errors).

5. Any failure to propagate a search causes abandonment of that part of the search.

6. Undefined attributes are regarded as not matched or found, but cause no errors in search.

7. This error, if detected, should be ignored; processing continues.

8. This error would occur as a result of a bind argument with a name containing too many RDNs for the DSA. Use either S(UA) or S(IC).

9. DSAs should use the time-limit service control with local timeout to limit the remote validation of credentials; if the operation fails as a result, S(UA) is used.

10. For a single-entry search, N(NSO) may be used.

11. Either the whole attribute should be removed, or the deleteOldRDNflag should be ignored.

12. Wherever S(UWP) appears in the above tables beside E_ARG_BOUNDS, a ROSE "Rej" is also admissible.

13. The error is returned when there are no partial results, otherwise a partialOutcome-Qualifier with the appropriate limitProblem is returned (cf Directory Documents, Part 3, item g of clause 12.8.2, and Part 3, clause 10.1.3.3.1).

14. In every case where a security error occurs, except in bind, SC(NI) may be used in place of the specified problem, to support a Security Policy which states that no information on the problem may be divulged. In the case of the bind, SC(NI) is not available.

---

12. SECURITY

The Security Architecture specified in ISO 7498/Part 2 - Security Architecture (as presented in ISO/TC 97/SC 21/N1528) shall be used as a basis for further work in the Special Interest Group on Security.

The security services that are to be implemented first shall include confidentiality, integrity, authentication and access control. Non-repudiation of the source shall also be included for consideration for implementation. These services are defined and discussed in more detail in ISO 7498/Part 2 - Security Architecture.

12.1 Definitions

The following definitions, based on the definitions in ISO 7498/Part 2, are to be used when interpreting chapter 12.

Access Control:	The provision of a security system that establishes and enforces which users or processes can get access to what data or processing facilities.
Authentication Information:	Information used to establish the validity of a claimed identity.
Authorization:	The granting of access rights.
Confidentiality:	A security service that protects data from unauthorized disclosure.
Connection:	A state of communication that exists between two communicating entities by establishing an association between them, providing one or more data paths between them allowing sequential transfers of data, and then terminating the association.
Connectionless:	A state of communication that provides transfer of data from one entity to another without a preestablished association.
Data Integrity:	The property that data has not been altered or destroyed in an unauthorized manner.

Data Origin Authentication:	The corroboration that the source of data received is as claimed.
Digital Signature:	Data that allows a recipient of information to verify the source and integrity of the information.
Peer-entity Authentication:	The corroboration that peer entities in an association are as claimed.
Repudiation:	Denial by one or both of the entities of an association of having participated in all or part of the association or communication of the association.
Selective Field Protection:	The protection of specified fields of data in a communication.
Traffic Analysis:	The inference of information from observation of traffic flow in communications (presence, absence, amount, direction and frequency).
Traffic Flow Confidentiality:	A confidentiality service to protect against traffic analysis.

12.2 Matrix of Security Services and OSI Layers

The following matrix shows the layers of the OSI architecture at which certain security services are considered to be desirable. The entries in the matrix are "H" for high level of desirability, "M" for medium desirability, and "L" for low level of desirability. No entry in the matrix means that the service is not considered desirable. This matrix was produced from a similar matrix in ISO 7498/Part 2 which showed the layers of the architecture that could be used to provide the security service. The level of desirability was established by the members of the Special Interest Group in Security of the OSI Implementors Workshop.

Note: The Matrix is a consensus of the opinions of the members as to where selected security services should be placed. It should not be

considered restrictive and interpreted as meaning that the security services cannot be placed elsewhere in the OSI architecture or have other implementation priorities. This will depend upon the differing security needs of specific applications. Also, it should not be considered complete in that other security services may exist that should be incorporated in the architecture.

Table 12.1. OSI Layers Desirable for Placing Security

SERVICE	1	2	3	4	5	6	7
1.(a) Peer entity authentication			L	H			H
(b) Data origin authentication			L	L			H
2. Access Control Service			M	M			H
3.(a) Connection confidentiality	L	L	L	H		H	H
(b) Connectionless confidentiality		L	H	L		H	H
(c) Selective field confidentiality						H	H
(d) Traffic flow confidentiality	M		L				L
4.(a) Connection integrity with recovery				H			L
(b) Connection integrity without recovery			N	N			N
(c) Selective field connection integrity							N
(d) Connectionless integrity			H	L			L
(e) Selective field connectionless integrity							H
5.(a) Non-repudiation: originator							L
(b) Non-repudiation: receiver							L

Implementation priority:
H (high)
M (medium)
L (Low)
N (No Priority)

Table 1. ISO 7498/Part 2: Security Addendum -- NIST/OSI Workshop Summary Of SIG-SEC Discussions of Security Service Placement, May, 1987

Notes: The following notes are for explanation of the above matrix and comments.

A security system should be considered to be an integrated set of

security services that are placed at selected OSI layers. The services should be selected based on a risk analysis for the computer system being protected. Security mechanisms must be then chosen that will provide the security services and incorporated in the software and hardware of the computer system and controlled by the OSI software and hardware at the selected layer(s).

For example, authentication, access control, confidentiality and integrity are selected as the major security goals for an OSI system. A connection oriented transport protocol is being implemented. An example of the use of the Matrix could be in an electronic mail system, to illustrate this the following specific services and layers were chosen:

Peer entity authentication: Layer 4
Data origin authentication: Layer 7

Access Control: Layer 7

Connection confidentiality: Layer 4
Selective field confidentiality: Layer 7

Connection integrity with recovery: Layer 4
Connectionless integrity: Layer 7

The layer 7 services were chosen to support the mail system that would protect the selective paragraphs of an electronic message as directed by the user. A mail system is considered connectionless. Access control is a function of only layer 7.

The layer 4 services were chosen to provide a reliable transport service from the sender to the intended receive of the electronic message. A full connection integrity and confidentiality service with peer entity authentication will assure that all information gets to the receiver correctly and confidentially.

Note: The security protocols and mechanisms that provide these services are beyond the scope of this chapter at this time. The mechanisms and standards for their interoperability are presently being defined and will be added to this chapter as they become available.

13. FUTURE SECURITY

Editor's Note: This section is reserved for future stable Security Architecture Agreements. These agreements may be found in the aligned section of the Working Implementation Agreements Document. When these agreements become stable, they will be moved into this section 13 of this document. Consult section 13 of the Working Agreements Document for current information.

14. ISO VIRTUAL TERMINAL PROTOCOL

14.1 INTRODUCTION

The NIST/OSI Workshop Virtual Terminal (VT) SIG is making implementation agreements for the OSI Basic Class VT Service and Protocol, ISO 9040 and ISO 9041.

These implementation agreements fall into the following categories.

- Functionality to be implemented, i.e.,, functional units, etc.
- Identification and specification of VT profiles to be supported by conforming implementations.
- Agreements with regard to implementation issues not specified in ISO 9040 and ISO 9041.
- Resolution of problems with ISO 9040 and ISO 9041 identified during implementation.
- Statement of requirements to meet conformance to these agreements.

14.2 SCOPE AND FIELD OF APPLICATION

14.2.1 Phase Ia Agreements

The Telnet profile is intended to support the following usage:

- a simple line at a time or character at a time dialogue, and
- an application level gateway supporting Internet Telnet and ISO VTP interoperation.

The Transparent profile supports the exchange of uninterpreted sequences of characters. This includes support of VT-users who wish to control terminals directly through the use of embedded control characters and escape sequences.

14.2.2 Phase Ib Agreements

The Forms profile is intended to support forms-based applications with local entry and validation of data by the terminal system.

This profile is now aligned with the EWOS VT EG Functional Standard.

14.2.3 Phase II Agreements

The X.3 profile supports functionality similar to the CCITT recommendations and could be used to implement an X.3 to ISO-VT gateway.

See Working Agreements regarding Page profiles.

14.3 STATUS

14.3.1 Status of phase Ia

Phase Ia of the VT Agreements was stabilized May 5, 1988. This phase covers the Telnet and Transparent profiles. No future enhancements will be made to this phase.

14.3.2 Status of phase Ib

Phase Ib of the VT Agreements was first stabilized December 16, 1988. This phase covers the Forms profile. Alignment with EWOS required substantial modifications which were ratified September 15, 1989.

14.3.3 Status of phase II

Phase II is still in progress and includes the remaining profile work for Scroll, Page (S-mode) and Page (A-mode) profiles.

The X.3 profile of phase II was stabilized December 15, 1989.

14.4 ERRATA

Editor's Note: "Defect Report" material may be included here, including versions of implementor agrements to which it applies.

TECHNICAL

06/90-1 Forms Profile. The "FEICO Update Syntax" ASN.1 comment which follows the definition of the PriValue type was corrected to support multi-octet repertoires.

06/90-2 Forms Profile. The descriptive text for the Field Entry Instruction Violation FEE was corrected to

indicate that both an entry-control index and a FEPR index are required to identify the FEPR concerned.

06/90-3 Forms Profile. The descriptive text and update syntax for the Violation FEC were corrected to indicate that both a FEICO-name and an index are required to identify a FEIR.

06/90-4 Forms Profile. The update syntax for the writeString FER was corrected to align with the descriptive text for this FER.

06/90-5 Forms Profile. The descriptive text for the repertoire assignment profile argument was corrected to properly identify the default value as the GL set ISO 2375 Reg. No. 6 (ASCII).

06/90-6 Forms Profile. The concept of a "current keystroke" was inserted into the definition of the FEICO to remove ambiguity in the use of the ST and UT COs. Various FEEs, FECs and FERs were redefined.

ALIGNMENT

06/90-7 Forms Profile. A definitive note was added to define how the host is notified of the current entry location when data entry terminates and the VTE-parameter access-outside-fields has the value "allowed."

06/90-8 Forms Profile. Three font-assignment profile arguments were added to accomodate INTAP requirements.

09/90-1 Forms Profile. The emphasis subattribute "h" was added with values "F" (Framed) and "C" (Encircled).

09/90-2 Telnet Profile. Four editorial comments were incorporated to align with the corresponding EWOS Functional Standard.

EDITORIAL

06/90-9 Forms Profile. Two definitive notes were added to clarify the secondary attributes comparison mechanisms for the FEIs and FECs that test equality of characters.

06/90-10 Forms Profile. A definitive note was added to clarify the effect of associating multiple Character-oriented FEIs of the same type with the same field.

06/90-11 Forms Profile. An introductory paragraph in the section "Field Entry Condition Definitions" was rewritten for clarification.

06/90-12 Forms Profile. The descriptive text for the Write String field entry reaction was modified to indicate precisely how and where the associated string is to be written.

09/90-3 X3 Profile. The reference to COs P3 and P4 contained in comments relating to DEVICE-1 were corrected to reference elements 3 and 4 of the PAD CO.

14.5 CONFORMANCE

Conformant VT implementations are required to support the ISO 9041 Clause 13 requirements plus the additional conformance requirements identified below. .

Figure 14.1 shows conformance status for VT facilities which are optional in the ISO VT standard. The terms used in the figure are defined as indicated below.

- "Mandatory" indicated the facility must be provided by all implementations which conform to these agreements.
- "Optional" indicates that the VT facility is not required to meet minimum conformance requirements but has been identified as providing additional useful capabilities.
- "Profile Dependent" indicates that the requirement for the facility, if any, is included in the profile definitions.
- "Not Addressed" indicates that the VT facility is outside the scope of these agreements.

Conformance Status	Mandatory	Optional	Profile Dependent	Not Addressed
Switch Profile **		X		
Multiple Interaction Negotiation **				X
Negotiated Release **				X
Urgent Data **		X		
Break **	X			
Delivery Control*			X	
Enhanced Access Rules **			X	
Structured COs **			X	
Blocks **				X
Fields **			X	
RIOs **			X	
S-mode			X	
A-mode			X	
Mode Switching Capability		X		

Figure 14.1. Conformance Status for VT Facilities.

*It is not anticipated that new profiles will use quarantined delivery control.

**Functional Units.

For each mode of operation (A-mode and S-mode) which is implemented, the default profile for that mode as defined in ISO 9040 must be supported. Implementations that support A-mode must support the A-mode default profile and at least one additional Workshop approved A-mode profile. The Transparent profile does not count as an additional A-mode profile. Implementations that support S-mode must support the

S-mode default profile and at least one additional Workshop approved S-mode profile.

For each profile implemented, VTE parameter ranges or values specified in the Workshop-agreed profile and associated notes must be supported.

14.6 PROTOCOL

14.6.1 Protocol Elements

All protocol elements required by the ISO 9040 VT kernel and Break functional units are selected.

All protocol elements required by the Switch Profile functional unit are selected if this functional unit is used. See figure 14.1.

All protocol elements required by the Urgent Data functional unit are selected if this functional unit is used. See figure 14.1.

14.6.2 Mapping of Protocol Elements

Mapping of protocol elements on to ACSE or Presentation Services is as defined in ISO 9041.

14.6.3 Protocol Data Unit Structure

Protocol data unit structure is as defined in ISO 9041.

14.7 OIW Registered Control Objects

The following Control Objects are used by more than one profile. Some of the CO parameters are left with undefined values that must be assigned by the profile in which the Control Object is used.

14.7.1 Sequenced Application (SA)

This is a Control object used to convey signals from the application to the terminal in sequence with other updates.

14.7.1.1 Entry Number

To be supplied by Registration Authority.

14.7.1.2 Name of Sponsoring Body

OSI Implementors' Workshop (OIW), VTSIG.

14.7.1.3 Date

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

14.7.1.4 Identifier

```
oiw-vt-co-misc-sa OBJECT IDENTIFIER ::=
     {oiw-vt-co-misc sa(0)}
```

14.7.1.5 Descriptor Value

"OIW VT CO for conveying Sequenced Application Signals"

14.7.1.6 CO Parameters

CO-structure	1
CO-priority	"normal"
CO-category	"symbolic"
CO-size	11

14.7.1.7 CO Values and Semantics

The following table lists the allowed symbolic values together with the integers used to reference these values in the ASN.1 update syntax of ISO 9041:

audible_alarm	0
newlines_enabled	1
newlines_disabled	2
restore	3
visual_alarm	4
keypad_enabled	5
keypad_disabled	6
keyboard_locked	7
keyboard_unlocked	8
device_disconnect	9
break_signal	10

The semantics of each value must be specified in the VTE profile which references this CO.

14.7.1.8 Additional Information

None

14.7.1.9 Usage

Defined in profile.

14.7.2 Unsequenced Application (UA)

This is a Control object used to convey urgent signals from the application to the terminal.

14.7.2.1 Entry Number

To be supplied by Registration Authority.

14.7.2.2 Name of Sponsoring Body

OSI Implementors' Workshop (OIW), VTSIG.

14.7.2.3 Date

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

14.7.2.4 Identifier

```
oiw-vt-co-misc-ua OBJECT IDENTIFIER::=
     {oiw-vt-co-misc ua(1)}
```

14.7.2.5 Descriptor Value

"OIW VT CO for conveying Unsequenced Application Signals"

14.7.2.6 CO Parameters

CO-structure	1
CO-priority	"urgent"
CO-category	"symbolic"
CO-size	11

14.7.2.7 CO Values and Semantics

Same as in SA.

14.7.2.8 Additional Information

None.

14.7.2.9 Usage

Defined in profile.

14.7.3 Sequenced Terminal (ST)

A keyboard can generate many signals that may be given special meaning to the application. This CO is general enough to convey any keyboard event.

14.7.3.1 Entry Number

To be supplied by Registration Authority.

14.7.3.2 Name of Sponsoring Body

OSI Implementors Workshop (OIW), VTSIG.

14.7.3.3 Date

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

14.7.3.4 Identifier

```
oiw-vt-co-misc-st OBJECT IDENTIFIER ::=
     {oiw-vt-co-misc st(2)}
```

14.7.3.5 Descriptor Value

"OIW VT CO for conveying Sequenced Terminal Signals"

14.7.3.6 CO Parameters

CO-structure	1
CO-priority	"normal"
CO-category	"integer"
CO-size	65535

14.7.3.7 CO Values and Semantics

The values of the CO are composite, with values from table below giving meaning to the values in the hex range 00-FF when added to them.

hex value	meaning
100	special key - labeled (see list)
200	function key depressed
400	control key depressed
800	shift key depressed
1000	alt key depressed

The special key and the function key are mutually exclusive. If neither the function keys nor the special keys are pressed, then the value in the hex range 00-FF will be that of the normal, unshifted code combination generated by the alpha-numeric key. Values in the hex range 00-FF are not valid values for the data element of this Control Object.

The control, shift, and alt keys may appear in any combination with the special or function keys.

The shift key must occur in combination with at least one of the other keys in the above table to cause the value to fall outside the repertoire of the display object.

When the special key is depressed, the value of the CO content will be as given in the ASN.1 module below for the

value in the hex range of 00-FF. Otherwise, the value will be defined to be the IA5 value associated with the key.

```
STCO DEFINITIONS ::= BEGIN

Key ::= INTEGER {

break           (0),   bell              (1),   backSpace        (2),
tab             (3),   backTab           (4),   lineFeed         (5),
carReturn       (6),   cancel            (7),   substitute       (8),
escape          (9),   plus             (10),   minus           (11),
multiply       (12),   divide           (13),   leftArrow       (14),
rightArrow     (15),   upArrow          (16),   downArrow       (17),
insert         (18),   delete           (19),   insertLine      (20),
deleteLine     (21),   home             (22),   end             (23),
pageUp         (24),   pageDown         (25),   pa1             (26),
pa2            (27),   pa3              (28),   help            (29),
statusProcess  (30),   interruptProcess (31),   terminateProcess (32),
abortOutput    (33),   formFeed         (34),   clear           (35),
print          (36),   refresh          (37),   systemRequest   (38),
endOfRecord    (39),   endOfFile        (40),   suspendProcess  (41)

-- Names for combination keystrokes are formed by converting the
-- initial letter to upper case and prefixing with 'ctrl', 'shift' or
-- 'alt', which adds 1024, 2048 or 4096 respectively to the value.
-- These prefixes may be used in combination with one another by a
-- repetition of this conversion process, provided that they appear
-- from left to right in the order 'ctrl', 'shift', 'alt'.  ASN.1
-- formally does not allow such descriptive additions but it would be
-- very lengthy to write them all in full  --    }

END   *(STCO DEFINITIONS)*
```

VTE profile definitions may refer to this module for convenience in describing semantics.

14.7.3.8 Additional Information

None.

14.7.3.9 Usage

Defined in profile.

14.7.4 Unsequenced Terminal (UT)

Keyboard events may need to be conveyed urgently, out of sequence with normal updates. This CO is used to signal such events from the terminal to the application.

14.7.4.1 Entry Number

To be supplied by the Registration Authority.

14.7.4.2 Name of Sponsoring Body

OSI Implementors Workshop (OIW), VTSIG

14.7.4.3 Date

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

14.7.4.4 Identifier

```
oiw-vt-co-misc-ut OBJECT IDENTIFIER ::=
     {oiw-vt-co-misc ut(3)}
```

14.7.4.5 Descriptor Value

"OIW VT CO for conveying Unsequenced Terminal Signals"

14.7.4.6 CO Parameters

CO-structure	1
CO-priority	"urgent"
CO-category	"integer"
CO-size	65535

14.7.4.7 CO Values and Semantics

Same as in ST.

14.7.4.8 Additional Information

None.

14.7.4.9 Usage

Defined in profile.

14.8 OIW Defined Profiles

These profiles are defined using the conventions specified in Annex A of ISO 9040.

14.8.1 Telnet Profile

OIW VTE-Profile Telnet-1988 (r1, r2)

14.8.1.1 Introduction

This profile provides support for TELNET-like operation for users of the ISO Virtual Terminal Service. It is based on the IS version of ISO 9040 and ISO 9041.

14.8.1.2 Association Requirements

14.8.1.2.1 Functional Units

The Urgent Data Functional Unit is optional, but should be used whenever available.

14.8.1.2.2 Mode

This is an A-mode profile.

14.8.1.3 Profile Body

```
Display-objects = *(double occurrence)*
{
     {
     display-object-name = D, *(DISPLAY)*
     do-access           = "WACA,"
     dimensions          = "two,"
```

```
            x-dimension    =
            {
                 x-bound         = "unbounded,"
                 x-addressing    = "no constraint,"
                 x-absolute      = "no,"
                 x-window        = profile-argument-r1
            },
            y-dimension    =
            {
                 y-bound         = "unbounded,"
                 y-addressing    = "higher only,"
                 y-absolute      = "no,"
                 y-window        = 1
            },
       erasure-capability    = "yes,"
       repertoire-capability = 2,
       repertoire-assignment = profile-argument-r2,
       repertoire-assignment = <ESC> 2/5 2/15 4/2
       },
       {
       display-object-name = K, *(KEYBOARD)*
       do-access           = "WACI,"
       dimensions          = "two,"
            x-dimension    =
            {
                 x-bound         = "unbounded,"
                 x-addressing    = "no constraint,"
                 x-absolute      = "no,"
                 x-window        = profile-argument-r1
            },
            y-dimension    =
            {
                 y-bound         = "unbounded,"
                 y-addressing    = "higher only,"
                 y-absolute      = "no,"
                 y-window        = 1
            },
       erasure-capability    = "yes,"
       repertoire-capability = 2,
       repertoire-assignment = profile-argument-r2,
       repertoire-assignment = <ESC> 2/5 2/15 4/2
       }
  },

  Control-objects = *(multiple occurrence)*
  {
       { *(SYNCHRONIZE)*
            CO-name         = SY,
            CO-access       = "NSAC,"
            CO-category     = "symbolic,"
```

```
        CO-size         = 1,
        CO-priority     = "urgent"
    },
    { *(DISPLAY-SIGNAL)*
        CO-name         = DI,
        CO-access       = "WACA,"
        CO-category     = "boolean,"
        CO-size         = 5,
        CO-priority     = "normal,"
        CO-trigger      = "selected"
    },
    { *(KEYBOARD-SIGNAL)*
        CO-name         = KB,
        CO-access       = "WACI,"
        CO-category     = "boolean,"
        CO-size         = 5,
        CO-priority     = "normal,"
        CO-trigger      = "selected"
    },
    { *(NEGOTIATION BY INITIATOR)*
        CO-name         = NI,
        CO-access       = "WACI,"
        CO-category     = "boolean,"
        CO-size         = 4,
        CO-priority     = "normal,"
        CO-trigger      = "selected"
    },
    { *(NEGOTIATION BY ACCEPTOR)*
        CO-name         = NA,
        CO-access       = "WACA,"
        CO-category     = "boolean,"
        CO-size         = 4,
        CO-priority     = "normal,"
        CO-trigger      = "selected"
    },
    { *(GO-AHEAD)*
        CO-name         = GA,
        CO-access       = "NSAC,"
        CO-category     = "boolean,"
        CO-size         = 1,
        CO-priority     = "normal,"
        CO-trigger      = "selected"
    }
},
```

```
Device-objects = *(double occurrence)*
{
     {
     device-name = DISPLAY-DEVICE,
     device-display-object = D,
     device-default-CO-initial-value = 1."true,"*("on")*
     device-minimum-X-array-length = 1,*(no constraint)*
     device-minimum-Y-array-length = 1,*(no constraint)*
     device-control-object = SY,
     device-control-object = NA,
     device-control-object = DI,
     device-control-object = GA,
          *(SYNC,NEGOTIATE-ACCEPTOR,DISPLAY-SIGNAL,
               GO-AHEAD)*
     device-default-CO-access = "WACA,"
     device-default-CO-priority = "normal"
     *(other device object parameters assume corresponding
     DO values)*
     },
     {
     device-name = KEYBOARD-DEVICE,
     device-display-object = K,
     device-default-CO-access = "WACI,"
     device-default-CO-priority = "normal,"
     device-default-CO-initial-value = 1."true,"*("on")*
     device-minimum-X-array-length = 1,*(no constraint)*
     device-minimum-Y-array-length = 1,*(no constraint)*
     device-control-object = SY,
     device-control-object = NI,
     device-control-object = KB,
     device-control-object = GA,
          *(SYNC,NEGOTIATE-INITIATOR,KEYBOARD-SIGNAL,
               GO-AHEAD)*
     *(other device object parameters assume corresponding
     DO values)*
     }
},
Type of delivery control = "simple-delivery-control."
```

14.8.1.4 Profile Arguments

r1 - is used to represent the line length as the value of VTE parameter x-window for both display objects. This argument is mandatory and takes a nonnegative integer value. This argument is identified by the identifier for x-window for display object D.

r2 - is used to designate the default repertoire for both display objects. This argument is optional, if not

present the full US ASCII set is the default. This argument is identified by the identifier for repertoire assignment for the display object D.

14.8.1.5 Profile dependent Control Object Information

This profile does not reference any Control Objects which are not defined within this profile.

14.8.1.6 Profile Notes

14.8.1.6.1 Definitive Notes

1. Booleans in the KB and DI control objects are used in this profile to correspond to TELNET commands as follows:

Control Object	Boolean	TELNET
DI/KB	1	IP
DI/KB	2	AO
DI/KB	3	AYT
DI/KB	4	DM
DI/KB	5	BREAK

   The equivalent of a TELNET command is achieved by selecting the boolean that corresponds to the desired TELNET command. Selecting a boolean in the DI or KB control object means setting the value of the desired boolean to "true." The usage of the mask element of the boolean update is as specified in ISO 9041.

2. The equivalent of a TELNET SYNCH command is achieved by updating the SY control object with the single symbolic value of "SYNCH" (which is mapped onto the integer value 1), and immediately updating the DI (or KB) control object selecting the DM boolean. IP, AO, AYT, or BREAK commands may be accompanied by a SYNCH command by updating the SY control object and then updating the DI or KB control object selecting both the DM and the other desired boolean. When an update to the SY control object is received subsequent display object updates are discarded until an update to the DI or KB control object is received selecting the DM bit. If a VT-BREAK is received after an SY

CO update has been received and prior to the corresponding DI or KB CO update selecting the DM boolean, the discarding of updates is terminated. This is necessary because the VT-BREAK may have caused the DI or KB CO update to be purged.

3. The NI and NA control objects are used to emulate the TELNET option negotiation facility. The facility is symmetric, allowing either party to open negotiation for a change of mode, and every negotiation must be accepted or rejected by the opposite party. The rules for negotiation for each of the option controls are as stated in RFC 854 and as given below.

   a. Only open negotiation for a change from the current state.

   b. Only acknowledge negotiation for a change from the current state.

   c. Do not send any object updates with a negotiation outstanding except an update to the NI (or NA) control object to acknowledge negotiation.

   For full symmetry, both the NI and NA control objects have the same value definition and consist of 4 booleans with the semantics given below.

BIT	Option	Value
1	Remote Echo	"false" Echo is local; "true" Echo is remote
2	Suppress Go Ahead	"false" Go Ahead; "true" Suppress Go Ahead
3	Binary WACA	"true" use binary WACA; "false" use default or negotiated repertoire for WACA display object
4	Binary WACI	"true" use binary WACI; "false" use default or negotiated repertoire for WACI display object

   Booleans 3 and 4 control the use of the Transparent character set for the D and K display

objects respectively. A value of "true" indicates the use of the binary repertoire; "false" indicates the use of the negotiated repertoire. When a party wants to change a repertoire assignment, it must complete a successful TELNET negotiation to change the option control. Then the party with the access rights to the display object in question is required to perform the corresponding secondary attribute modal update.

4. The TELNET EC (erase character) command will be mapped to a pointer relative (x:= x-1) update and an erase current update. The TELNET EL (erase line) command should will be mapped to an erase-full-x-array update (an erase operation where the extent is defined as <"start-x,"(Yc,Xc-1)> and a pointer update to set x = 1. These X dimension updates are the only times when backward explicit addressing is permitted.

5. The X address of the pointer can be moved forward only by implicit pointer addressing. Addressing of the Y dimension is limited to the next X-array update operation.

6. The VT next X-array update operation will be sent in place of the TELNET NVT "CR,LF" sequence.

7. While the "binary" repertoire is being used no mapping to pointer addressing or erase operations will be done.

8. The repertoire designation "7-bit ASCII (G0+C0)" refers to the repertoire invoked by ISO 2022 defined character set designating escape sequences <ESC> 2/8 4/2, "void," <ESC> 2/1 4/0. The repertoire designation "7-bit ASCII (G0 only)" refers to the repertoire invoked by the ISO 2022 defined character set designating escape sequence <ESC> 2/8 4/2. The designation "binary" refers to the "Virtual Terminal Service Transparent Set" registered in the International Register under ISO 2375 register value 125 and invoked by the escape sequence <ESC> 2/5 2/15 4/2.

9. No termination event list is specified so that data buffering and delivery can be controlled according to context. If local echoing is enabled, the local newline or enter event shall trigger a VT-DELIVER request. With remote echo a

timeout or buffer length may be used to trigger a VT-DELIVER request. This buffer length may be 1.

14.8.1.6.2 Informative Notes

1. Users of this profile should refer to the TELNET specification (MIL-STD-1782) and RFCs 854 and 855 for semantics of the TELNET commands. These documents can be obtained by contacting SRI International, DDN Network Information Center, 333 Ravenswood Ave., Menlo Park, CA 94025, (415) 859-3695.

2. An update to the GA control object is equivalent to the TELNET Go Ahead command.

3. If the "go ahead" facility has been negotiated then following a VT-BREAK, only the association acceptor has the right to send data. In the event of VT-BREAK the echo control objects are reinitialized to "false," meaning local echo. If remote echo is desired it must be re-negotiated following VT-BREAK.

4. Negotiation of TELNET options other than echo, transmit binary, and SUPPRESS GO AHEAD is not supported by this profile. Negotiations for these three options can take place at any time during a session.

14.8.1.7 Specific Conformance Requirements

The following character sets are required:

- The G0 character set for U.S. 7-bit ASCII (values 32-126),
- The full U.S. 7-bit ASCII (values 0-127), and
- The transparent character set, see Definitive Note 8 in section 14.8.1.6.1.

14.8.2 Transparent Profile

14.8.2.1 Introduction

This profile is intended to provide a transparent mode of operation which allows VT-users to exchange transparently

uninterpreted sequences of characters but with the added benefit of delivery control to enable the VT-users to determine when the character sequences are to be delivered. This profile may be used when VT-users wish to control terminals directly through the use of embedded control characters.

14.8.2.2 Association Requirements

14.8.2.2.1 Functional Units

No additional functional units are required by this profile.

14.8.2.2.2 Mode

This is an A-mode profile.

14.8.2.3 Profile Body

```
Display-objects *(double occurrence)* =
(
     (
     display-object-name = D1,
     do-access           = "WACA,"
     dimensions          = "one,"
          x-dimensions   =
          (
               x-addressing = "not-permitted"
          ),
     repertoire-assignment = profile-argument-r1
     ),
     (
     display-object-name = D2,
     do-access           = "WACI,"
     dimensions          = "one,"
          x-dimension    =
          (
               x-addressing = "not-permitted"
          ),
     repertoire-assignment = profile-argument-r1
     )
),
type-of-delivery-control = "simple-delivery-control."
```

14.8.2.4 Profile Arguments

r1 - is optional and enables negotiation of a value for the VTE-parameter repertoire-assignment for the two display objects (which always have the same value of repertoire assignment when the profile is called). The default value of this argument is the "Virtual Terminal Transparent Set" registered in the International Register under ISO 2375 register value 125, invoked by the escape sequence <ESC> 2/5 2/15 4/2. This argument is identified by the identifier for repertoire-assignment for display object D1.

14.8.2.5 Profile dependent Control Object Information

This profile does not reference any Control Objects.

14.8.2.6 Profile Notes

1. This profile is intended primarily for applications requiring a simultaneous two way exchange of sequences of uninterpreted characters. The semantics usually associated with the display object are not used; for the purposes of this profile, the primary attributes of the character-box graphic elements are actually octets which are passed directly to the real device. There is no relationship between the elements of the X-array and the character boxes of the real device; the secondary attributes of the display object are not utilized. The only operation on the display object which must be supported is the text operation. An alternative repertoire may be selected.

14.8.2.7 Specific Conformance Requirements

Support for the default (transparent) character set is required. It is strongly recommended that the profile argument not be used.

14.8.3 Forms Profile

OIW VTE-Profile Forms-1989 (r1,r2, . . . r28)

14.8.3.1 Introduction

This S-mode VTE-profile is intended for supporting the use of forms based, field oriented data entry applications between a terminal and a host system.

It provides facilities for:

- defining and using screen forms,
- defining field validation and field entry rules, and
- controlling and validating field entry.

This VTE-profile includes support of an optional terminal-end locally attached printer.

14.8.3.2 Association Requirements

14.8.3.2.1 Functional Units

The following VT functional units are required for operation with this profile:

- Enhanced access-rules,
- Structured COs,
- Fields, and
- Reference Information Objects

The following VT functional units are optional for operation with this profile:

- Urgent Data

14.8.3.2.2 Mode

This is an S-mode profile.

14.8.3.3 Profile Body

```
Display-objects *(single occurrence)* =
     {
     display-object-name = A,
     DO-access           = "WAVAR,"
     dimensions          = "three,"
          x-dimension    =
               {
               x-bound        = profile-argument-r1,
               x-addressing   = "no constraint,"
               x-absolute     = "yes,"
               x-window       = x-bound
               },
          y-dimension    =
               {
               y-bound        = profile-argument-r2,
               y-addressing   = "no constraint,"
               y-absolute     = "yes,"
               y-window       = y-bound
               },
          z-dimension    =
               {
               z-bound        = "unbounded,"
               z-addressing   = "no constraint,"
               z-absolute     = "no,"
               z-window       = profile-argument-r3
               },
          erasure-capability = "yes,"
          repertoire-capability *(implicitly defined by r4)*,
          repertoire-assignment = profile-argument-r4,

          font-capability  *(implicitly defined by r5)*,
          font-assignment = profile-argument-r5,

          DO-emphasis = profile-argument-r6,

          foreground-colour-capability = profile-argument-r7,
          foreground-colour-assignment = profile-argument-r8,
          background-colour-capability = profile-argument-r7,
          background-colour-assignment = profile-argument-r9,

          block-definition-capability = "no,"
          field-definition-capability = "yes,"
          max-fields = "unbounded,"
          max-field-elements = profile-argument-r10,
          access-outside-fields = profile-argument-r11
          },
```

```
Control-objects =
{
     { *(Field Definition CO)*
     CO-name             = FD,
     CO-type-identifier  = vt-b-sco-fdco,
     CO-structure        = "non-parametric,"
     CO-access           = "WAVAR" + profile-argument-r12,
     CO-priority         = "normal,"
     CO-trigger          = "not-selected"
     },

     { *(Field Entry Instructions CO)*
     CO-name             = EI,
     CO-type-identifier  = "mandatory-feico,"
     CO-structure        = "non-parametric,"
     CO-access           = "WAVAR" + profile-argument-r12,
     CO-priority         = "normal,"
     CO-trigger          = "not-selected"
     },

     { *(Field Entry Pilot CO)*
     CO-name             = EP,
     CO-type-identifier  = "mandatory-fepco,"
     CO-structure        = "non-parametric,"
     CO-access           = "WAVAR" + profile-argument-r12,
     CO-priority         = "normal,"
     CO-trigger          = "not-selected"
     },

     { *(Context CO)*
     CO-name             = CC,
     CO-type-identifier  = vt-b-sco-cco,
     CO-structure        = 6,
     CO-access           = "WAVAR,"
     CO-priority         = "normal,"
     CO-trigger          = "not-selected"
     },

     { *(Transmission Policy CO)*
     CO-name             = TP,
     CO-type-identifier  = vt-b-sco-tpco,
     CO-structure        = 1,
     CO-access           = "WAVAR" + profile-argument-r12,
     CO-priority         = "normal,"
     CO-trigger          = "not-selected,"
     CO-category         = "boolean,"
     CO-size             = 4
     },
```

```
{ *(Multiple occurrence of optional COs. All
unspecified VTE-parameters of such COs are determined
by their CO-type-identifier through their registered
definition. They may include parameters specified to
be additional profile arguments, which should follow
the appropriate CO-type-identifier argument value)*

CO-name             = profile-argument-r13,
CO-type-identifier  = profile-argument-r14
},

{ *(Form Waiting Time CO)*
CO-name             = WT,
CO-type-identifier  = "waiting-time,"
CO-structure        = 1,
CO-access           = "WAVAR,"
CO-priority         = "normal,"
CO-trigger          = "not-selected,"
CO-category         = "integer,"
CO-size             = 65535
},
```

(The initial value for WT is zero, implying that a Form Waiting Time is not to be used.)

(The following four COs, (SA, UA, ST, and UT), are registered with the OIW registration authority and are referenced by this profile.)

```
{ *(As defined in 14.7.1 of this document)*
CO-name             = SA,
CO-type-identifier  = oiw-vt-co-misc-sa,
CO-structure        = 1,
CO-access           = "WAVAR" + profile-argument-r12,
CO-priority         = "normal,"
CO-trigger          = "not-selected,"
CO-category         = "symbolic,"
CO-size             = 11
},

{ *(As defined in 14.7.2 of this document)*
CO-name             = UA,
CO-type-identifier  = oiw-vt-co-misc-ua,
CO-structure        = 1,
CO-access           = profile-argument-r12,
CO-priority         = "urgent,"
CO-trigger          = "not-selected,"
CO-category         = "symbolic,"
CO-size             = 11
},
```

```
        { *(As defined in 14.7.3 of this document)*
        CO-name              = ST,
        CO-type-identifier   = oiw-vt-co-misc-st,
        CO-structure         = 1,
        CO-access            = "WAVAR" + opposite of
                                        profile-argument-r12,
        CO-priority          = "normal,"
        CO-trigger           = "not-selected,"
        CO-category          = "integer,"
        CO-size              = 65535
        },

        { *(As defined in 14.7.4 of this document)*
        CO-name              = UT,
        CO-type-identifier   = oiw-vt-co-misc-ut,
        CO-structure         = 1,
        CO-access            = opposite of profile-argument-r12,
        CO-priority          = "urgent,"
        CO-trigger           = "not-selected,"
        CO-category          = "integer,"
        CO-size              = 65535
        }
),


Device-objects  *(single or double occurrence)* =
{
        {
        device-name = D,
        device-default-CO-access = "WAVAR,"
        device-default-CO-priority = "normal,"
        device-default-CO-trigger = "not-selected,"
        device-default-CO-initial-value = 1."true,"
        device-repertoire-assignment = profile-argument-r15,
        device-font-assignment = profile-argument-r16,
        device-emphasis = profile-argument-r17,
        device-foreground-colour-assignment =
                                profile-argument-r18,
        device-background-colour-assignment =
                                profile-argument-r19,
        device-minimum-X-array-length = profile-argument-r20,
        device-minimum-Y-array-length = profile-argument-r21,
        device-control-object = FD,
        device-control-object = CC,
        device-control-object = SA,
        device-control-object = UA,
        device-control-object = ST,
        device-control-object = UT,
        device-control-object = WT,
        device-control-object = TP,
```

```
        device-control-object = profile-argument-r22,
        device-display-object = A
        },

        IF r23 = "true" THEN   *(define printer)*
        {
        device-name = P,
        device-default-CO-access = "NSAC,"
        device-default-CO-priority = "high,"
        device-default-CO-trigger = "not-selected,"
        device-default-CO-initial-value = 1."false,"
        device-repertoire-assignment = profile-argument-r24,
        device-font-assignment = profile-argument-r25,
        device-emphasis = profile-argument-r26,
        device-foreground-colour-assignment =
                                  profile-argument-r27,
        device-background-colour-assignment =
                                  profile-argument-r28,
        device-minimum-X-array-length = profile-argument-r29,
        device-minimum-Y-array-length = profile-argument-r30,
        device-control-object = FD,
        device-control-object = SA,
        device-control-object = UA,
        device-control-object = profile-argument-r31,
        device-display-object = A
        }
},

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

FIXED Field Entry Instruction Definitions - non-parametric

Optional Field
- field entry is optional. This FEI is provided for completeness only, as a field not linked to one of the Mandatory field, Selectable field or Protected field FEIs is necessarily optional. This FEI can never be violated.

Mandatory Field
- field entry is mandatory. Violation of this FEI will occur if all array elements of this field are empty when one of the reactions FER01 (Transmit updates) or FER02 (Relinquish WAVAR) is initiated. See also the specification of these reactions given below.

Protected Field
- the field is protected from field entry. Violation of this FEI will occur if an attempt is made to change the

primary or secondary attribute of any array element of this field.

Fill Field
- all array elements k=1 through k=last must have a primary attribute. Violation of this FEI will occur if any array element of this field is empty when one of the reactions FER01 (Transmit updates) or FER02 (Relinquish WAVAR) is initiated. See also the specification of these reactions given below.

Echo Received Character
- allowed field entry characters are to be echoed as received. This FEI is provided for completeness only, as by default characters will be echoed as received unless the field is linked to either the Echo Off or the Echo Specified Character FEI. This FEI can never be violated.

Echo Off
- received field entry characters should not be echoed. This FEI can never be violated.

Ignore Case
- if this FEI is linked to a field, upper and lower case alphabetic characters should be considered as equivalent during the validation of field input against all other FEIs linked to the same field. This affects the interpretation of the Allowed First Characters, Allowed Characters, Disallowed Characters and Allowed String Values FEIs, including the precedence rules between the first three of these FEIs. This FEI can never be violated.

Inhibit Logical Rendition Attribute Operation
- no form of logical attribute operation, with the exception of character repertoire switching as given below, can be performed on the field. Character repertoire changes are permitted if also permitted by Allowed First Characters or Allowed Characters, see below. This FEI is intended to be used when the rendition secondary attributes are to be kept under "application" control. See, for example, Allowed First Characters for a case of reference to the field modal values.

DYNAMIC Field Entry Instruction Definitions - parametric

Selectable field
- the field is selectable, i.e., field entry is not permitted but information is conveyed by the selection of one such field from a number of alternatives.

The manner in which the field that is the current candidate for selection is displayed on the real device is determined by the optional "visit" parameter of this FEI. This parameter specifies the secondary attributes to be used for showing or highlighting this candidate to the user. If it is omitted, an implementation-dependent default is used.

The manner in which the field that is actually selected is displayed on the real device is determined by the optional "select" parameter of this FEI. This parameter specifies the secondary attributes to be used for showing or highlighting the selected field to the user. If it is omitted, an implementation-dependent default is used.

The mechanisms for moving among candidates and for actually selecting the current candidate are implementation defined. Typically, a selectable field will be considered as a candidate for selection when the cursor is placed on a character within the selectable field. Actual selection generates the Field Selected FEE. A selected field is indicated in a delivered update by an addressing operation setting k=1 and f and z to indicate the selected field. These values will be reported to the host in the CCO if WAVAR is relinquished in reaction to this FEE. Violation of this FEI will occur if an attempt is made to change the primary or secondary attribute of any array element of this field.

Echo Specified Character
- specifies the character which is to be echoed to the user in response to each allowed character entered into the field. The secondary attributes of the echoed character may be specified. Any secondary attribute that is not given an explicit value in the FEI takes a default value in accordance with Definitive Note 4. This FEI can never be violated.

Minimum Entry
- all array elements k=1 through k=Minimum Entry must have a primary attribute. If Minimum Entry exceeds field size, then all positions in the field must be filled. Violation of this FEI will occur if any of the specified array elements are empty when one of the reactions FER01 (Transmit updates) or FER02 (Relinquish WAVAR) is initiated. See also the specification of these reactions given below. When a field is associated with both the Optional Field FEI and a Minimum Entry FEI, the field is optional but if entry is elected, the number of characters specified by the Minimum Entry FEI must then be entered.

Allowed First Characters
- specifies a set of allowed characters for the first character position of the field. Either primary attributes alone or both primary and secondary attributes may be checked; see Definitive Note 3.

Allowed Characters
- specifies a set of allowed characters for all character positions within the field. Either primary attributes alone or both primary and secondary attributes may be checked; see Definitive Note 3. If Allowed First Characters and Allowed Characters are both specified for a particular field, then the set of Allowed First Characters applies to the first character position of the field and the set of Allowed Characters applies to the second through last character positions of the field.

Disallowed Characters
- specifies a set of disallowed characters for all character positions within a field. Either primary attributes alone or both primary and secondary attributes may be checked; see Definitive Note 3. If Allowed First Characters and Disallowed Characters are both specified for a particular field, then the set of Allowed First Characters applies to the first character position of the field and the set of Disallowed Characters applies to the second through last character positions of the field. When a field is associated with Allowed Characters FEI(s) and Disallowed Characters FEI(s) that have characters in common, the common characters are considered as disallowed.

Entry Invoke Character
- specifies the attributes to be used for showing or highlighting to the user where the next character entry is to be made. Both primary and secondary attributes, or secondary attributes alone, may be specified to over-ride the corresponding values present in the array element concerned. Any secondary attribute that is not given an explicit value in the FEI takes a default value in accordance with Definitive Note 4. Fields that are not linked to an Entry Invoke Character FEI, utilize a device dependent entry invoke character which may or may not be represented in the character repertoire negotiated for the device. This FEI can never be violated.

Waiting Time
- specifies the number of seconds to wait for field entry to complete after the cursor has been positioned within the field. Fields that are not associated with a Waiting Time FEI are not subject to the "Field Waiting Time Expired"

Field Entry Event. Note that an overall waiting time for an entire form may be set by use of the "waiting-time" control object defined in Definitive Note 1. This FEI can never be violated.

Allowed String Values
- specifies a list of strings which identify valid field values. The strings are specified as either a discrete OCTET STRING or a range of OCTET STRING, or combination of both.

Ranges are specified using a lower "value" OCTET STRING and a higher "value" OCTET STRING. The "value" of an OCTET STRING is the integer value derived from the collating sequence corresponding to the repertoire explicitly or implicitly specified for the OCTET STRING. For example, the ISO 646 string 'AB' has the integer value 4142(16) and the string 'ABC' has the value 414243(16).

When strings of unequal length are compared, the smaller string is filled on the right with enough spaces to make the strings of equal length. The comparison of ISO 646 strings 'AB' and 'ABC' would be accomplished by first converting the string 'AB' to 'AB' thus creating the value 414220(16) to be compared against the value 414243(16). The value of the space character is derived from the collating sequence corresponding to the repertoire identified in the field modal attributes. If this repertoire does not contain a space, then the value 20(16) is used.

Either primary attributes alone or both primary and secondary attributes may be checked; see Definitive Note 3. A single set of secondary attribute values may be specified for each individual OCTET STRING or range of OCTET STRINGs.

Allowed Numeric Values
- specifies a list of numeric strings which identify valid field values. The strings are specified as either a discrete OCTET STRING or a range of OCTET STRING, or a combination of both.

Ranges are specified using a lower "value" OCTET STRING and a higher "value" OCTET STRING. The "value" of an OCTET STRING is the integer value derived from the collating sequence corresponding to the repertoire explicitly or implicitly specified for the OCTET STRING. For example, the ISO 646 string '12' has the integer value 3132(16) and the string '123' has the integer value 313233(16).

When strings of unequal length are compared, the smaller string is filled on the left with enough zero characters to make the strings of equal length. The comparison of ISO 646 strings '12' and '123' would be accomplished by first converting the string '12' to '012' thus creating the value 303132(16) to be compared against the value 313233(16). The value of the zero character is derived from the collating sequence corresponding to the repertoire identified in the field modal attributes. If this repertoire does not contain a zero, then the value 30(16) is used.

Either primary attributes alone or both primary and secondary attributes may be checked; see Definitive Note 3. A single set of secondary attribute values may be specified for each individual OCTET STRING or range of OCTET STRINGs.

Mutually Exclusive FEIs

Some FEIs specify field entry validation rules that are in conflict with the rules specified by other FEIs. For example, a particular field cannot be both "protected" and "mandatory." Such conflicting FEIs cannot be associated with the same field. The following table defines the sets of conflicting FEIs.

Table 14.1. Sets of conflicting FEIs

FEI	Conflicting FEIs
Optional Field	Mandatory Field, Selectable Field, Protected Field.
Mandatory Field	Optional Field, Selectable Field, Protected Field.
Selectable Field	All except Entry Invoke Character and Waiting Time.
Protected Field	All.
Fill Field	Selectable Field, Protected Field, Allowed String Values, Allowed Numeric Values.
Echo Received Character	Selectable Field, Protected Field, Echo Off, Echo Specified Character.
Echo Off	Selectable Field, Protected Field, Echo Received Character, Echo Specified Character.
Ignore Case	Selectable Field, Protected Field.
Inhibit Logical Rendition Attribute Operation	Selectable Field, Protected Field.
Echo Specified Character	Selectable Field, Protected Field, Echo Off, Echo Received Character
Minimum Entry	Selectable Field, Protected Field.
Allowed First Characters	Selectable Field, Protected Field, Allowed String Values, Allowed Numeric Values.
Allowed Characters	Selectable Field, Protected Field, Allowed String Values, Allowed Numeric Values.
Disallowed Characters	Selectable Field, Protected Field, Allowed String Values, Allowed Numeric Values.
Entry Invoke Character	Protected Field.
Waiting Time	Protected Field.
Allowed String Values	Selectable Field, Protected Field, Fill Field, Allowed First Characters, Allowed Characters, Disallowed Characters, Allowed Numeric Values.

Allowed Numeric Values — Selectable Field, Protected Field, Fill Field, Allowed First Characters, Allowed Characters, Disallowed Characters, Allowed String Values.

FEICO Update Syntax

In the following syntax, ASN.1 Value Assignments have been used to attach value references to values of type NULL. This enables the values to be referenced by these names alone, without the need to follow the identifier explicitly with the value NULL.

```
FEI DEFINITIONS ::= BEGIN

FEI ::= CHOICE {
fei0                        [0]  IMPLICIT NULL,
fei1                        [1]  IMPLICIT NULL,
fei2                        [2]  IMPLICIT NULL,
fei3                        [3]  IMPLICIT NULL,
fei4                        [4]  IMPLICIT NULL,
fei5                        [5]  IMPLICIT NULL,
fei6                        [6]  IMPLICIT NULL,
fei7                        [7]  IMPLICIT NULL,
selectableField             [8]  IMPLICIT SEQUENCE {
     visit     [0] IMPLICIT SecAttributes OPTIONAL,
     select    [1] IMPLICIT SecAttributes OPTIONAL },
echoSpecifiedCharacter      [9]  IMPLICIT Character,
minimumEntries              [10] IMPLICIT INTEGER,
allowedFirstCharacters      [11] IMPLICIT CharacterValues,
allowedCharacters           [12] IMPLICIT CharacterValues,
disallowedCharacters        [13] IMPLICIT CharacterValues,
entryInvokeCharacter        [14] CHOICE {
               [0]  IMPLICIT Character,
               [1]  IMPLICIT SecAttributes },
waitingTime                 [15] IMPLICIT INTEGER,
allowedStringValues         [16] IMPLICIT CharacterValues,
allowedNumericValues        [17] IMPLICIT CharacterValues }

optionalField               FEI  ::= fei0 NULL
mandatoryField              FEI  ::= fei1 NULL
protectedField              FEI  ::= fei2 NULL
fillField                   FEI  ::= fei3 NULL
echoReceivedChar            FEI  ::= fei4 NULL
echoOff                     FEI  ::= fei5 NULL
ignoreCase                  FEI  ::= fei6 NULL
inhibitLogRendAttOp         FEI  ::= fei7 NULL

Character ::= SEQUENCE {
     primaryValue   [0]  IMPLICIT PriValue,
```

```
        attributes      [1]  IMPLICIT SecAttributes OPTIONAL }
        -- When used as one element of a comparison, secondary
        -- attributes are to be compared only if the attributes
        -- element is present.

CharacterValues ::= SEQUENCE OF SEQUENCE {
        lowValue        [0]  IMPLICIT Character,
        highValue       [1]  IMPLICIT PriValue OPTIONAL }
        -- The default for highValue is the associated
        -- lowValue.  Octet values specified for highValue
        -- are constrained by the repertoire corresponding
        -- to the lowValue value.  The relationship
        -- [lowValue <= highValue] must be true.

PriValue ::= OCTET STRING
        -- The octet string comprising a value of the PriValue
        -- type is constrained to the encoding of a sequence
        -- of characters from the repertoires negotiated for
        -- the associated Display Object.  When used in the
        -- ASN.1 module FEI, the octet string is restricted to
        -- the encoding of a single character except for its
        -- use in allowedStringValues and allowedNumeric-
        -- Values.

SecAttributes ::= SEQUENCE {
        repertoire          [0]  IMPLICIT INTEGER OPTIONAL,
        foregroundColour    [1]  IMPLICIT INTEGER OPTIONAL,
        backgroundColour    [2]  IMPLICIT INTEGER OPTIONAL,
        emphasis            [3]  IMPLICIT PrintableString
                                                  OPTIONAL,
        font                [4]  IMPLICIT INTEGER OPTIONAL }

END *(FEI DEFINITIONS)*
```

FEICO "mandatory-feico" Initial Content

For each FEIRxx, xx identifies the integer value to be used as "feirList recordIndex" in FDCOUpdate operations. FEICOUpdate operations must use an "index" greater than 127. Note that the character oriented FEIRs for the initial FEICO utilize the default secondary attributes, and that the Selectable Field FEI uses implementation-dependent defaults for the 'visit' and 'select' secondary attributes. The FEIR contents are specified in terms of ASN.1 Value Notation appropriate to the FEICO Update Syntax specified above.

```
FEIR00    -- Not used --

FEIR01    optionalField
```

```
FEIR02    mandatoryField

FEIR03    selectableField { }

FEIR04    protectedField

FEIR05    fillField

FEIR06    echoReceivedChar

FEIR07    echoOff

FEIR08    ignoreCase

FEIR09    inhibitLogRendAttOp

FEIR10    allowedCharacters {{ lowValue {'41'H},
          highValue '5A'H }} -- A,B,...,Z --

FEIR11    allowedCharacters {{ lowValue {'61'H},
          highValue '7A'H }} -- a,b,...,z --

FEIR12    allowedCharacters {{ lowValue {'30'H},
          highValue '39'H }} -- 0,1,...,9 --

FEIR13    disallowedCharacters {{ lowValue {'41'H},
          highValue '5A'H }} -- A,B,...,Z --

FEIR14    disallowedCharacters {{ lowValue {'61'H},
          highValue '7A'H }} -- a,b,...,z --

FEIR15    disallowedCharacters {{ lowValue {'30'H},
          highValue '39'H }} -- 0,1,...,9 --

FEIR16-FEIR127 -- These values are reserved --
```

Field Entry Event Definitions

The Field Entry Events for the mandatory FEPCO are listed below. A parameter of type "Range" is a sequence of integer pairs, each with an optional bitmask. Each pair gives the end points of an interval of integer values. An integer value lies within the range specified if, after applying the bitmask (if given) to its binary form, it lies in any of these intervals. The end points of an interval are included in the values of that interval.

It is permissible for the ranges specified by the FEEs referenced in the entry control FEPR-list of a field to overlap. When an event occurs which is referenced in this

way by more than one FEPR linked to the current field, the FEPR invoked is the first FEPR in the FEPR-list which both references the event and for which the Field Entry Conditions are satisfied.

FEE00
- not used.

FEE01 Logical Keystroke event (Range)
- this event takes a range of integers as a parameter, and occurs when a Logical Keystroke occurs within the specified range. The Logical Keystroke is either initiated by the Logical Keystroke FER or by the human user, see Definitive Note 8.

FEE02 Field entry complete
- this event is generated by entry of a character into the last position in a field. It need not imply that all character positions in the field have been entered, since these positions are not necessarily written sequentially. Local cursor movements, for example, may be used during local editing to move the current entry position around the screen.

FEE03 Field selected
- this event is generated by the selection of a field that is linked to the Selectable Field FEI. The means by which the current candidate for selection is actually selected is implementation dependent.

FEE04 Field Waiting Time expired
- the field waiting time specified by the Waiting Time FEI linked to the current field has been exceeded. Fields not linked to such an FEI are not subject to this event.

FEE05 Field Entry Instruction violation
- some of the defined FEIs imply Field Entry Validation by the terminal VT-user. Fields linked to such FEIs are candidates for erroneous field entry. This event is generated when such a violation occurs, thus enabling linkage to Field Entry Reactions that may signal a visual or audible indication of such a violation, or alternatively may terminate local entry and relinquish WAVAR. A Violation FEC is available to allow different reactions according to which FEIR is violated. When the reaction is to relinquish WAVAR, the Entry-control index and FEPR index elements of the Context Control Object will inform the host which FEPR caused the return. If this FEPR has made use of the Violation FEC, this FEC will identify to the host that the violated FEIR was one of those in the list that forms the

parameter value for the FEC. Unique identification of the FEIR is obtained if this list contains only one FEIR. The host can then take whatever action is appropriate to the FEIR or FEIRs so identified.

Field Entry Condition Definitions

The elementary Field Entry Conditions for the mandatory FEPCO are defined below. Composite conditions can be built by use of the specified parameters, and an individual FEPR can include multiple conditions in accordance with 20.3.5.2 of ISO 9040.

A parameter of type Action is specified either as an explicit integer value or as the current keystroke, see Definitive Note 8. Such a parameter evaluates to an integer of the type STCO.Key defined in clause 14.7.3.7. That clause also defines names of logical keystrokes associated with these integers. The local actions associated with such values are defined in Definitive Note 9.

FEC00
- not used.

FEC01 No previous field
- the current field has no currently defined previous field, in the sense of 20.3.3.4 of ISO 9040.

FEC02 No next field
- the current field has no currently defined next field, in the sense of 20.3.3.4 of ISO 9040.

FEC03 Start of field
- the current location for the next character entry is at the first location in the current field.

FEC04 End of field
- the current location for the next character entry is at the last location in the current field.

FEC05 At tab stop
- the current location for the next character entry is at a tabulation stop defined by the optional Horizontal Tabulation CO {ewos-vt-co-misc-ht} registered with the EWOS Registration Authority. If this CO is not present in the VTE, this condition is deemed to be always satisfied.

FEC06 At characters (Set of character values)

- the current location for the next character entry is at an array element whose current value is one of the specified characters. The set of characters is specified and interpreted in accordance with Definitive Note 3.

FEC07 Exits field (Action)
- the local action designated by the parameter value would move the location for the next character entry out of the current field.

FEC08 Exits forward path (Action)
- the local action designated by the parameter value would move the location for the next character entry out of the forward navigation path starting at the current field.

FEC09 Exits backward path (Action)
- the local action designated by the parameter value would move the location for the next character entry out of the backward navigation path starting at the current field.

FEC10 Exits x-array (Action)
- the local action designated by the parameter value would move the location for the next character entry out of the current x-array.

FEC11 Exits y-array (Action)
- the local action designated by the parameter value would move the location for the next character entry out of the current y-array.

FEC12 Not FEC (FEC)
- this condition is satisfied precisely when the FEC given as its parameter is not satisfied.

FEC13 And FECs (Set of FEC)
- this condition is satisfied when each of the conditions in the set comprising its parameter is satisfied.

FEC14 Or FECs (Set of FEC)
- this condition is satisfied when at least one of the conditions in the set comprising its parameter is satisfied.

FEC15 Violation (Set of FEIR Identifiers)
- this condition is provided for use in conjunction with the Field Entry Instruction Violation FEE. Its parameter is an FEIR-list specified as a set of FEIR identifiers. Each identifier is a pair <FEICO-name, index> where index is an integer addressing a record in the FEICO whose name is specified. This FEC is satisfied if the FEIR whose violation generated this event is one of the FEIRs in this FEIR-list.

If it is used in conjunction with any other FEE then this condition is true.

FEC16 Unconditional
- this condition is always true. It is given for completeness only, and has the same effect as an empty set of conditions in an FEPR.

Field Entry Reaction Definitions

The Field Entry Reactions for the mandatory FEPCO are defined below. The significance of a parameter of type "Action" is as described for Field Entry Conditions. A parameter of type "ResetAttribute" may take either of the two values "reset" and "noReset." Such a parameter controls the effect of an erase operation on the secondary attributes of the erased elements, corresponding to the values "yes" and "no" for the reset-attribute parameter of a LOGICAL-ERASE operation as defined in 19.2.3.5 of ISO 9040.

FER00
- not used.

FER01 Transmit updates
- the host copy of the CCA is updated to correspond to the terminal copy by the transmission of all undelivered update operations. The operations required to update field contents are controlled by the T-policy component of the Field Definition Record for the fields concerned. However, if this FER generates an FEI violation in accordance with the specifications of the FEICO(s) present in the VTE, and if the current field is also linked to an FEPR with event FEE05 (FEI violation) and satisfied conditions, then this FER is not performed and that FEPR is activated; the original FEPR is abandoned.

FER02 Relinquish WAVAR
- the action described under Transmit Updates is performed, followed by return of the WAVAR access right to the host. However, if this FER generates an FEI violation in accordance with the specifications of the FEICO(s) present in the VTE, and if the current field is also linked to an FEPR with event FEE05 (FEI violation) and satisfied conditions, then this FER is not performed and that FEPR is activated; the original FEPR is abandoned.

FER03 Erase field right (Reset attribute)
- the primary attribute value is cancelled for all elements of the current field from the current character entry

location to the end of the field. The effect on the secondary attribute values is determined by the reset-attribute parameter as described above.

FER04 Erase path right (Reset attribute)
- the primary attribute value is cancelled for all elements of all unprotected fields in the forward navigation path containing the current field, from the current character entry location onwards. Note that the forward navigation path may not terminate, as its definition in 20.3.3.4 of ISO 9040 does not prohibit looping. When a loop is entered during this operation, the operation terminates when all elements of the entered loop have been erased. The effect of this operation on the secondary attribute values is determined by the reset-attribute parameter as described above.

FER05 Local action (Action)
- that local action is performed which is designated by the given parameter value. The specification of these local actions is given in Definitive Note 9.

FER06 Logical Keystroke (Action)
- initiate the FEPR processing which would occur if the given keystroke had occurred. This may itself cause the Logical Keystroke FER and hence recursive processings of FERs. Processing of current FERs is suspended until this recursive processing is complete. During recursive processing, the current keystroke is taken as the argument to this FER. When the recursive processing is complete, the previous keystroke is restored and processing of current FERs is resumed.

FER07 Update ST CO (Action)
- the integer value corresponding to the given parameter is written to the Sequenced Terminal CO. This FER will usually be followed by a Transmit Updates or Relinquish WAVAR FER to communicate the update to the application.

FER08 Update UT CO (Action)
- the integer value corresponding to the given parameter is written to the Unsequenced Terminal CO. This update will be communicated to the application immediately.

FER09 Execute RIO record (RIO record id)
- an EXECUTE-RECORD operation is performed on the RIO record specified in the parameter, in accordance with 22.4.1 of ISO 9040.

FER010 Call RIO record (RIO record id)

- a CALL-RECORD operation is performed on the RIO record specified in the parameter, in accordance with 22.4.2 of ISO 9040.

FER11 Visual indication
- present a visual indication in response to Field Entry Instruction violations.

FER12 Audible indication
- present an audible indication in response to Field Entry Instruction violations.

FER13 Conditional branch (if: FEC, then: Optional sequence of FER, else : Optional sequence of FER)
- if the condition given by the "if" parameter is satisfied then perform the sequence of reactions given by the "then" parameter, else perform the sequence of reactions given by the "else" parameter.

FER14 Prevent further entry
- it is recommended that if a type-ahead buffer is in use by the local user interface, this reaction should prevent further entry into the buffer. Attempted entry may then sound an alarm or be signalled by some other local means, but is not an FEI violation. If the WAVAR access right is relinquished without this reaction being invoked, the buffer may continue to accept entries. Entry into the buffer is resumed when WAVAR is next returned to the terminal. It is not a violation of this profile specification if the terminal VT-user does not behave in the intended manner.

FER15 Write disallowed character
- the most recent disallowed character is written as if it were not disallowed. If there has been no disallowed character, the effect is null. This FER is used when it is desired to trap the entry of a particular character, not to forbid it but instead to generate some other reactions in addition to the character entry.

FER16 Write string (Character string)
- the character string given as a parameter is written as LOGICAL TEXT to the current entry location without regard to FEICO control. If the end of the field is reached before the string has been written in its entirety, the reaction is terminated prematurely.

Field Entry Pilot Update Syntax

In the following syntax, ASN.1 Value Assignments have been used to attach value references to values of type NULL. This enables the values to be referenced by these names alone, without the need to follow the identifier explicitly with the value NULL.

```
FEPR DEFINITIONS ::= BEGIN

FEE ::= CHOICE {
     logicalKeystroke          [1] IMPLICIT Range,
     fee02                     [2] IMPLICIT NULL,
     fee03                     [3] IMPLICIT NULL,
     fee04                     [4] IMPLICIT NULL,
     fee05                     [5] IMPLICIT NULL  }

     fieldEntryComplete        FEE ::= fee02 NULL
     fieldSelected             FEE ::= fee03 NULL
     fieldWaitTimeExpired      FEE ::= fee04 NULL
     feiViolation              FEE ::= fee05 NULL


FEC ::= CHOICE {
     fec01              [1]  IMPLICIT NULL,
     fec02              [2]  IMPLICIT NULL,
     fec03              [3]  IMPLICIT NULL,
     fec04              [4]  IMPLICIT NULL,
     fec05              [5]  IMPLICIT NULL,
     atChar             [6]  IMPLICIT FEI.CharacterValues,
     exitsField         [7]  Action,
     exitsForwardPath   [8]  Action,
     exitsBackwardPath  [9]  Action,
     exitsXarray        [10] Action,
     exitsYarray        [11] Action,
     not                [12] FEC,
     and                [13] IMPLICIT SET OF FEC,
     or                 [14] IMPLICIT SET OF FEC,
     violation          [15] IMPLICIT SET OF SEQUENCE
                        { feicoName      PrintableString,
                          recordIndex    INTEGER },
     fec16              [16] IMPLICIT NULL  }

noPreviousField            FEC ::= fec01 NULL
noNextField                FEC ::= fec02 NULL
startField                 FEC ::= fec03 NULL
endField                   FEC ::= fec04 NULL
atTab                      FEC ::= fec05 NULL
unconditional              FEC ::= fec16 NULL
```

```
FER ::= CHOICE {
     fer01              [1]  IMPLICIT NULL,
     fer02              [2]  IMPLICIT NULL,
     eraseFieldRight    [3]  IMPLICIT ResetAttribute,
     erasePathRight     [4]  IMPLICIT ResetAttribute,
     local              [5]  Action,
     logicalKeystroke   [6]  Action,
     updateSTCO         [7]  Action,
     updateUTCO         [8]  Action,
     executeRIO         [9]  IMPLICIT RIORecordID,
     callRIO            [10] IMPLICIT RIORecordID,
     fer11              [11] IMPLICIT NULL,
     fer12              [12] IMPLICIT NULL,
     branch             [13] IMPLICIT SEQUENCE {
          if      [1] FEC,
          then    [2] IMPLICIT SEQUENCE OF FER OPTIONAL,
          else    [3] IMPLICIT SEQUENCE OF FER OPTIONAL  },
     fer14              [14] IMPLICIT NULL,
     fer15              [15] IMPLICIT NULL,
     writeString        [16] IMPLICIT SEQUENCE OF
                        FEI.Character
     -- The string written by this FER is the
     -- concatenation of the strings specified by
     -- the individual FEI.Character values. -- }

transmitUpdates             FER  ::= fer01 NULL
relinquishWAVAR             FER  ::= fer02 NULL
visualIndication            FER  ::= fer11 NULL
audibleIndication           FER  ::= fer12 NULL
preventFurtherEntry         FER  ::= fer14 NULL
writeDisallowedChar         FER  ::= fer15 NULL


RIORecordID ::= SEQUENCE {
     rioName       [1] IMPLICIT PrintableString OPTIONAL,
     -- optional if there is only 1 RIO in the VTE
     recordID      [2] IMPLICIT PrintableString  }


Range ::= SEQUENCE OF SEQUENCE {
                   [1] IMPLICIT STCO.Key,
                   [2] IMPLICIT STCO.Key  OPTIONAL,
     mask          [3] IMPLICIT BIT STRING OPTIONAL  }

     -- The first two values of each trio represent an
     -- interval of logical keystroke values.  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.
     -- If the optional mask is given, then the value being
```

```
     -- tested is bitwise logically ANDed with the mask
     -- before being compared with the end points of the
     -- interval.


ResetAttribute ::= BOOLEAN

reset                ResetAttribute ::= TRUE
noReset              ResetAttribute ::= FALSE

Action ::= CHOICE {
                     [1] IMPLICIT STCO.Key,
     current         [2] IMPLICIT NULL  }

currentKeystroke   Action ::= current NULL

-- The ASN.1 module STCO is defined in the specification of
-- the Sequenced Terminal CO in clause 14.7.3.  STCO.Key is
-- an integer type with a named number list, each named
-- number representing a specific logical keystroke as
-- defined for that CO.


END *(FEPR DEFINITIONS)*
```

FEPCO "mandatory-fepco" Initial Content

For each FEPRxx, xx identifies the integer value to be used as "feprList recordIndex" in FDCOUpdate operations. FEPCOUpdate operations must use an "index" greater than 127. The FEPR contents are specified in terms of ASN.1 Value Notation appropriate to the FEPCO Update Syntax specified above. Note that "shiftTab" is a named integer of type STCO.Key. The local action it designates is defined in Definitive Note 9 to be movement of the current character entry position to the first location of the next field in the forward navigation path.

FEPR No	Component	ASN.1 Description
FEPR00		-- Not used --
FEPR01	FEE	logicalKeystroke { { 0, 65535 } }
	FEC	unconditional
	FER01	updateSTCO currentKeystroke
	FER02	relinquishWAVAR
FEPR02	FEE	fieldEntryComplete
	FEC	noNextField
	FER	relinquishWAVAR

FEPR03	FEE	fieldEntryComplete
	FEC	not noNextField
	FER	local shiftTab
FEPR04	FEE	fieldSelected
	FEC	unconditional
	FER	relinquishWAVAR
FEPR05	FEE	fieldWaitTimeExpired
	FEC	noNextField
	FER	relinquishWAVAR
FEPR06	FEE	fieldWaitTimeExpired
	FEC	not noNextField
	FER	local shiftTab
FEPR07	FEE	feiViolation
	FEC	unconditional
	FER	visualIndication
FEPR08	FEE	feiViolation
	FEC	unconditional
	FER	audibleIndication
FEPR09-FEPR127		-- Reserved --

14.8.3.4 Profile Arguments

r1 - is optional and provides for the negotiation of a value for the VTE-parameter x-bound. It takes an integer value greater than zero. Default is 80.

r2 - is optional and provides for the negotiation of a value for the VTE-parameter y-bound. It takes an integer value greater than zero. Default is 24.

r3 - is optional and provides for the negotiation of a value for the VTE-parameter z-window. It takes an integer value greater than zero. Default is 1.

r4 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter repertoire-assignment. The value for the VTE-parameter repertoire-capability is implied by the number of occurrences of this profile argument. Default is a single occurrence with the value {value iso2022 {'2842'H}} of ASN.1 type

CDS.RepertoireAssignment as defined in ISO 9041, designating the GL set ISO 2375 Reg. No. 6 (ASCII).

r5 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter font-assignment. The font-assignment-type component of a font-assignment value is an ASN.1 OBJECT IDENTIFIER that designates a registered syntax and semantics for the font-assignment-value component. The value for the VTE-parameter font-capability is implied by the number of occurrences of this profile argument. If there are no explicit occurrences of this profile argument then the font-capability and font-assignment VTE-parameters take the default values specified in ISO 9040.

r6 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter DO-emphasis. The syntax and semantics for this VTE-parameter are specified in Definitive Note 6, and for this profile argument are specified in B.17.4 of ISO 9040. The default value for the occurrence corresponding to each unspecified subattribute is the ASN.1 PrintableString of length 1 specifying the explicit modal default value for that subattribute.

r7 - is optional and provides for the negotiation of a value for the VTE-parameters foreground-colour-capability and background-colour-capability. Default is 8. This argument is identified by the identifier for the VTE-parameter foreground-colour-capability for display object A.

r8 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter foreground-colour-assignment. The default values for unspecified occurrences of this profile argument are the corresponding values from the ordered list {"white," "black," "red," "cyan," "blue," "yellow," "green," "magenta"}. There are no default values if the value of the VTE-parameter foreground-colour-capability exceeds 8.

r9 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter background-colour-assignment. The default values for

unspecified occurrences of this profile argument are the corresponding values from the ordered list {"black," "white," "cyan," "red," "yellow," "blue," "magenta," "green"}. There are no default values if the value of the VTE-parameter background-colour-capability exceeds 8.

r10 - is optional and provides for the negotiation of a value for the VTE-parameter max-field-elements. Default is 1.

r11 - is optional and provides for the negotiation of a value for the VTE-parameter access-outside-fields. Default is "not allowed."

r12 - is mandatory and provides for the negotiation of a value for the VTE-parameter CO-access for the Field Definition, Field Entry Instruction, Field Entry Pilot, Transmission Policy, Sequenced Application, Unsequenced Application, Sequenced Terminal, and Unsequenced Terminal control objects. If the VT-association initiator is the terminal VT-user, it takes the value "WACA," otherwise it takes the value "WACI." This argument is identified by the identifier for CO-access for control object UA.

r13 - is optional, may occur a number of times and provides for the negotiation of a value for the VTE-parameter CO-name for optional registered COs. By default no optional COs are invoked.

r14 - is optional, may occur a number of times and provides for the negotiation of a value for the VTE-parameter CO-type-identifier for optional registered COs. The particular generic type concerned is determined from the CO-type-identifier by the register entry. The value vt-b-sco-nullrio selects an empty RIO. An occurrence of the previous argument specifies the presence of an optional CO in the VTE-profile. An occurrence of this argument is required for every occurrence of the previous argument. By default no optional COs are invoked.

r15 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-repertoire-assignment for the main device. Default is "null" for each unspecified occurrence.

r16 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-font-assignment for the main device. Default is "null" for each unspecified occurrence.

r17 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-emphasis for the main device. The syntax and semantics for this VTE-parameter are specified in Definitive Note 6, and for this profile argument are specified in B.17.4 of ISO 9040. Default is "null" for each unspecified occurrence.

r18 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-foreground-colour-assignment for the main device. Default is "null" for each unspecified occurrence.

r19 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-background-colour-assignment for the main device. Default is "null" for each unspecified occurrence.

r20 - is optional and provides for the negotiation of a value for the VTE-parameter device-minimum-X-array-length for the main device. It takes an integer value greater than zero. Default is the value of x-bound for the display object.

r21 - is optional and provides for the negotiation of a value for the VTE-parameter device-minimum-Y-array-length for the main device. It takes an integer value greater than zero. Default is the value of y-bound for the display object.

r22 - is optional, may occur a number of times and provides for the negotiation of additional values for the VTE-parameter device-control-object for the main device. By default there are no additional values.

r23 - is a special profile argument identified by the special-profile-arg-ident "Pp-1." It is optional and provides for the negotiation of a printer device. Default is "false."

r24 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-repertoire-assignment for the printer device. Default is "null" for each unspecified occurrence.

r25 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-font-assignment for the printer device. Default is "null" for each unspecified occurrence.

r26 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-emphasis for the printer device. The syntax and semantics for this VTE-parameter are specified in Definitive Note 6, and for this profile argument are specified in B.17.4 of ISO 9040. Default is "null" for each unspecified occurrence.

r27 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-foreground-colour-assignment for the printer device. Default is "black" for each unspecified occurrence.

r28 - is optional, may occur a number of times in an ordered list and provides for the negotiation of a value(s) for the VTE-parameter device-background-colour-assignment for the printer device. Default is "white" for each unspecified occurrence.

r29 - is optional and provides for the negotiation of a value for the VTE-parameter device-minimum-X-array-length for the printer device. It takes an integer value greater than zero. Default is the value of x-bound for the display object.

r30 - is optional and provides for the negotiation of a value for the VTE-parameter device-minimum-Y-array-length for the printer device. It takes an integer value greater than zero. Default is the value of y-bound for the display object.

r31 - is optional, may occur a number of times and provides for the negotiation of additional values for the VTE-parameter device-control-object for the printer device. By default there are no additional values.

14.8.3.5 Profile Dependent Control Objects

This profile uses the OIW registered Control Objects SA, UA, ST and UT. The profile defined values are specified in the body of this profile. The CO specifications require the usage of each CO to be specified in the profile. This is as follows.

14.8.3.5.1 Sequenced Application CO

This Control Object is defined in clause 14.7.1. It has CO-category "symbolic." Update of this CO with the value "audible_alarm" sounds an audible alarm in the terminal. Update with the value "visual_alarm" generates a visual indication of a signal from the application. All other values have no effect.

14.8.3.5.2 Unsequenced Application CO

This Control Object is defined in clause 14.7.2. It has CO-category "symbolic." Update of this CO with the value "audible_alarm" sounds an audible alarm in the terminal. Update with the value "visual_alarm" generates a visual indication of a signal from the application. All other values have no effect.

14.8.3.5.3 Sequenced Terminal CO

This Control Object is defined in clause 14.7.3. It has CO-category "integer." It is updated by the Update ST CO FER, and may be used to communicate uninterpreted keystrokes to the application.

14.8.3.5.4 Unsequenced Terminal CO

This Control Object is defined in clause 14.7.4. It has CO-category "integer." It is updated by the Update UT CO FER and is used to communicate uninterpreted keystrokes to the application urgently.

14.8.3.6 Profile Notes

14.8.3.6.1 Definitive Notes

1. The WT control object provides a mechanism for the application VT-user to specify a time in which all

the fields of a form must be completed. The terminal VT-user starts the timer at the time when it receives WAVAR. If the timer expires, further entry by the device is stopped, all undelivered updates are transmitted, and WAVAR is relinquished. The undelivered updates are transmitted followed by an update to this control object. The WT update is made using the current value of the WT control object. The device-control-object VTE-parameter is used to link this CO to the input device that it controls. The data element of this CO specifies the waiting time in seconds. A zero value signifies that a Form Waiting Time is not to be used. The initial value of this data element is zero.

2. If there are two or more Character-oriented FEIs of the same type associated with the same field, they are equivalent to a single FEI of that type whose parameter is the concatenation of the individual parameter values.

3. The following parameteric FEIs and FECs defined in clause 14.8.3.3 test equality of characters:

    Allowed First Characters FEI
    Allowed Characters FEI
    Disallowed Characters FEI
    Allowed String Values FEI
    Allowed Numeric Values FEI
    At Characters FEC

    The characters for each such FEI or FEC are specified by a parameter that includes an optional set of secondary attributes. If this set is included, the test is on both primary and secondary attributes; otherwise it is on primary attributes only. If the test is on primary attributes only, then characters which pass the test are allowed, disallowed or accepted, as appropriate, irrespective of the values of their secondary attributes. The set of secondary attributes need not specify an explicit value for every secondary attribute; in particular the empty set is permissible. Default values are used for unspecified secondary attributes. These are determined in accordance with Definitive Note 4.

4. The parameter values for a number of FEIs, FECs and FERs require default values to be used for

secondary attributes when such values are not specified explicitly by the parameter. The first choice default for each secondary attribute value is the field modal attribute value at the time that the FEI, FEC or FER is accessed. A first choice default value of "null" is resolved as specified in 19.2.3.1 of ISO 9040 for the LOGICAL-TEXT update operation.

5. When the Character oriented FEIs associated with a particular field have characters in common, the precedence algorithm given below is used.

   The Allowed First Characters FEI takes precedence over the Allowed Characters and Disallowed Characters FEIs for field character position k=1. The Disallowed Characters FEI takes precedence over the Allowed Characters FEI for all field character positions.

   The following example illustrates the conflict resolution algorithm. When a particular field is linked to the following three Character oriented FEIs:

   Allowed First Characters = a
   Allowed Characters = a,b
   Disallowed Characters = a

   the field must be entered with the letter "a" in the first character position of the field. The remaining character positions in the field are limited to containing the letter "b." Therefore field entry would be limited to a form such as "abbbb. ..."

6. The following syntax and semantics is mandatory for the emphasis and device-emphasis VTE-parameters. The scheme of B.17.3 of ISO 9040 is to be adopted except that the maximum length for an ASN.1 PrintableString used as an emphasis value is increased from 6 characters to 8 characters. Values "B" (Boxed) and "C" (Encircled) are deleted from subattribute "b." Two further subattributes are added, denoted by "g" and "h." The table of allowed character values, ISO 6429 SET GRAPHIC RENDITION parameter values and associated semantics given in B.17.3 of ISO 9040 is augmented by the addition of:

Subattribute "g" values

= "I"	3	Italicized characters

= "U" * 23 Upright, not italicized characters
= " " - No change

As in B.17.3 of ISO 9040, * indicates the value which is the explicit modal default value for the subattribute. Not all the values of this scheme need to have a 1-1 correspondence with emphasis levels available on the real device. The device object defines the real mapping.

7. When default values are defined for a multiple-occurrence profile argument and fewer occurrences are negotiated than are required by the value of a parent VTE-parameter, the remaining occurrences still take the specified default values.

8. Every action corresponding to the operation of an object updating device shall be assigned a non-negative integer value. This value shall be interpreted as a logical keystroke in accordance with the definitions of the Sequenced Terminal CO and Unsequenced Terminal CO in clauses 14.7.3 and 14.7.4.

   Values in the range 0-255 are used to generate entry of characters into the Display Object from the available repertoires. Values greater than 255 generate the Logical Keystroke FEE and thus have effects that are under the control of the FEPCOs present in the VTE.

9. A minimum set of local actions is defined within this profile, but implementors may extend this as required. A host implementation thus may not know what local action is being over-ridden when it requests that a particular logical keystroke should be notified to the host. To prevent this from limiting the capabilities of the terminal, two keystroke combinations that differ only in the inclusion or otherwise of the ALT key are required to have the same potential local action. Host implementations are advised not to over-ride the action of both such keystrokes.

   The defined minimum set of local actions concerns control of the current entry location. At any time when the terminal possesses the WAVAR access right, there is a well-defined Display Object array element which is the current candidate for

update by a character entry operation, as described in Informative Note 4. If this element lies outside of any field, or within a protected field, update is prohibited unless the negotiated value of the VTE-parameter access-outside-fields is "yes," but the array element is still defined. Neither this location nor that of any cursors which the implementation may use to indicate such elements is recorded in the CCA. It is separate from the current position of either the display pointer or the logical pointer, and movement of this entry location is a purely local action.

Table 14.2. Local actions that move entry location

Name	Unshifted action	Shifted action
leftArrow	x = x-1	k = k-1
rightArrow	x = x+1	k = k+1
upArrow	y = y-1	f = f-1, k = 1
downArrow	y = y+1	f = f+1, k = 1
home	(x,y,z) = "start-y"	(k,f,z) = "start-k"
end	(x,y,z) = "end-y"	(k,f,z) = "end-k"
pageUp	z = z-1, x = 1, y = 1	z = z-1, f = 1, k = 1
pageDown	z = z+1, x = 1, y = 1	z = z+1, f = 1, k = 1
tab		f = next(f), k = 1
backTab		f = previous(f), k = 1

The names given in the first column of table 14.2 are the identifiers of named integers of type STCO.Key. The ASN.1 module STCO is defined as part of the specification of the Sequenced Terminal Control Object in clause 14.7.3. These identifiers or the corresponding integers are used to designate the local actions specified in the second column. If the initial lower case letter of such a name is converted to upper case and prefixed with "shift" then it designates the local action specified in the third column.

In this table, "=" is used as an assignment operator. The unshifted actions reference array elements by normal (x, y, z) coordinates while the shifted actions reference them by logical (k, f, z) coordinates. The values next(f) and previous(f) are defined in 19.1.3.2.2 of ISO 9040, "start-k" and "end-k" are defined in 19.1.3.5 of

ISO 9040, and "start-y" and "end-y" are defined in 19.1.1.4 of ISO 9040.

If the initial or final coordinate values are undefined then the local action is implementation-dependent. However, a host implementation can use the mandatory FEPCO to control the behavior in such circumstances. Field Entry Conditions are provided to test whether a particular local action would make the entry location leave the current field or navigation path, as defined in clause 14.8.3.3.

10. If the VTE-parameter access-outside-fields takes the value "allowed", when data entry terminates, the display pointer shall be aligned with the current entry location by an explicit or implicit addressing operation. In this way, the value of the display pointer notifies the application of the current entry location.

11. Use of the values "F" (Framed) and "C" (Encircled) for emphasis subattribute "h" causes groups of characters within a single field which have this subattribute value to be outlined by a frame. The two subattributes differ only in that the external corners of the frame are squared if value "F" is used and rounded if value "C" is used. An external corner is where two lines meet in a L shape, as distinct from a T junction and from the intersection of two lines. The nature of the external corner is controlled by the subattribute value of the array element on the inside of the corner.

    More precisely, a character box element is defined to be within frame (f,z) if it is in the field with coordinates (f,z) and has either the framed or encircled attribute. A character box element is defined to be without frame if either it is not in any field or it does not have either the framed or encircled attribute. In the image of a y-array on the real device, a line is drawn between two adjacent images of character box elements if they are within different frames, or if one is within a frame and one is without frame. In addition if a character box element is within some frame, a line is drawn along any edge of that element which is not in common with any other character box

element, i.e., along any edges which are part of the image of the boundary of the Display Object.

14.8.3.6.2 Informative Notes

1. Updates by the application VT-user (only possible within the z-window) are not necessarily immediately imaged to the (human user of the) terminal VT-user unless the real window of the device is currently positioned over such an update. Such updates may move the real window if a VT-DELIVER indication is received.

   When WAVAR is relinquished by the application VT-user the window may be moved so that the field addressed by the CCO is within the window.

   Application VT-user addressing operations that advance z to a higher address which is outside of the z-window cause the z-window to move and include one or more new y-arrays for which no fields are defined. As the z-window moves, one or more y-arrays at lower addresses will no longer be included in the z-window. The field definition records for such y-arrays are implicitly deleted.

2. Several of the descriptions of Field Entry Instructions refer to 'empty' array elements of the Display Object. This is to be interpreted in the sense of 13.2 of ISO 9040. Note that in this sense an array element containing a space character is not empty. The representation of an empty array element on the real device is implementation-dependent, but for this reason it is recommended that the representation used should be distinct from that of a space character.

3. The descriptions of a number of Field Entry Conditions refer to the current field and to the current location for the next character entry. Typically this current location will be indicated to the human user by a visible cursor. When this location lies within a defined field, that field is the current field and the Entry Invoke Character FEI may be used to specify the nature of the visible cursor. However, a terminal implementation may allow the visible cursor to be moved outside of any defined field. While this is so, the representation of the cursor is

implementation dependent, the current field is undefined and no FEPRs are active.

14.8.3.7 Specific Conformance Requirements

For further agreement.

14.8.4 X3 Profile

OIW VTE-Profile X3-1989 (r1, r2, r3, r4, r5, r6)

14.8.4.1 Introduction

This profile provides support for CCITT X.3 PAD compatible operation.

The purpose of this profile is two-fold:

- to provide a transitional environment for applications that assume the availability of X.3 parameters with which to control the behavior of the terminal-system.
- to facilitate a gateway function between ISO-VTP and X.3.

14.8.4.2 Association Requirements

14.8.4.2.1 Functional Units

The Structured CO Functional Unit is mandatory.

The Urgent Data Functional Unit is optional.

14.8.4.2.2 Mode

This is an A-mode profile.

14.8.4.3 Profile Body

```
Display-objects  =
{
     {
     display-object-name = D1,
     DO-access           = profile-argument-r1,
     dimensions          = "one,"
```

```
x-dimension    =
{
     x-bound         = "unbounded,"
     x-addressing    = "not-permitted,"
     x-absolute      = "no,"
     x-window        = 0
},
```

```
     repertoire-assignment    = <ESC> 2/5 2/15 4/2
                              *( VTS Transparent Set )*
     },

     {
     display-object-name = D2,
     DO-access           = opposite of profile-argument-r1,
     dimensions          = "one,"
          x-dimension =
          {
               x-bound        = "unbounded,"
               x-addressing   = "not-permitted,"
               x-absolute     = "no,"
               x-window       = 0
          },
     repertoire-assignment    = <ESC> 2/5 2/15 4/2
                              *( VTS Transparent Set )*
     },
},


Control-objects     =
{
     { *( PAD -
     Each element of the PAD CO represents a CCITT PAD
     parameter.  The CO-element-id of each element has been
     chosen so that it would be same value as the CCITT PAD
     parameter number that it represents.  The PAD CO is
     used both to set CCITT PAD parameter-equivalent values
     and to reply to an update to the READ CO.  See
     Definitive Note 25 for conventions concerning updates
     to this CO. )*
     CO-name         = PAD,
     CO-structure    = 22,
     CO-access       = "NSAC,"
     CO-priority     = "normal,"
     CO-trigger      = "not-selected,"
     {    *( X.3 parameter 1 -- PAD recall )*
          CO-element-id  = 1,
          CO-category    = "transparent,"
          CO-size        = 8 },
     {    *( X.3 parameter 2 -- PAD echo )*
          CO-element-id  = 2,
          CO-category    = "boolean,"
          CO-size        = 1 },
     {    *( X.3 parameter 3 -- Data Forwarding Character )*
          CO-element-id  = 3,
          CO-category    = "boolean,"
          CO-size        = 7 },
```

```
{    *( X.3 parameter 4 -- Idle Timer Delay )*
     CO-element-id  = 4,
     CO-category    = "integer,"
     CO-size        = 255 },
{    *( X.3 parameter 5 -- Ancillary Device Control )*
     CO-element-id  = 5,
     CO-category    = "boolean,"
     CO-size        = 1 },
{    *( X.3 parameter 6 -- Control of PAD Signals )*
     CO-element-id  = 6,
     CO-category    = "transparent,"
     CO-size        = 4 },
{    *( X.3 parameter 7 -- PAD on receipt of Break )*
     CO-element-id  = 7,
     CO-category    = "boolean,"
     CO-size        = 5 },
{    *( X.3 parameter 8 -- Discard Output )*
     CO-element-id  = 8,
     CO-category    = "boolean,"
     CO-size        = 1 },
{    *( X.3 parameter 9 -- Padding After <CR> )*
     CO-element-id  = 9,
     CO-category    = "integer,"
     CO-size        = 7 },
{    *( X.3 parameter 10 -- Line Folding )*
     CO-element-id  = 10,
     CO-category    = "integer,"
     CO-size        = 255 },
{    *( X.3 parameter 11 -- Device Speed )*
     CO-element-id  = 11,
     CO-category    = "symbolic,"
     CO-category    = 19 },
{    *(X.3 parameter 12 -- Flow Control by Device )*
     CO-element-id  = 12,
     CO-category    = "boolean,"
     CO-size        = 1 },
{    *( X.3 parameter 13 -- Insert <LF> after <CR> )*
     CO-element-id  = 13,
     CO-category    = "boolean,"
     CO-size        = 3 },
{    *( X.3 parameter 14 -- Linefeed Padding )*
     CO-element-id  = 14,
     CO-category    = "integer,"
     CO-size        = 7 },
{    *( X.3 parameter 15 -- Editing )*
     CO-element-id  = 15,
     CO-category    = "boolean,"
     CO-size        = 1 },
```

```
{    *( X.3 parameter 16 -- Character Delete )*
     CO-element-id  = 16,
     CO-category    = "character,"
     CO-repertoire-assignment *( any from CO )*
                    = "void," "void," <ESC> 2/1 4/0,
     CO-size        = 1 },
{    *( X.3 parameter 17 -- Line Delete )*
     CO-element-id  = 17,
     CO-category    = "character,"
     CO-repertoire-assignment *( any from CO )*
                    = "void," "void," <ESC> 2/1 4/0,
     CO-size        = 1 },
{    *( X.3 parameter 18 -- Line Display )*
     CO-element-id  = 18,
     CO-category    = "character,"
     CO-repertoire-assignment *( any from CO )*
                    = "void," "void," <ESC> 2/1 4/0,
     CO-size        = 1 },
{    *( X.3 parameter 19 -- Editing Service Signals )*
     CO-element-id  = 19,
     CO-category    = "transparent,"
     CO-size        = 8 },
{    *( X.3 parameter 20 -- Echo Mask )*
     CO-element-id  = 20,
     CO-category    = "boolean,"
     CO-size        = 8 },
{    *( X.3 parameter 21 -- Parity Treatment )*
     CO-element-id  = 21,
     CO-category    = "boolean,"
     CO-size        = 2 },
{    *( X.3 parameter 22 -- Page Wait )*
     CO-element-id  = 22,
     CO-category    = "integer,"
     CO-size        = 256 }
},

{ *( READ -
Each boolean of the READ CO represents an element-id of
the PAD CO with the same identifying value.  The READ
CO is used to request the current values of PAD CO,
which may have been changed by some local agent.  See
the description of the PAD CO for how the update to
this CO modifies the access to the PAD CO. )*
CO-name        = READ,
CO-structure   = 1,
CO-access      = opposite of profile-argument-r1,
CO-priority    = "normal,"
CO-trigger     = "not-selected,"
CO-category    = "boolean,"
CO-size        = 22
```

```
},
{ *( Break Out-of-Band -
receipt of this control object represents "X.25
Interrupt"; use is applicable when boolean 1 of
element-id 7 in PAD CO has the value "true." )*
CO-name         = BO,
CO-structure    = 1,
CO-access       = profile-argument-r1,
CO-priority     = "urgent,"
CO-trigger      = "not-selected,"
CO-category     = "symbolic,"
CO-size         = 1
},

{ *( Break In-Band -
receipt of this control object represents "indication
of break"; use is applicable when boolean 3 of element-
id 7 in PAD CO has the value "true." )*
CO-name         = BI,
CO-structure    = 1,
CO-access       = profile-argument-r1,
CO-priority     = "normal,"
CO-trigger      = "selected,"
CO-category     = "symbolic,"
CO-size         = 1
},
```

```
{ *( CUD -
This CO is used to optionally convey Call User Data
which is normally carried in the CCITT PAD call.  The
CO is not updatable, but may be given initial content
value during association establishment by special
profile arguments r2 and r3.  The CO is parametric,
with two elements, one representing the protocol
identifier field, and the other representing the call
data field containing user data. )*
CO-name         = CUD,
CO-structure    = 2,
CO-access       = "no-access,"
{    *( Protocol Identifier )*
     CO-category     = "character,"
     CO-repertoire-assignment *( VTS Transparent Set )*
                     = <ESC> 2/5 2/15 4/2,
     CO-size         = 4 },
{    *( User Data )*
     CO-category     = "character,"
     CO-repertoire-assignment *(VTS Transparent Set )*
                     = <ESC> 2/5 2/15 4/2,
     CO-size         = 124 }
},

{ *( DTE -
This CO is used to optionally indicate the calling and
called DTE addresses which are normally available in a
true CCITT PAD environment.  They may not be updated,
but may be given initial content values during the
association establishment by special profile arguments
r4 and r5. )*
CO-name         = DTE,
CO-structure    = 2,
CO-access       = "no-access,"
{    *( Calling DTE address )*
     CO-element-id   = 1,
     CO-category     = "character,"
     CO-repertoire-assignment *(VTS Transparent Set )*
                     = <ESC> 2/5 2/15 4/2,
     CO-size         = 15 },
{    *( Called DTE address )*
     CO-element-id   = 2,
     CO-category     = "character,"
     CO-repertoire-assignment *(VTS Transparent Set )*
                     = <ESC> 2/5 2/15 4/2,
     CO-size         = 15 }
},
```

```
        { *( FAC -
        This CO is used to optionally indicate the CCITT
        facilities which are normally negotiable during the
        establishment of a PAD virtual circuit.  The
        negotiation takes place in the VT association
        establishment via special profile argument r6, where
        the initiator may propose the initial content value,
        and the acceptor may return other values. )*
        CO-name          = FAC,
        CO-structure     = 1,
        CO-access        = "no-access,"
        CO-category      = "character,"
        CO-repertoire-assignment *(VTS Transparent Set )*
                         = <ESC> 2/5 2/15 4/2,
        CO-size          =  127
        },
    },

    Device-objects *(double occurrence)* =
    {
        {
        device-name = DEVICE-1,
        device-default-CO-access = profile-argument-r1,
        device-default-CO-priority = "normal,"
        device-default-CO-trigger = "not-selected,"
        device-default-CO-initial-value = 1."true,"
        device-minimum-X-array-length = 1, *(no constraint)*
        device-control-object = { BI, BO, PAD },
        device-display-object = D1
        *(termination parameters are controlled explicitly
        through the values assigned to elements 3 and 4 of the
        PAD Control Object)*
        },
        {
        device-name = DEVICE-2,
        device-default-CO-access =
                              opposite of profile-argument-r1,
        device-default-CO-priority = "normal,"
        device-default-CO-trigger = "not-selected,"
        device-default-CO-initial-value = 1."true,"
        device-minimum-X-array-length = 1, *(no constraint)*
        device-control-object = { READ, PAD },
        device-display-object = D2
        }
    },
    Type of delivery control = "simple-delivery-control."
```

14.8.4.4 Profile Arguments

r1 - is mandatory, and is used to establish the access rules for the display objects and several of the control objects. This argument takes one of the values "WACI" or "WACA." It is identified by the identifier for DO-access for display object D1.

r2 - is an optional special profile argument, and is used to set the intial content value of element 1 of the CUD CO. The value is encoded from the binary form to the ASN.1 type PrintableString according to the algorithm described in Definitive Note 24. This argument is assigned the identifier "Pp-1."

r3 - is an optional special profile argument, and is used to set the initial content value of element 2 of the CUD CO. The value is encoded from the binary form to the ASN.1 type PrintableString according to the algorithm described in Definitive Note 24. This argument is assigned the identifier "Pp-2."

r4 - is an optional special profile argument, and is used to set the initial content value of element 1 of the DTE CO. The value is encoded from the binary form to the ASN.1 type PrintableString according to the algorithm described in Definitive Note 24. This argument is assigned the identifier "Pp-3."

r5 - is an optional special profile argument, and is used to set the initial content value of element 2 of the DTE CO. The value is encoded from the binary form to the ASN.1 type PrintableString according to the algorithm described in Definitive Note 24. This argument is assigned the identifier "Pp-4."

r6 - is an optional special profile argument, and is used to set the initial content value of the FAC CO. The value is encoded from the binary form to the ASN.1 type PrintableString according to the algorithm described in Definitive Note 24. This argument is assigned the identifier "Pp-5."

14.8.4.5 Profile Notes

14.8.4.5.1 Definitive Notes

1. The value assigned to element 1 of PAD CO selects the character used to return control to the terminal-system. The valid values and associated meanings are:

value	meaning
0	not-permitted
1	1/0 character (DLE)
32-126	graphic character

2. The value assigned to element 2 of PAD CO determines whether or not characters are echoed at the terminal-system. When the value of this boolean is "true," then the characters are echoed at the terminal-system.

3. The values assigned to element 3 of PAD CO control the forwarding of characters from the terminal-system to the application-system based on the character value. The defined booleans and associated meanings are:

boolean	meaning
1	alphanumeric (A-Z, a-z, 0-9)
2	character 0/13 (CR)
3	characters 1/11 (ESC), 0/7 (BEL), 0/5 (ENQ), 0/6 (ACK)
4	characters 7/15 (DEL), 1/8 (CAN), 1/2 (DC2)
5	characters 0/3 (ETX), 0/4 (EOT)
6	characters 0/9 (HT), 0/10 (LF), 0/11 (VT), 0/12 (FF)
7	all others in column 0 and 1 not already included above

4. The value assigned to element 4 of PAD CO controls the forwarding of characters from the terminal-system to the application-system based on the duration of idle time elapsed between consecutive characters received by the terminal-system from the device. The valid values include any non-negative integer 0-255; a value between 1 and 255 indicates the time-out in twentieths of a second;

a value of 0 means that a time-out is not a forwarding condition.

5. The value assigned to element 5 of PAD CO determines whether the XON/XOFF flow-control characters (1/1 and 1/3) are available for use by the terminal-system. When the value of this element is "true," then the flow-control characters are available, and the terminal-system may use them to indicate to the device its readiness to accept characters from it.

6. The value assigned to element 6 of PAD CO determines whether the terminal-system issues messages, called PAD service signals, to the device during the association. The specific service signals are not a part of this profile definition, only the control of their issue.

7. The values assigned to element 7 of PAD CO determine the behavior at the terminal-system when a Break is received from the device. The defined booleans and associated meanings are:

boolean	meaning
1	update BO CO
2	release the association
3	update BI CO
4	return control to terminal-system
5	discard data from application-system

When all booleans have the value "false," there is no action at the terminal-system when a Break is received

A useful combination of booleans with value "true" is (1,3,5). When a Break is received, the terminal-system updates both the BO CO and the BI CO and discards all display-object updates from the application-system until it receives an update to the PAD CO for element 8. The result is that the data path has been cleared in both directions. Notice that this is non-destructive of control object updates.

8. The value assigned to element 8 of PAD CO determines whether or not the terminal-system discards data from the application-system. This element works with element 7 to acknowledge the

receipt of the Break and resume normal processing of display-object updates. The only valid value of this boolean in an update is "false."

9. The value assigned to element 9 of PAD CO indicates the number of padding characters to be generated by the terminal-system to the device following a carriage return character. The valid values are integers in the range 0-7.

10. The value assigned to element 10 of PAD CO indicates the number of graphic characters sent to the device after which the terminal-system will insert a carriage return. The valid values are integers in the range 0-255, where a value of 0 means that this function is not performed.

11. The value assigned to element 11 of PAD CO indicates the bit-transmission speed of the device. This element may only appear in an update sent to the application-system in response to an update of the READ CO when boolean 11 has the value "true."

12. The value assigned to element 12 of PAD CO determines whether the XON/XOFF flow-control characters (1/1 and 1/3) are available for use by the device. When the value of this element is "true," then the flow-control characters are available, and the device may use them to indicate to the terminal-system its readiness to accept characters from it.

13. The values assigned to element 13 of PAD CO determine under which situations a linefeed is inserted following a carriage return character. The valid values and associated meanings are:

boolean	meaning
1	insert linefeed after carriage return sent to device
2	insert linefeed after carriage return received from device
3	insert linefeed after carriage return echoed to the device

14. The values assigned to element 14 of PAD CO determine the number of padding characters generated by the terminal-system to the device

following a linefeed character. The valid values are any number in the range 0-7.

15. The value assigned to element 15 of PAD CO determines whether or not the terminal-system performs data-editing. When this CO has value "true," the values of the elements 3 and 4 of the PAD CO are ignored.

16. The value assigned to element 16 of PAD CO determines which character is used in editing the line to signify the function "delete character." The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true."

17. The value assigned to element 17 of PAD CO determines which character is used in editing to signify the function "delete line." The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true."

18. The value assigned to element 18 of PAD CO determines which character is used in editing to signify the function "display line." The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true."

19. The value assigned to element 19 of PAD CO determines whether the terminal-system provides for editing of PAD service signals. The valid values and meanings are as follows:

value	meaning
0	no editing
1	editing as for a paper device
2	editing as for a glass device
8	editing using one editing character
32-126	editing using one editing character

20. The values assigned to element 19 of PAD CO determines which characters are NOT to be echoed to the device by the terminal-system. If no bits are set, then all characters are to be echoed, assuming that element 2 has the value "true." The defined booleans and associated meanings are:

boolean	meaning
1	Do not echo 0/13 (CR)
2	Do not echo 0/10 (LF)
3	Do not echo 0/11 (VT), 0/9 (HT) 0/12 (FF)
4	Do not echo 0/7 (BEL) or 0/8 (BS)
5	Do not echo 1/11 (ESC) or 0/5 (ENQ)
6	Do not echo 0/6 (ACK), 1/5 (NAK), 0/2 (STX), 0/1 (SOH), 0/4 (EOT), 1/7 (ETB) or 0/3 (ETX)
7	Do not echo the editing characters defined by elements 16, 17 and 18 of the PAD CO
8	Do not echo 7/15 (DEL) or any of the other characters belonging to CO or C1 which are not already mentioned above

21. The value assigned to element 21 of PAD CO determines the treatment of parity on the characters received from and sent to the device from the terminal-system. The defined booleans and associated meanings are:

boolean	meaning
1	parity is checked on characters received from the device
2	parity is generated on characters sent to the device

22. The value assigned to element 22 of PAD CO determines the number of linefeeds that the terminal-system may send to the device before it must wait for input from the device request it to continue displaying characters. The range of valid values is 0-255, where a value of 0 indicates that the terminal-system need never wait.

23. The text operation is the only operation allowed on the display objects.

24. Special profile arguments r2-r6 have binary values. However, due to a restriction in the standards 9040 and 9041, those binary values must be conveyed in the ASN.1 type PrintableString. This is accomplished by mapping the value of each semi-octet in the string of binary octets to an octet whose value falls in the value range of a PrintableString. The semi-octet values in the range 0000 - 1001 are mapped into the PrintableString values '0' - '9', whereas the semi-octet values in the range 1010 - 1111 are mapped into the PrintableString values 'A' - 'F'. The result is a string of characters which is exactly twice the length of the original string of binary octets.

25. The value of CO-access for the PAD CO is "NSAC," however a convention is followed that determines when a VT-user may update the PAD CO. Only the VT-user with access to the Display Object D2 may update the PAD CO except immediately after it has updated the READ CO. When the READ CO is received by the opposite VT-user, it is treated as a request to update the PAD CO with the parameter values it is currently using, at which point that VT-user is required to respond.

14.8.4.5.2 Informative Notes

1. Users of this profile should refer to CCITT Recommendations X.3, X.28 and X.29 for the original model for this profile.

2. The following values for the elements of the PAD CO are taken from the CCITT Simple standard profile and may prove useful:

element-id	Value
1	1 - possible to return control to the terminal-system using 0/1 (DLE)
2	1."true" - echo performed at the terminal-system
3	1."false," 2."true," 3."true," 4."true," 5."true," 6."true," 7."true" - forward on receipt of any character in C0 and C1
4	0 - no time-out used for forwarding condition
5	1."true" - terminal-system use XON/XOFF to flow-control the device
6	1."true" - service signals are sent
7	2."true," all others "false" - release the association when a Break is received from the device
8	1."false" - deliver data to device
9	0 - do not pad after CR

10	0 - do not fold the line
11	- read-only
12	1."true" - device use XON/XOFF to flow-control the terminal-system
13	0 - do not insert LF after CR
14	0 - do not pad after LF
15	1."false" - do not edit data
16	7/15 (DEL) - character delete
17	1/8 (CAN) - line delete
18	1/2 (DC2) - line display
19	1 - edit as for paper
20	0 - echo all characters
21	0 - no parity checking or generation
22	0 - no page wait

3. The following values for the elements of the PAD CO are taken from the CCITT Transparent standard profile and may prove useful.

element-id	Value
1	0 - control may not be returned to the terminal-system
2	1."false" - the terminal-system does not perform character echo
3	all booleans "false" - no forwarding on character value
4	20 - forward on time-out of 1 second

5	1."false" - terminal-system may not flow-control device
6	1."false" - service signals are never sent
7	2."true," all others "false" - release the association
8	1."false" - deliver data to device
9	0 - no pad after CR
10	0 - no line folding
11	- read-only
12	1."false" - device may not flow-control terminal-system
13	0 - no LF insert after CR
14	0 - no pad after LF
15	1."false" - no editing data
16	7/15 (DEL) - character delete
17	1/8 (CAN) - line delete
18	1/2 (DC2) - line display
19	1 - edit as for paper
20	0 - echo all characters
21	0 - no parity checking or generation
22	0 - no page wait

14.8.4.6 Specific Conformance Requirements

None.

14.9 APPENDIX A

14.9.1 Specific ASE Requirement

For specific ASE Requirements identified by the Upper Layer SIG for Virtual Terminals, see chapter 5, entitled "Upper Layers" in this document.

14.10 APPENDIX B - CLARIFICATIONS

14.10.1 Defaults

When a profile argument is not present in either the offer or value list, the default for the corresponding VTE parameter is specified by ISO 9040 if it is not given by the argument description in the profile.

14.11 APPENDIX C - OBJECT IDENTIFIERS

General identifiers:

```
oiw-vt          OBJECT IDENTIFIER ::=
     { iso(1) identified-organization(3) oiw(14) vtsig(12) }

oiw-vt-pr       OBJECT IDENTIFIER ::=
     { oiw-vt       vteProfile(1) }

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

oiw-vt-co-misc OBJECT IDENTIFIER ::=
     { oiw-vt-co    cotypemisc(0) }

oiw-vt-co-tcco OBJECT IDENTIFIER ::=
     { oiw-vt-co    cotypetcco(4) }
```

Profile defined by OIW VT SIG:

```
oiw-vt-pr-telnet-1988          OBJECT IDENTIFIER ::=
     { oiw-vt-pr    telnet-1988(0) }

oiw-vt-pr-transparent-1988     OBJECT IDENTIFIER ::=
     { oiw-vt-pr    transparent-1988(1) }

oiw-vt-pr-forms-1989           OBJECT IDENTIFIER ::=
     { oiw-vt-pr    forms-1989(2) }
```

```
oiw-vt-pr-x3-1989              OBJECT IDENTIFIER ::=
     { oiw-vt-pr    x3-1989(4) }
```

Control Objects defined by OIW VT SIG:

```
oiw-vt-co-misc-sa              OBJECT IDENTIFIER ::=
     { oiw-vt-co-misc    sa(0) }

oiw-vt-co-misc-ua              OBJECT IDENTIFIER ::=
     { oiw-vt-co-misc    ua(1) }

oiw-vt-co-misc-st              OBJECT IDENTIFIER ::=
     { oiw-vt-co-misc    st(2) }

oiw-vt-co-misc-ut              OBJECT IDENTIFIER ::=
     { oiw-vt-co-misc    ut(3) }
```

15. TRANSACTION PROCESSING

Editor's Note: This section is a placeholder for future Transaction Processing (TP) Agreements. ~~The TP Special Interest Group is newly formed and held its first regular meeting in March, 1989.~~ Any new text from this group will be inserted here.

16. OFFICE DOCUMENT ARCHITECTURE

Editor's Note: There is international alignment work taking place which is likely to produce an International Standardized Profile (ISP) corresponding to the level of ODA functionality being addressed by these agreements. The Plenary, in December 1989, approved a resolution that this DAP should be superseded by the equivalent internationally aligned ODA DAP when it is submitted for processing as an ISP. The intent is to rename this chapter "OFFICE DOCUMENT ARCHITECTURE LEVEL 3 DAP." The contents herein were last modified by the Plenary in June 1989.

16.1 INTRODUCTION

This is the definition of an ODA document application profile (AP) named NIST Level 3 DAP. The NIST Level 3 DAP is suitable for interchanging a document in formatted form, processable form or formatted processable form. This DAP has been prepared by the ODA Special Interest Group of the National Institute of Standards and Technology (NIST) Workshop for Implementors of OSI. The DAP is defined in accordance with ISO 8613-1 and CCITT T.411 and follows the standardized proforma and notation defined in ISO 8613-1 proposed Draft Addendum (to be published).

Note: The agreements defining this ODA DAP will be superseded by the equivalent internationally aligned ODA DAP when it is submitted for processing as an International Standardized Profile (ISP).

16.2 SCOPE AND FIELD OF APPLICATION

This DAP specifies interchange formats for the transfer of structured documents between equipment designed for word or document processing. Such documents may contain characters, raster graphics and geometric graphics content.

The documents supported by 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 document application profile describes documents which can be interchanged in the following form, as defined in ISO 8613:

- Formatted form,

- Processable form, and
- Formatted processable form.

The architecture level have matching functionalities so that the interchange formats of a document are convertible from a processable form into any other form.

This DAP is independent of the processes carried out in an end system to create, edit or reproduce which, for example, may be by means of communication links or storage media.

16.3 REFERENCES

ISO 2022 Information Processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques

ISO 6937-1 Information Processing - Coded character sets for text communication - Part 1: General introduction

ISO 6937-2 Information Processing - Coded character sets for text communication - Part 2: Latin alphabetic and non-alphabetic graphic characters

ISO 8613-1 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 1: Introduction and General Principles

ISO 8613-2 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 2: Document Structures

ISO 8613-4 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 4: Document Profile

ISO 8613-5 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 5: Office Document Interchange Format

ISO 8613-6 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 6: Character Content Architecture

ISO 8613-7 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 7: Raster Graphics Content Architecture

ISO 8613-8 Information Processing - Text and Office Systems - Office

Document Architecture (ODA) and Interchange Format - Part 8: Geometric Graphics Content Architecture

ISO 8613-1 PDAD ... "Document Application Profile Proforma and Notation" (to be published)

ISO 8632-1 Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture description information - Part 1: Functional Specification

ISO 8632-3 Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture description information - Part 3: Binary Encoding

ISO 8859-1 Information Processing - 8-bit single byte coded graphic character sets - Part 1: Latin Alphabet No. 1

ISO 8859-7 Information Processing - 8-bit single byte coded graphic character sets - Part 7: Latin/Greek Alphabet

ISO 8824 Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation 1 (ASN.1)

ISO 8825 Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation 1 (ASN.1)

CCITT T.6 - Facsimile coding scheme and coding control functions for Group 4 Facsimile Apparatus, 1984

CCITT T.411 Open Document Architecture (ODA) and Interchange Format - Introduction and general principles, 1988

CCITT T.502 Document Application Profile PM.1 for the interchange of processable form documents

PrENV ... Q111 ODA document application profile - processable and formatted documents - basic character content (to be published)

PrENV ... Q112 ODA document application profile - processable and formatted documents - extend mixed mode (to be published)

INTAP ... AE-1126 ODA document application profile ...

PAGODA ... CORE-11 ODA document application profile - processable and formatted documents - basic character content (to be published)

PAGODA ... CORE-26 ODA document application profile - processable and formatted documents - advanced mixed mode (to be published)

PAGODA ... Core-36 ODA document application profile - processable and formatted document - enhanced mixed mode (to be published)

16.4 DEFINITIONS AND ABBREVIATIONS

The definitions given in ISO 8613-1 are applicable to this document.

The following additional definitions are applicable to this document.

Generating Support Statement (GSS)

> A statement which states the range of support of an originating system. An originating system generates ODIF data streams. A GSS defines a subset of all possible data streams supported by an implementation which an origination capability. A GSS is specified by completing the GSSP defined in Annex A of this document.

Generating Support Statement Proforma (GSSP)

> A definition of the conformance requirements of a profile in terms of a list of requirements for implementations to originate data streams which conform to the profile. A GSSP defines the format for all GSSs.

Implementation Characteristic Statement (ICS)

> A statement which states the range of support of an implementation to a DAP.

Receiving Support Statement (RSS)

> A statement which states the range of support of a receiving system. A receiving system interprets ODIF data streams. A RSS defines functions and fall-backs supported by an implementation with a reception capability. A RSS is specified by completing the RSSP defined in Annex A of this document.

Receiving Support Statement Proforma (RSSP)

> A definition of the conformance requirements of a profile in terms of a list of requirements, including fall-backs, for implementations to receive data streams which conform to the profile. A RSSP defines the format for all RSSs.

16.5 POSITION OF THIS DAP IN THE TAXONOMY OF RELATED DAPS

There is only one DAP currently being developed by the NIST ODA SIG. This DAP is a Level 3 DAP, in relation to the following hierarchy.

Level 1 Basic documents consisting of character content only. This level of document contains the functionality found in most word processor products.

Level 2 Extended documents consisting of character, raster graphics and geometric graphics content. This level of document contains enhanced functionality to support multi-media document processors evolved from word processor precursors.

Level 3 Advanced documents consisting of character, raster graphics and geometric graphics content. This level of document contains the functionality traditionally found in most "desktop publishing" or personal publishing products.

16.6 CONFORMANCE

In order to conform to this DAP, a data stream representing a document must meet the requirements specified in clause 16.5.1.

Clause 16.5.2 specifies the requirements for implementations that originate and/or receive data streams conforming to this DAP.

16.6.1 Data stream conformance

The following requirements apply to the encoding of data streams that conform to this ISP.

- The data stream shall be encoded in accordance with the ASN.1 encoding rules defined in ISO 8825,
- The data stream shall be structured in accordance with the interchange format defined in clause 16.8 of this DAP,
- The encoded document shall be structured in accordance with one of the document architecture classes specified in clause 16.7 of this DAP. In addition, the encoded document shall contain all required constituents specified for that class and contain only constituents permitted or required for that class as specified in clause 16.7 of this DAP,

- The encoded constituents shall contain all required attributes as specified in clause 16.7 of this DAP,

- The encoded attributes shall have values within the range of permissible values specified in clause 16.7 of this DAP,

- The encoded document shall be structured in accordance with the abstract document architecture defined in ISO 8613,

- The encoded document shall be structured in accordance with the characteristics defined in clause 16.6 of this DAP.

16.6.2 Implementation conformance

This clause 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 Generator Support Statement (GSS) and/or Receiver Support Statement (RSS) Proforma as defined in Annex A of this DAP.

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 RSS defined in accordance with Annex A of this DAP.

16.7 CHARACTERISTICS SUPPORTED BY THIS DAP

16.7.1 Overview

The NIST Level 3 DAP, in accordance with ISO 8613, allows documents to be represented in the following forms:

- processable form, which facilitates the revision of a document by a recipient;
- formatted form, which facilities the reproduction of a document as intended by the originator;

- formatted processable form, which facilitates the

reproduction of a document as intended by the originator or facilitates the revision of a document.

16.7.1.1 Specification of constituents

This section specifies the required and optional constituents used for the representation of documents that conform to the NIST Level 3 DAP.

Constituents specified as 'required' must occur in any document that conforms to the NIST Level 3 DAP. Constituents listed as 'optional' may or may not be present in the document depending upon the requirements of the particular document.

16.7.1.2 Formatted form documents

Required Constituents

- a document profile
- layout object descriptions representing a specific layout structure

Optional Constituents

- layout object class descriptions representing a 'partial' generic layout structure
- presentation styles

16.7.1.3 Processable Form Documents

Required Constituents

- a document profile
- logical object class descriptions representing a 'complete' generic logical structure
- logical object descriptions representing a specific logical structure

Optional Constituents

- layout object class descriptions representing a 'complete' generic layout structure

- layout styles

- presentation styles

16.7.1.4 Formatted Processable Form Documents

Required Constituents

- a document profile

- logical object class descriptions representing a 'complete' generic logical structure

- logical object descriptions representing a specific logical structure

- layout object class descriptions representing a 'complete' generic layout structure

- layout object descriptions representing a specific layout structure

Optional Constituents

- layout styles

- presentation styles

The following sections describe the logical and layout features that can be represented in documents conforming to this document application profile. The features are described in terms that are typical of the capabilities of current document processors. The features are grouped into logical features and layout features in order to relate them to their ODA representation.

Documents conforming to this document application profile may contain any or all of the following content architectures:

a) character text,
b) raster graphics, and
c) geometric graphics.

16.7.2 Logical characteristics

16.7.2.1 Document logical structure

The logical structure of documents comprise a title, passages, numbered segments (e.g., chapters, sections or numbered paragraphs), paragraphs, phrases, figures, footnotes and references. Numbered segments can be nested and automatic numbering systems are provided for.

The logical structure of a document conforming to this document application profile consists of a hierarchy of logical objects. The following is an example of a generic document logical structure derived from this document application profile:

```
Document
     Passage(s)
          Paragraph
               Text
               Footnote
                    Footnote reference
                    Footnote body
               Text
               Figure
               Text
          Figure
          Numbered Segment
               Number
               Title
               Passage
                    Paragraph
                    Figure
          Numbered Segment
          ....
```

16.7.2.2 Document structure elements

16.7.2.2.1 Document

A document is composed of a sequence of numbered segments or passages each of which is optionally titled and consists of a sequence of paragraphs and/or figures and/or further passages or numbered segments.

16.7.2.2.2 Passage

A passage consists of any logical sequence of paragraphs, figures, and/or further passages or

numbered segments that can be regarded as an entity for reading or for layout presentation.
For example, separate passages may be provided for:

a) the contents to be placed on the title page of a report,
b) the body of the report, and
c) the contents to be placed in appendices.

A table is a particular case of a passage. A single paragraph or a single figure is a simple case of a passage.

16.7.2.2.3 Numbered Segment

A segment is a part of a document which has an automatic number which precedes any other contents and which serves to uniquely identify the numbered segment.

The contents of a numbered segment may begin with a segment title starting on the same line as the segment number.

The document originator may define different classes of numbered segments having in common some presentation features and/or some layout features. For example, the document originator may define a class of numbered segments which always begin on a new page, and another class of numbered segments which are laid out using a special left or right margin offset.

An automatically generated segment number consists of either a number or a series of numbers separated by instances of an arbitrary specified character string separator. In the case of a series of numbers, the segment number is equal to the automatically generated segment number, if any, of the enclosing segment followed by a single index number to uniquely identify the segment.

Index numbers are generated sequentially within any numbered segment. The method of numbering may be a combination of the following:

a) Arabic numerals,
b) Upper/lower case letters, or
c) Upper/lower case Roman numerals.

16.7.2.2.4 Paragraph

A paragraph is a contiguous amount of content in the intended reading order. For example, a paragraph may contain a mixture of embedded phrases, figures, footnote references and character text.

A paragraph may contain zero, one or more embedded phrases.

A paragraph may contain zero, one or more embedded figures and, optionally, figure references. Multiple consecutive figures and/or figure references, without intervening text, are permitted.

A paragraph may contain zero, one or more embedded footnote references. Multiple consecutive footnote references, without intervening text, are also permitted.

A paragraph may comprise a number of character sequences concatenated together, for example if the character sequences were separately derived or generated.

The document originator may define different classes of paragraphs having in common some presentation features and/or some layout features. For example, the document originator may define classes of paragraphs for "abstract," "standard paragraph," and "summary."

16.7.2.2.5 Phrase

A phrase is an amount of text content that is intended to be distinguished. The phrase can be used to construct a paragraph containing sections of text with different presentation features. For example, in the midst of a paragraph a phrase may be highlighted to distinguish a quote.

A phrase can consist of one or more text content, references, footnotes or other phrases.

16.7.2.2.6 Figure

A figure is an amount of geometric graphics or raster graphics content designed to occupy a rectangular area.

One or more paragraphs can be associated with a figure, for example to provide captions or notes.

16.7.2.2.7 Footnote

A footnote consists of a footnote reference and a footnote body.

The footnote body is a contiguous amount of text that can be read out of sequence from the paragraph containing a reference to it.

16.7.2.2.8 Footnote reference

A footnote reference may have an automatically generated label or one supplied by the user. (Both types of footnote references may be present.) If the label is automatically generated then the label may be represented by 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 non-negative value at the beginning of the document and reset at any segment or passage, as required.

16.7.2.2.9 Reference

A general purpose reference mechanism is provided within paragraph. For example, this reference may be used to reference figures, page numbers, chapter numbers, etc. in other parts of the document.

16.7.3 Layout characteristics

16.7.3.1 Document layout structure

The page layout structure allows for several pagesets (e.g., for title page, foreword and annexes in addition to the main text body, which could be in chapters or sections) The body area of a page may contain multiple columns and areas for graphics. Header and footer contents can also include figures.

The following is an example of a generic document layout structure derived from this document application profile:

```
Document
     Page set
          Page
               Header area
               Body area
                    Single frame
                    Multiple columns
                    Individual frame(s)
                    Mixed set of frames
               Footer area
               ....
```

16.7.3.2 Document layout structure elements

16.7.3.2.1 Document

A document consists of a sequence of one or more page sets.

16.7.3.2.2 Page set

The pages within a page set all have the same dimensions and orientation (landscape or portrait) but may differ in layout and/or content of the header and footer areas.

There may be an optional first page of one particular page layout and this may be followed by either of the following:

a) Repeated pages with the same layout
b) Repeated pages designed for alternating recto and verso layout

16.7.3.2.3 Page layout

This document application profile supports various page dimensions. The default dimensions is the common assured reproduction area of ISO A4 and North American Letter. In addition, support is provided for the assured reproduction area of ISO A4 and North American Letter, as well as, the full page dimensions of ISO A4 and North American Letter. Both portrait and landscape orientations are supported.

A page layout consists of:

a) An optional header area that is reserved for header contents,
b) A single body area, or
c) An optional footer area that is reserved for footer contents.

Particular header and footer contents are associated with each page layout.

16.7.3.2.4 Body area layout

The body area may be subdivided into rectangular frames. Thus the layout may consist of any sequence of:

a) Single frame of fixed width, equal or less than body area width, and fixed height or height adjustable to fit contents,

b) Set of multiple column frames of fixed widths per column and fixed height or height adjustable to fit contents,

c) Individual frames with fixed position and dimensions, potentially overlaid, allowing for support of forms,

d) Mixed set of frames with various properties, e.g., fixed-size figure frame with fixed-sized caption frame beneath and adjustable height text frame beside both.

See figure 16.1 for illustrations.

Frames which have fixed position and dimensions are permitted to overlap.

16.7.3.2.5 Header area layout

This is a rectangular area above the body area. It may be subdivided into a number of rectangular frames, for example to contain textual information and graphics such as company logos.

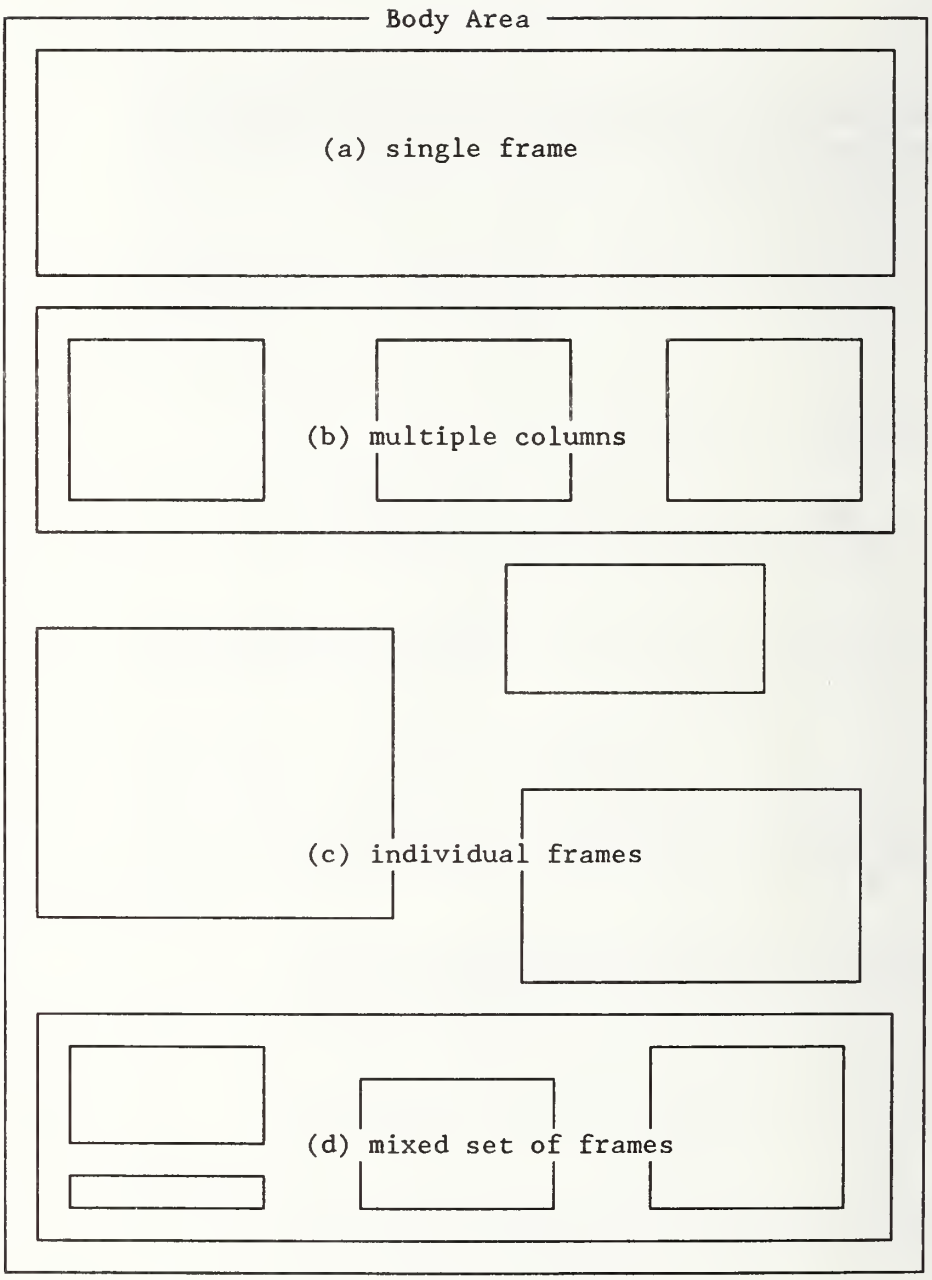


Figure 16.1. Examples of layout within body area.

16.7.3.2.6 Footer area layout

This is a rectangular area below the body area. It may be sub-divided into a number of rectangular frames, for example to contain textual information and graphics such as company logos.

16.7.3.2.7 Header contents and footer contents

Header contents or footer contents may consist of a sequence of paragraphs and/or figures that are constrained to be laid out entirely within the corresponding header or footer area.

One or more automatically generated page numbers may be included anywhere within header contents and/or footer contents.

Header contents or footer contents must not include any footnote or footnote reference or automatic segment numbers.

16.7.3.2.8 Page numbering

An automatically generated page number may occur at any position within header contents or footer contents. Page numbers may represented in Arabic numerals, lower/upper case Roman numerals or lower/upper case letters.

Page numbers are generated sequentially and the sequence can be restarted from any positive integer value at the beginning of any page set.

16.7.3.2.9 Layout of document logical contents

The sequence of passages and/or segments is laid out in one or more body areas such that it flows through the sequence of pages in the document.

Controls are needed in order to break the flow of contents at appropriate points. For example, following the passages to be placed on the title page of a document it may be required to control the flow in order to direct subsequent text onto a new page of a different page layout.

16.7.3.2.10 Layout of passage (or segment) contents

A passage may be laid out in any of the following ways:

a) As separate passages (see below),
b) Below the previous text within a containing passage, or

c) As a sequence of passages.

16.7.3.2.11 Layout of passage contents

Controls are available to guide the layout of passages or their subordinate paragraphs and figures.

A passage can be positioned at a fixed position (e.g., the start) of a new body area or in a new frame below the previous contents of a body area.

In case of sets of multiple columns, content generally flows from the bottom of one column of the set to the top of the next column to the right.

Regardless of content type, the various paragraphs and figures in a passage may be laid out within specified frames.

The various methods of subdivision of body areas may be combined with certain frames being designated for flowing text and other frames for particular contents. Thus text may appear to flow around other contents. For example, several figures can be contained within a passage and effect of text flow around the figures and their captions can be produced. See figure 16.2 for illustration.

A new set of multiple frames can occur beneath a similar set. Thus parallel text (e.g., multilingual) can be synchronized or a table effect can be generated. See figure 16.3 for illustration.

A variation of the table technique can be used for labelling and annotating paragraphs.

A complete passage can be constrained to be contained in the same body area or frame (by indivisibility).

16.7.3.2.12 Layout controls

The following properties may be specified to control where body area or page breaks occur:

a) New column set (New Layout Object)

This specifies that the contents should be laid out in

the first column (or frame) of a new set of columns (or frames).

b) Unconditional column break (New Layout Object)

This indicates that the contents must be displayed in the next column (or frame).

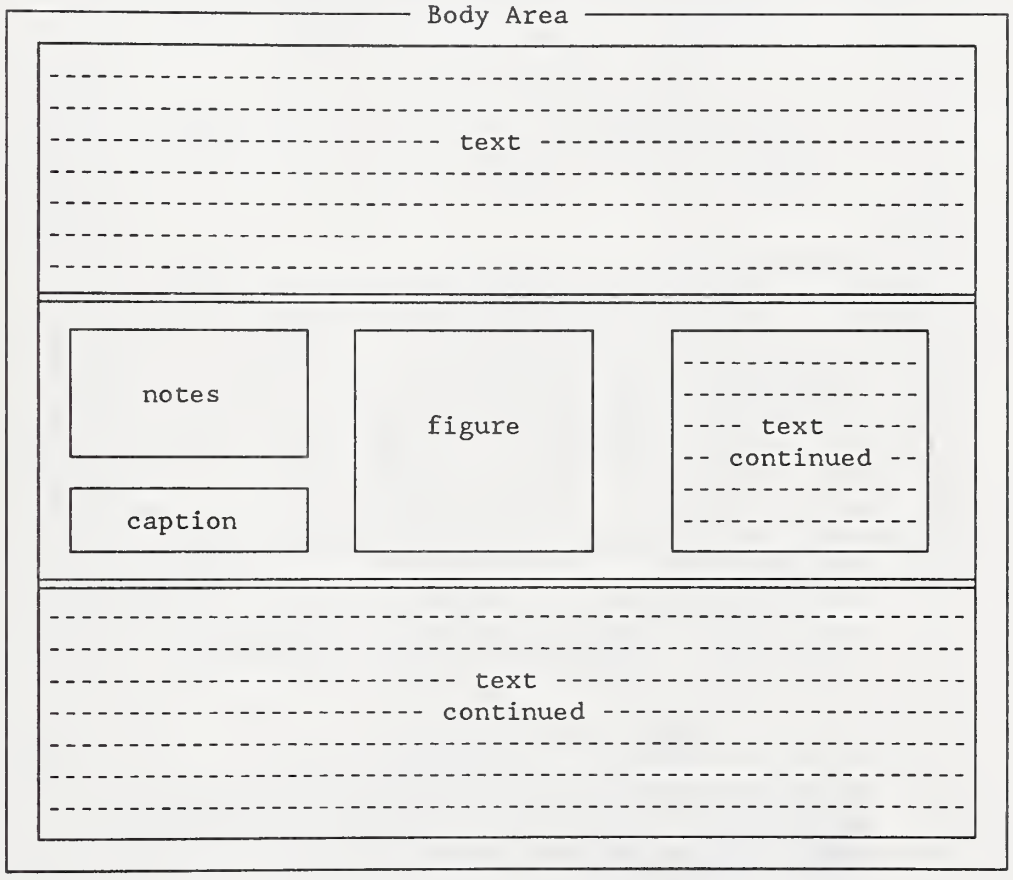


Figure 16.2. Example of text flow around figure.

Note: "Caption" and "Notes" contain formatted character content.

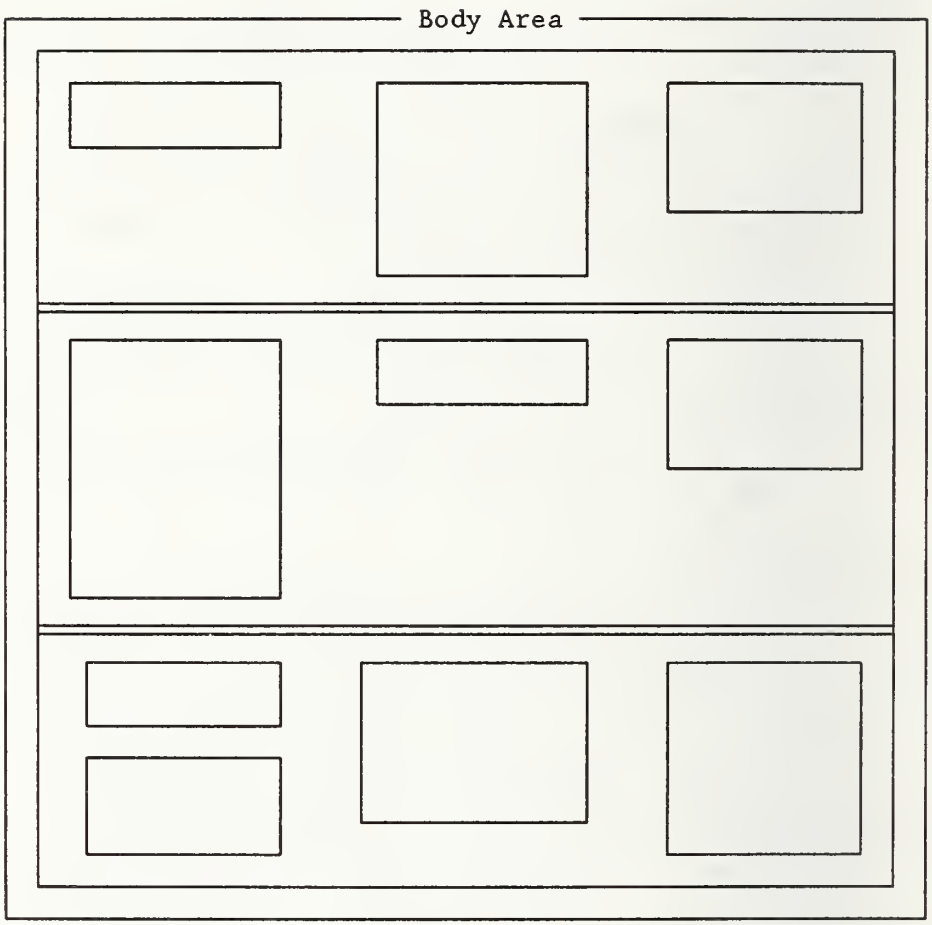


Figure 16.3. Example of synchronized text.

c) Layout object class

This indicates that the contents concerned must be displayed in a specified frame, e.g., to control figure positioning.

d) New page set (New Layout Object)

This indicates that the contents should be laid out in a new page set.

e) New page layout (New Layout Object)

This indicates that the contents should be laid out on a new page of a particular page layout.

f) Unconditional page break (New Layout Object)

This indicates that the contents must be displayed in the body area of the next page.

g) Indivisibility

This indicates that a passage (segment, paragraph or figure) must be laid out within a single frame, body area or page set.

h) Same page/same area (Same Layout Object)

This specifies that the start of a passage, numbered segment, paragraph, or figure, for example, must be laid out in the same frame or body area as the end of the previous content (for example, to keep a first paragraph with a title)

16.7.3.2.13 Layout of paragraph contents

A paragraph may or may not specify its own margins, alignment and tab stops. The indentation of the first line may be different from the remainder of the paragraph. The separation between successive paragraphs can be controlled.

Within a passage the contents of a paragraph may be laid out in two or more frames to allow text to flow around a figure. The figure may or may not be logically associated with that paragraph.

By using the widow and orphan features, layout of paragraphs can also be controlled in order to determine how a paragraph should be divided across two pages.

The orphan size specifies the minimum number of lines of text that must be allocated to the first body area or frame.

The widow size specifies the minimum number of lines of text that must be allocated to the last body area or frame when a paragraph is split over two or more body areas or frames.

16.7.3.2.14 Layout of figure contents

A figure may occur beneath the previous contents of a body area or frame or can be specified to occupy a particular frame.

Any paragraphs associated with the figure, for example, to provide captions or notes, can be positioned to occupy rectangular areas positioned above, below or beside the figure.

16.7.3.2.15 Layout of footnote contents

A footnote body may be placed at the bottom of a body area of a page and may or may not be constrained to be entirely in the same body area as the reference to it. If multiple footnotes occur in the same body area the corresponding footnote bodies can be placed in the body area in the same order as the reading order of their references.

16.7.4 Content characteristics

A document may contain any of the following types of content:

a) Character text utilizing graphic characters as defined in ISO 6937 and ISO 8859 together with character presentation techniques defined in ISO 8613-6,

b) Raster graphics images encoded according to CCITT Recommendations T.4 and T.6 and in unencoded bitmap form,

c) Geometric graphics images in accordance with the minimum capabilities of ISO 8632, each figure consisting of a single picture only.

16.7.4.1 Character content

The document may contain character content portions as specified in ISO 8613-6.

16.7.4.1.1 Character repertoire

The character sets supported by this DAP include ISO 6937-2, ISO 8859-1 and ISO 8859-7.

The character subrepertoires of ISO 6937-2 supported by this DAP include the:

- Minimum subrepertoire (2),
- Subrepertoire of graphic character for teletex (3),
- Unique grapics characters allocated to ISO 646 (5), and
- Western European data processing and interchange or ISO 8859-1 (8).

16.7.4.1.2 Character presentation

Character presentation is controlled by the presentation attributes specified in ISO 8613-6.

Fonts may be specified and selected at any point in the text. Up to ten different fonts can be selected within a given portion of content.

16.7.4.1.3 Character set features and control functions

The effects of graphic rendition can be changed at any point within the text of a paragraph.

Sequences of characters within a piece of contiguous text may be subscripted or superscripted.

Text can be aligned with specific tabulation stops. Text strings and their justification can be terminated by a required newline and can be word-wrapped within the paragraph margins.

Non-breaking spaces and discretionary hyphens are supported in processable and formatted processable form content.

16.7.4.2 Raster graphics content

The document may contain raster graphics or image content portions as specified in ISO 8613-7.

16.7.4.3 Geometric graphics content

The document may contain geometric graphics content portions as specified in ISO 8613-8.

16.7.5 Miscellaneous features

16.7.5.1 Resources

Structure components (generic or specific logical or layout) as well as content portions (character, raster or geometric graphics) of documents may be located in "resource" document external to a particular document.

16.7.6 Document management features

A document profile is associated with the document to provide information to handle it as a whole.

The features specified by the document profile may contain those document management attributes defined in ISO 8613 part 4.

Document profile character sets may use ISO 6937-2 or ISO 8859-1.

16.8 SPECIFICATION OF CONSTITUENT CONSTRAINTS

16.8.1 Document profile

16.8.1.1 Macro Definitions

```
DEFINE(CHAR-SET-LIST,"
                -- ISO 8859-1 Primary Set as G0 --
                            {2/8 4/2, LS0} |
                -- ISO 8859-1 Supplementary Set as G2
                            {2/14 4/1, LS2R} |
                -- ISO 6937-2 Primary Set as G0 --
                            {2/8 4/0, LS0} |
                -- ISO 6937-2 Supplementary Set as G2 --
                            {2/14 4/10, LS2R}
                -- ISO 8859-7 Supplementary Set as G2 --
                            {2/14 4/12, LS2R}   ")
DEFINE(NON-BASIC-PAG-DIM,"
-- Assured Reproduction Areas --

-- Common North American Letter And ISO A4 Landscape --
```

```
   #horizontal <= 12400, #vertical <= 9240,
-- North American Letter Landscape --
   #horizontal <= 13200, #vertical <= 9240,
-- North American Legal Portrait --
   #horizontal <= 9240, #vertical <= 12400,
-- North American Legal Landscape --
   #horizontal <= 12400, #vertical <= 9240,
-- ANSI B Portrait --
   #horizontal <= 12520, #vertical <= 19560,
-- ANSI B Landscape --
   #horizontal <= 19560, #vertical <= 12520,
-- ISO A4 Landscape --
   #horizontal <= 13200, #vertical <= 9240,
-- ISO A3 Portrait --
   #horizontal <= 13200, #vertical <= 18480,
-- ISO A3 Landscape --
   #horizontal <= 18480, #vertical <= 13200,
-- ISO A2 Portrait --
   #horizontal <= 18480, #vertical <= 26040,
-- ISO A1 Portrait --
   #horizontal <= 26040, #vertical <= 36960,
-- ISO A0 Portrait --
   #horizontal <= 36960, #vertical <= 52080,

-- Full Page Sizes --

-- North American Letter Portrait --
   #horizontal <= 10200, #vertical <= 13200,
-- North American Letter Landscape --
   #horizontal <= 13200, #vertical <= 10200,
-- North American Legal Portrait --
   #horizontal <= 10200, #vertical <= 16800,
-- North American Legal Landscape --
   #horizontal <= 16800, #vertical <= 10200,
-- ANSI B Portrait --
   #horizontal <= 13200, #vertical <= 20400,
-- ANSI B Landscape --
   #horizontal <= 20400, #vertical <= 13200,
-- ISO A4 Portrait --
   #horizontal <= 9920, #vertical <= 14030,
-- ISO A4 Landscape --
   #horizontal <= 14030, #vertical <= 9920,
-- ISO A3 Portrait --
   #horizontal <= 14030, #vertical <= 19840,
-- ISO A3 Portrait --
   #horizontal <= 19840, #vertical <= 14030     ")

DEFINE(NON-BASIC-NOM-PAG-SIZ,"

-- North American Letter Landscape --
```

```
   #horizontal <= 13200, #vertical <= 10200,
-- North American Legal Portrait --
   #horizontal <= 10200, #vertical <= 16800,
-- North American Legal Landscape --
   #horizontal <= 16800, #vertical <= 10200,
-- ANSI B Portrait --
   #horizontal <= 13200, #vertical <= 20400,
-- ANSI B Landscape --
   #horizontal <= 20400, #vertical <= 13200,
-- ISO A4 Landscape --
   #horizontal <= 14030, #vertical <= 9920,
-- ISO A3 Portrait --
   #horizontal <= 14030, #vertical <= 19840,
-- ISO A3 Landscape --
   #horizontal <= 19840, #vertical <= 14030     ")

DEFINE(FDA,"      0")
DEFINE(PDA,"      1")
DEFINE(FPDA,"          2")
DEFINE(DAC,"
Document-profile{#Document-characteristics
 {#Document-architecture-class}}")

DEFINE(CF,"       {2 8 2 6 0}")
DEFINE(CP,"       {2 8 2 6 1}")
DEFINE(CFP,"      {2 8 2 6 2}")
DEFINE(RFP,"      {2 8 2 7 2}")
DEFINE(GFP,"      {2 8 2 8 0}")

DEFINE(PARTIAL,"  0")
DEFINE(COMPLETE," 1")
DEFINE(PRESENT,"  1")
```

16.8.1.2 Document profile constraints

16.8.1.2.1 Presence of document constituents

```
CASE    (($DAC) OF

$FDA:
PERM    Generic-layout-structure      {$COMPLETE};
REQ     Specific-layout-structure     {$PRESENT};
PERM    Presentation-styles           {$PRESENT};

$PDA:
PERM    Generic-layout-structure      {$COMPLETE};
REQ     Generic-logical-structure     {$COMPLETE};
```

```
REQ     Specific-logical-structure    {$PRESENT};
PERM    Layout-styles                 {$PRESENT};
PERM    Presentation-styles           {$PRESENT};

$FPDA:
REQ     Generic-layout-structure      {$COMPLETE};
REQ     Specific-layout-structure     {$PRESENT};
REQ     Generic-logical-structure     {$COMPLETE};
REQ     Specific-logical-structure    {$PRESENT};
PERM    Layout-styles                 {$PRESENT};
PERM    Presentation-styles           {$PRESENT};
}

PERM    External-document-class       {ANY};
PERM    Resource-document             {ANY};
PERM    Resources                     {ANY};
```

16.8.1.2.2 Document characteristics

```
REQ     Document-application-profile  {-- ASN.1 object identifier to be
                                      supplied --};

REQ     Doc-appl-profile-defaults     {
{  #Document-architecture-defaults    {

CASE    {($DAC) OF
$FDA:
   #Content-architecture-class   {$FC},
$PDA:
   #Content-architecture-class   {$PC},
$FPDA:
   #Content-architecture-class   {$FPC},
}

-- Common Assured Reproduction Area of --
-- North American Letter Portrait and ISO A4 Portrait --
   #Page-dimensions              {9240, 12400},

-- Nominal Page Size NAL Portrait, "unspecified" --
   #Medium-type                  {10200,13200,0} },

   #Character-contents-defaults  {
-- ISO 8859-1 subrepertoire --
   #Graphic-char-subrepertoire   {8} } };

REQ     Document-architecture-class   {ANY};
REQ     Content-architecture-class    {ANY};
REQ     Interchange-format-class      {if-a};
REQ     ODA-version                   {"ISO 8613," "1989-02-08"}·
```

```
REQ     Non-basic-doc-characteristics {
{  #Profile-character-sets          {$CHAR-SET-LIST};
   #Comment-character-sets          {$CHAR-SET-LIST};
   #Alternative-representation-character-sets
                                    {$CHAR-SET-LIST};
   #Document-constituent-attributes   {
{  #Page-dimensions                 {$NON-BASIC-PAG-DIM},
   #Medium-types                    {$NON-BASIC-NOM-PAG-SIZ} } } };

PERM    Additional-doc-characteristics     {
{  #Fonts-list                      {ANY},
   #Unit-scaling                    {ANY} } };
```

16.8.1.2.3 Document management attributes

```
PERM    Title                           {ANY};
PERM    Subject                         {ANY};
PERM    Document-type                   {ANY};
PERM    Abstract                        {ANY};
PERM    Document-date-and-time          {ANY};
PERM    Keywords                        {ANY};
PERM    Document-reference              {ANY};
PERM    Dates-and-times                 {ANY};
PERM    Originators                     {ANY};
PERM    Other-user-information          {ANY};
PERM    External-references             {ANY};
PERM    Local-file-references           {ANY};
PERM    Content-attributes              {ANY};
PERM    Security-information            {ANY};
```

16.8.2 Logical Constituent Constraints

16.8.2.1 Diagrams of Relationships of Logical Constituents

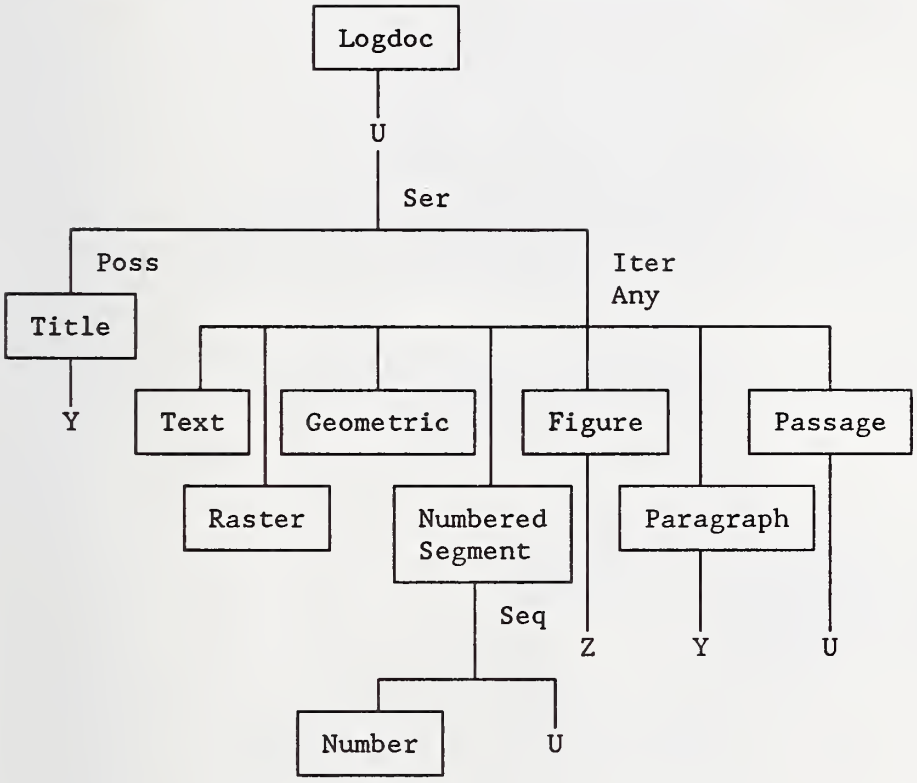


Figure 16.4. Diagram of logical structure (1 of 4).

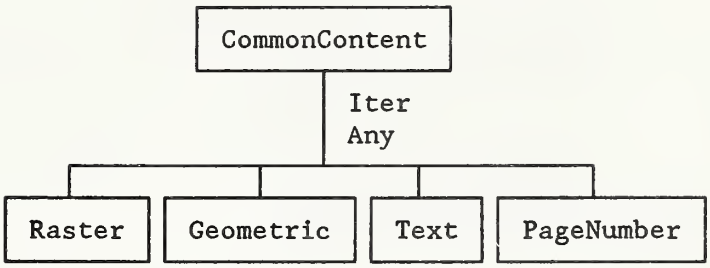


Figure 16.5. Diagram of logical structure (2 of 4).

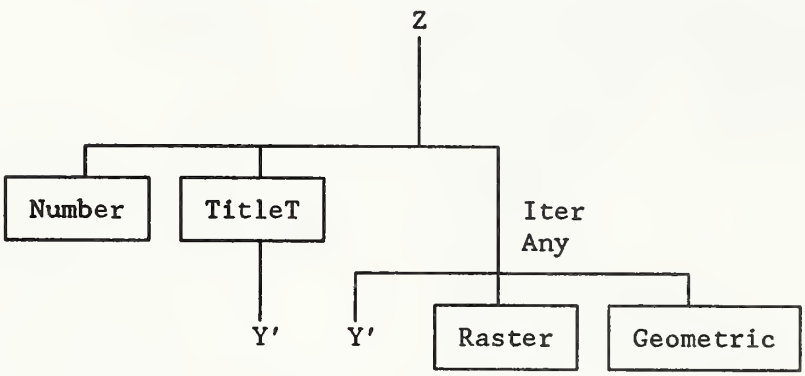


Figure 16.6. Diagram of logical structure (3 of 4).

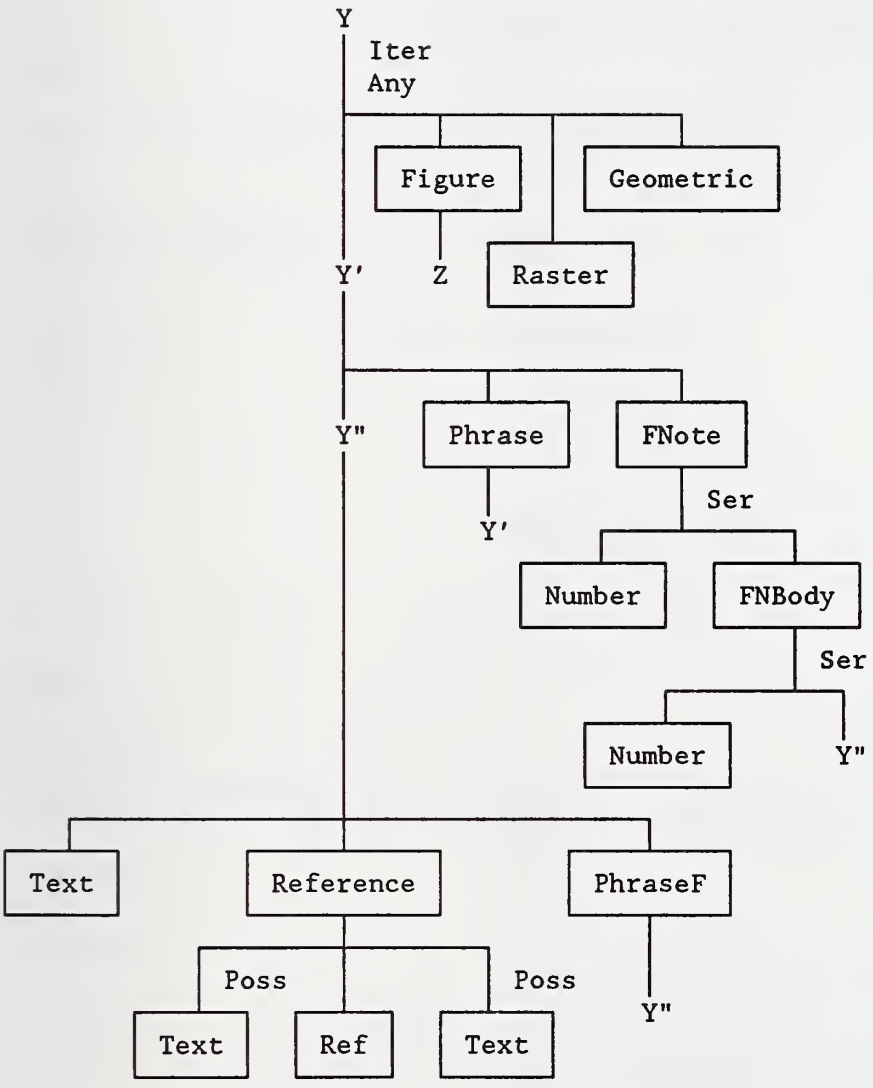


Figure 16.7. Diagram of logical structure (4 of 4).

16.8.2.2 Macro definitions

```
DEFINE(N,"
<n>                          :=   --any character string from the set
                                  of characters: "0," "1," "9"-- ")

DEFINE(NUMBERS,"
<numbers>                    :=   "number-"<$N> ")

DEFINE(NUMBERSTRINGS,"
<numberstrings>              :=   "numberstring-"<$N> ")

DEFINE(PREFIXES,"
<prefixes>                   :=   "prefix-"<$N> ")

DEFINE(SUFFIXES,"
<suffixes>                   :=   "suffix-"<$N> ")

DEFINE(SEPARATORS,"
<separators>                 :=   "separator-"<$N> ")

DEFINE(INITIALISEANY,"
<binding-pair-constraint> :=
                     {
     <$PREFIXES>, STRING_LITERAL | <$SUFFIXES>, STRING_LITERAL |
     <$SEPARATORS>, STRING_LITERAL | <$NUMBERS>, STRING_LITERAL |
     <$NUMBERSTRINGS>, " "
                     } + ")

DEFINE(USENUMBERS,"
<binding-pair-constraint> :=
                     {
     <$NUMBERS>,INC(B_REF(PREC(CURR_OBJ)) (<$NUMBERS>)) |
     <$NUMBERSTRINGS>, <hierarchic-expr> |
     <simple-expr>
                     } +

<hierarhic-expr>             :=   B_REF(SUP(CURR_OBJ))
                                  (<$NUMBERSTRINGS>) +
                                  B_REF(SUP(CURR_OBJ))
                                  (<$SEPARATORS>) + <simple-expr)")

<simple-expr>                :=   <string-function>
                                  (B_REF(CURR_OBJ)($NUMBERS)) |
                                  <string-function> (ORD(CURR_OBJ)) |
                                  <STRING-LITERAL>
<string-function>            :=   MK_STR | U_ALPHA | L_ALPHA | U_ROM
                                  | L_ROM ")
DEFINE(SEGMENTNUMBER,"
<string-expr-constraint> :=   [<pre-st1>] <num-st1> [<suf-st1>]
```

```
<num-st1>                  :=   B_REF(SUP(CURR_OBJ))
                                (<$NUMBERSTRINGS>)
<pre-st1>                  :=   B_REF(SUP(CURR_OBJ)) (<$PREFIXES>)
                                | STRING_LITERAL
<suf-st1>                  :=   B_REF(SUP(CURR_OBJ)) (<$SUFFIXES>)
                                | STRING_LITERAL ")

DEFINE(PAGENUMBER1,"
<string-expr-constraint>   :=   [<pgpre-st2>] <pgnum-st2>
                                [<pgsuf-st2>]
<pgpre-st2>                :=   STRING_LITERAL
<pgsuf-st2>                :=   STRING_LITERAL
<pgnum-st2>                :=   <string-function>
                                (<numeric-expr-1>)
<numeric-expr-1>           :=   B_REF(SUP(CURR_INST(
                                <class-or-type1>, CURR_OBJ)))
                                ($NUMBERS | "PGnum") |
                                B_REF(CURR_INST(<class-or-type2>,
                                CURR_OBJ)) ($NUMBERS | "PGnum")
<class-or-type-1>          :=   frame | OBJECT_CLASS_ID_OF({FrameH
                                | FrameJ | FrameK})
<class-or-type-2>          :=   page | OBJECT_CLASS_ID_OF(Page) ")

DEFINE(REF, "
<string-expr-constraint>   :=   [<pre-st4>] <num-st4> [<suf-st4>]
<pre-st4>                  :=   B_REF(<any-object>) (<$PREFIXES>) |
                                STRING_LITERAL
<suf-st4>                  :=   B_REF(<any-object>) (<$SUFFIXES>) |
                                STRING_LITERAL
<num-st4>                  :=   B_REF(<any-object>)
                                (<$NUMBERSTRINGS>)
<any-object>               :=   OBJECT_CLASS_ID_OF({Logdoc |
                                Passage | NumberedSegment | Number
                                | Title | TitleT | Paragraph |
                                Phrase | PhraseF | FNote | FNBody |
                                Figure | Text | Reference | Ref |
                                Raster | Geometric | CommonContent
                                | PageNumber | LayDoc | PageSet |
                                Page | Header | Footer | BodyFrame1
                                | BodyFrame2 | FrameA | FrameB |
                                FrameC | FrameD | FrameE | FrameF |
                                FrameG | FrameH | FrameI | FrameK |
                                Block}) ")
```

16.8.2.3 Factor constraints

```
FACTOR:   ANY-LOGICAL    (

GENERIC:
REQ  Object-class-identifier        {ANY};
PERM Resource                       {ANY};

SPECIFIC:
REQ  Object-identifier              {ANY};

SPECIFIC_AND_GENERIC:
PERM Protection                     {ANY};
PERM User-readable-comment          {ANY};
PERM User-visible-name              {ANY};
)

FACTOR:   COMP-LOGICAL   :ANY-LOGICAL   (

GENERIC:
REQ  Object-type                    {composite-logical-object};

SPECIFIC:
REQ  Subordinates                   {ANY};
PERM Object-type                    {composite-logical-object};

SPECIFIC_AND_GENERIC:
PERM Layout-style                   {STYLE_OF(LStyle3)};
PERM Default-value-lists            {ANY};
)

FACTOR:   BASIC-LOGICAL  :ANY-LOGICAL   (

GENERIC:
REQ  Object-type                    {basic-logical-object};

SPECIFIC:
PERM Object-type                    {basic-logical-object};
PERM Content-portions               {ANY};
)
```

16.8.2.4 Logdoc :ANY-LOGICAL (

```
GENERIC:
REQ  Object-type                    {document-logical-root};
REQ  Generator-for-subordinates     {ser(poss(Title),
                                    (iter(any(Paragraph,
                                    Figure,NumberedSegment,
                                    Text,Raster,Geometric,
                                    Passage))));
SPECIFIC:
```

```
REQ  Object-class                  {OBJECT_CLASS_ID_OF(Logdoc)};
REQ  Subordinates                  {ANY};
PERM Object-type                   {document-logical-root}

SPECIFIC_AND_GENERIC:
PERM Layout-style                  {STYLE_OF(LStyle1)};
PERM Bindings                      {$INITIALISEANY};
PERM Default-value-lists           {ANY};
PERM Application-comments          {"Logdoc"};
}
```

16.8.2.5 Passage :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates    {ser(poss(Title),
                                   iter(any(Paragraph,
                                   Figure,NumberedSegment,
                                   Text,Raster,Geometric,
                                   Passage))))};

SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(Passage)};

SPECIFIC_AND_GENERIC:
PERM Bindings                      {$INITIALISEANY |
                                   $USENUMBERS};
PERM Application-comments          {"Passage"};
}
```

16.8.2.6 NumberedSegment COMP:-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates    {seq(Number,{ser(poss(Title),
                                   iter(any(Paragraph,
                                   Figure,NumberedSegment,
                                   Text,Raster,Geometric,
                                   Passage)))))};
REQ  Bindings                      {$USENUMBERS};
REQ  Application-comments          {"NumberedSegment"};

SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(
                                   NumberedSegment)};
PERM Bindings                      {$INITIALISEANY |
                                   $USENUMBERS};
PERM Application-comments          {"NumberedSegment"};
SPECIFIC_AND_GENERIC:
}
```

16.8.2.7 Number :BASIC-LOGICAL {

```
GENERIC:
REQ  Content-generator              {$SEGMENTNUMBER};
REQ  Application-comments           {"Number"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Number)};
PERM Application-comments           {"Number"};

SPECIFIC_AND_GENERIC:
PERM Presentation-style             {STYLE_OF(PStyle1)};
PERM Layout-style                   {STYLE_OF(LStyle4)};
PERM Content-architecture-class     {$CF | $CP | $CFP};
}
```

16.8.2.8 Title :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {iter(any(Text, Reference,
                                    Phrase, FNote, Figure, Raster,
                                    Geometric))};
REQ  Application-comments           {"Title"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Title)};
PERM Application-comments           {"Title"};
}
```

16.8.2.9 TitleT :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {iter(any(Text, Reference,
                                    Phrase, PhraseF, FNote))};
REQ  Application-comments           {"TitleT"};
SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(TitleT)};
PERM Application-comments           {"TitleT"};
}
```

16.8.2.10 Paragraph :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {iter(any(Text, Reference,
                                    PhraseF, Phrase,  FNote,
                                    Figure, Raster, Geometric))};
REQ  Application-comments           {"Paragraph"};
SPECIFIC:
```

```
REQ  Object-class                    {OBJECT_CLASS_ID_OF(
                                     Paragraph)};
PERM Application-comments            {"Paragraph"};
}
```

16.8.2.11 Phrase :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates      {iter(any(Text, Reference,
                                     Phrase, PhraseF, FNote))};
REQ  Application-comments            {"Phrase"};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(Phrase)};
PERM Application-comments            {"Phrase"};
}
```

16.8.2.12 PhraseF :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates      {iter(any(Text, Reference,
                                     PhraseF))};
REQ  Application-comments            {"PhraseF"};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(PhraseF)};
PERM Application-comments            {"PhraseF"};
}
```

16.8.2.13 FNote :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates      {ser(Number, FNBody)};
REQ  Application-comments            {"FNote"};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FNote)};
PERM Application-comments            {"FNote"};

SPECIFIC_AND_GENERIC:
PERM Bindings                        {$INITIALISEANY |
                                     $USENUMBERS};
}
```

16.8.2.14 FNBody :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {ser(Number, Text, Reference,
                                    PhraseF};
REQ  Application-comments           {"FNBody"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(FNBody)};
PERM Application-comments           {"FNBody"};
}
```

16.8.2.15 Figure :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {ser(poss(Number), TitleT,
                                    iter(any(Text, Reference,
                                    PhraseF, Phrase, FNote,
                                    Raster, Geometric)))};
REQ  Application-comments           {"Figure"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Figure)};
PERM Application-comments           {"Figure"};

SPECIFIC_AND_GENERIC:
PERM Bindings                       {$INITIALISEANY |
                                    $USENUMBERS};
}
```

16.8.2.16 Text :BASIC-LOGICAL {

```
GENERIC:
REQ  Application-comments           {"Text"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Text)};
PERM Application-comments           {"Text"};

SPECIFIC_AND_GENERIC:
PERM Content-architecture-class     {$CF | $CP | $CFP};
PERM Content-portions               {ANY};
PERM Presentation-style             {STYLE_OF(PStyle2)};
PERM Layout-style                   {STYLE_OF(LStyle5)};
}
```

16.8.2.17 Reference :COMP-LOGICAL {

```
GENERIC:
REQ  Generator-for-subordinates     {ser(poss(Text), Ref,
                                    poss(Text))};
REQ  Application-comments           {"Reference"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(
                                    Reference)};
PERM Application-comments           {"Reference"};

SPECIFIC_AND_GENERIC:
PERM Bindings                       {$INITIALISEANY |
                                    $USENUMBERS};
}
```

16.8.2.18 Ref :BASIC-LOGICAL {

```
GENERIC:
REQ  Application-comments           {"Ref"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Ref)};
PERM Application-comments           {"Ref"};

SPECIFIC_AND_GENERIC:
PERM Content-generator              {$REF};
PERM Content-portions               {ANY};
PERM Content-architecture-class     {$CP | $CFP};
PERM Presentation-style             {STYLE_OF(PStyle2)};
PERM Layout-style                   {STYLE_OF(LStyle5)};
}
```

16.8.2.19 Raster :BASIC-LOGICAL {

```
GENERIC:
REQ  Application-comments           {"Raster"};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Raster)};
PERM Application-comments           {"Raster"};

SPECIFIC_AND_GENERIC:
PERM Content-architecture-class     {$RFP};
PERM Content-portions               {ANY};
PERM Presentation-style             {STYLE_OF(PStyle3)};
PERM Layout-style                   {STYLE_OF(LStyle6)};
}
```

16.8.2.20 Geometric :BASIC-LOGICAL {

```
GENERIC:
REQ  Application-comments          {"Geometric"};

SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(
                                   Geometric)};
PERM Application-comments          {"Geometric"};

SPECIFIC_AND_GENERIC:
PERM Content-architecture-class    {$GFP};
PERM Content-portions              {ANY};
PERM Presentation-style            {STYLE_OF(PStyle4)};
PERM Layout-style                  {STYLE_OF(LStyle6)};
}
```

16.8.2.21 CommonContent {

```
GENERIC:
REQ  Object-type                   {composite-logical-object};
REQ  Object-class-identifier       {ANY};
REQ  Generator-for-subordinates    {iter(any(Raster, Geometric,
                                   Text, PageNumber))};
REQ  Application-comments          {"CommonContent"};
PERM Resource                      {ANY};
PERM User-readable-comments        {ANY};
PERM User-visible-name             {ANY};
PERM Protection                    {ANY};
PERM Default-value-list            {ANY};
}
```

16.8.2.22 PageNumber {

```
GENERIC:
REQ  Object-type                   {basic-logical-object};
REQ  Object-class-identifier       {ANY};
REQ  Content-generator             {$PAGENUMBER1};
PERM Resource                      {ANY};
PERM Presentation-style            {STYLE_OF(PStyle2)};
PERM Content-architecture-class    {$CP};
PERM User-readable-comments        {ANY};
PERM User-visible-name             {ANY};
PERM Protection                    {ANY};
PERM Layout-style                  {STYLE_OF(LStyle2)};
PERM Application-comments          {"PageNumber"};
          }
```

16.8.3 Layout Constituent Constraints

16.8.3.1 Diagrams of Relationships of Layout Constituents

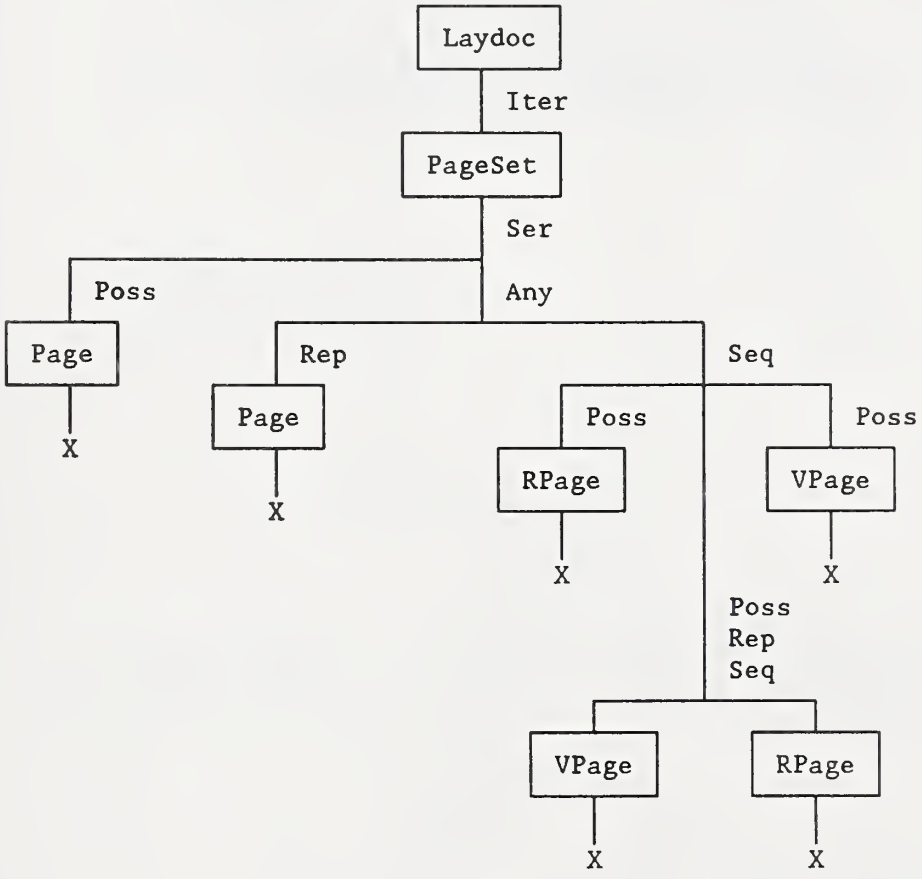
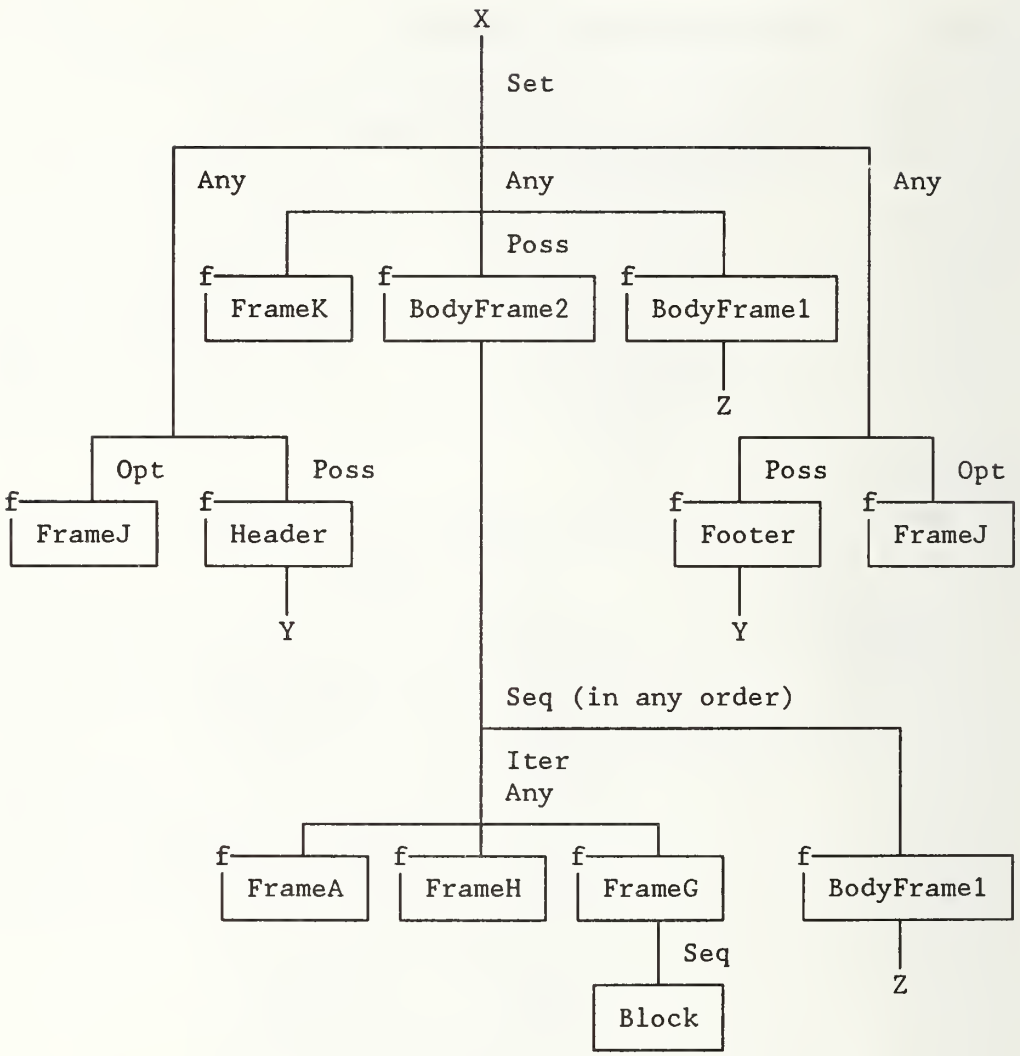
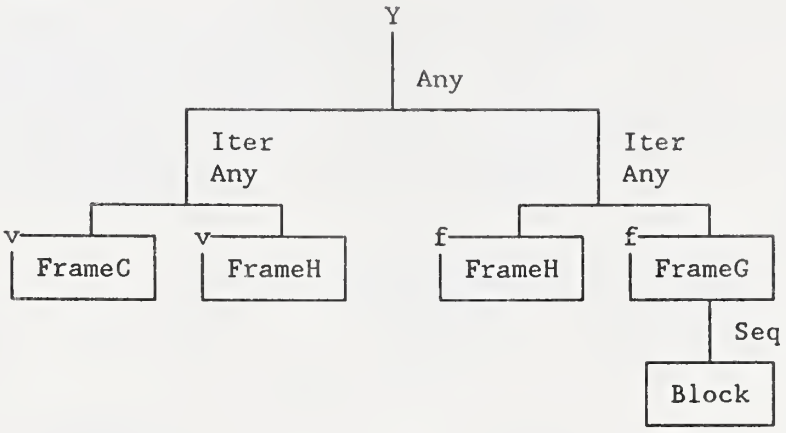


Figure 16.8. Diagram of Layout Structure (1 of 4).

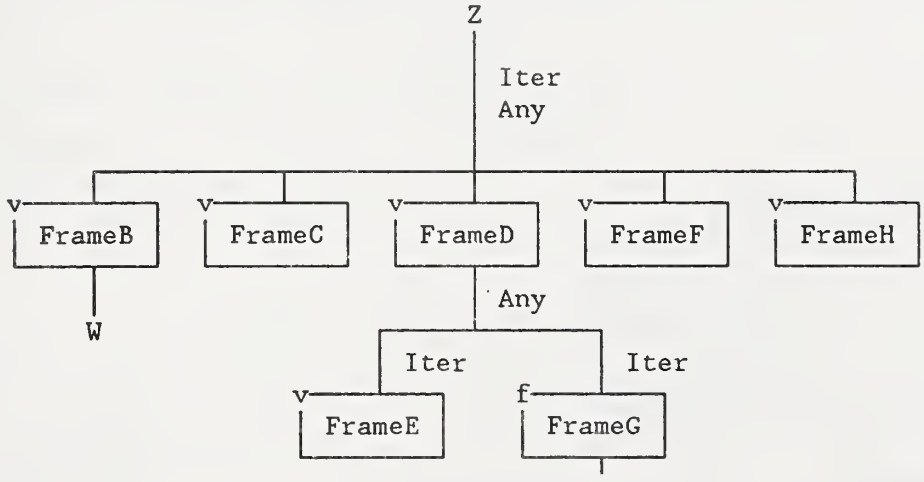


f: Fixed position
v: Variable position

Figure 16.9. Diagram of Layout Structure (2 of 4).

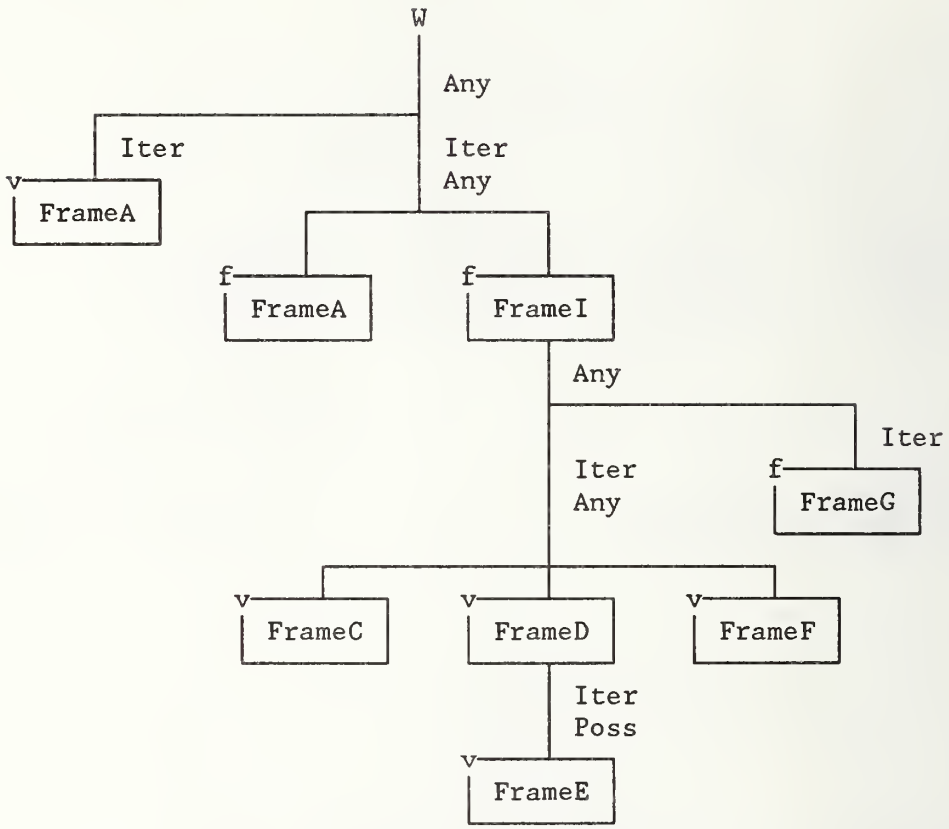


f: Fixed position
v: Variable position



f: Fixed position
v: Variable position

Figure 16.10. Diagram of Layout Structure (3 of 4).



f: Fixed position
v: Variable position

Figure 16.11. Diagram of Layout Structure (4 of 4).

16.8.3.2 Macro definitions

```
DEFINE(PAGENUMBER-2,"
<string-expr-constraint> :=   <string-function>
                              (<numeric-expression-2>)
<numeric-expression-2>   :=   B_REF( SUP_OBJ( CURR_INST( frame,
                              CURR_OBJ)))("PGnum") ")

DEFINE(PAGE-DIMENSIONS, "
-- Common Assured Reproduction Area of --
-- North American Letter and ISO A4 Portrait --
     #horizontal <= 9240, #vertical <= 12400 |
-- Assured Reproduction Area of --
-- North American Letter Portrait --
     #horizontal <= 9240, #vertical <= 12400 |
-- Assured Reproduction Area of ISO A4 Portrait --
```

```
     #horizontal <= 9240, #vertical <= 13200 ")
DEFINE(NOMINAL-PAGE-SIZE, "
-- North American Letter Portrait --
     #horizontal <= 10200, #vertical <= 13200 |
-- ISO A4 Portrait --
     #horizontal <= 9920, #vertical <= 14030 ")
```

16.8.3.3 Factor constraints

```
FACTOR:   ANY-LAYOUT     {

GENERIC:
REQ  Object-class                  {ANY};

SPECIFIC:
REQ  Object-identifier             {ANY};

SPECIFIC_AND_GENERIC:
PERM User-visible-name             {ANY};
PERM User-readable-comment         {ANY};
}

FACTOR:   ANY-PAGE  :ANY-LAYOUT    {

GENERIC:
REQ  Object-type                   {page};
REQ  Generator-for-subordinates    {set(any(opt(FrameJ),
                                   poss(Header)),
                                   any(opt(FrameK),
                                   poss(BodyFrame1),
                                   poss(BodyFrame2)),
                                   any(opt(FrameJ),
                                   poss(Footer)))};
PERM Resource                      {ANY};

SPECIFIC:
REQ  Subordinates                  {ANY};
PERM Object-type                   {page};

SPECIFIC_AND_GENERIC:
PERM Dimensions                    {$PAGE-DIMENSIONS};
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
PERM Page-position                 {ANY};
PERM Bindings                      {MANIPULATION(PGnum)};
}

FACTOR:   ANY-FRAME :ANY-LAYOUT    {
```

```
GENERIC:
REQ  Object-type                    {frame};
SPECIFIC:
PERM Object-type                    {frame};
)
```

16.8.3.4 Laydoc :ANY-LAYOUT {

```
GENERIC:
REQ  Object-type                    {document-layout-root};
REQ  Generator-for-subordinates     {iter(pageset)};
PERM Resource                       {ANY};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(Laydoc)};
REQ  Subordinates                   {ANY};
PERM Object-type                    {document-layout-root};

SPECIFIC_AND_GENERIC:
PERM Default-value-lists            {ANY};
PERM Bindings                       {Initialization(PGnum)};
PERM Application-comments           {"Laydoc"};
)
```

16.8.3.5 PageSet :ANY-LAYOUT {

```
GENERIC:
REQ  Object-type                    {pageset};
REQ  Generator-for-subordinates     {ser(poss(Page),
                                    any(rep(Page),
                                    seq(poss(RPage), poss rep
                                    seq(VPage, RPage),
                                    poss(VPage)))))};
PERM Resource                       {ANY};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(PageSet)};

REQ  Subordinates                   {ANY};
PERM Object-type                    {pageset};

SPECIFIC_AND_GENERIC:
PERM Bindings                       {Initialization(PGnum)};
PERM Application-comments           {"PageSet"};
)
```

16.8.3.6 Page :ANY-PAGE {

```
SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(Page)};

SPECIFIC_AND_GENERIC:
PERM Medium-type                   {
     #Nominal-page-size            {$NOMINAL-PAGE-SIZE},
     #Side-of-sheet                {"unspecified"}     );
PERM Application-comments          {"Page"};
}
```

16.8.3.7 RPage :ANY-PAGE {

```
SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(RPage)};

SPECIFIC_AND_GENERIC:
PERM Medium-type                   {
     #Nominal-page-size            {$NOMINAL-PAGE-SIZE},
     #Side-of-sheet                {"recto"} );
PERM Application-comments          {"RPage"};
}
```

16.8.3.8 VPage :ANY-PAGE {

```
SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(VPage)};

SPECIFIC_AND_GENERIC:
PERM Medium-type                   {
     #Nominal-page-size            {$NOMINAL-PAGE-SIZE},
     #Side-of-sheet                {"verso"} );
PERM Application-comments          {"VPage"};
}
```

16.8.3.9 Header :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates    {any(iter(any(FrameC,FrameH)),
                                   iter(any(FrameH,FrameG)))};
REQ  Position                      {#fixed{ANY}};
REQ  Dimensions                    {#horizontal{#fixed{ANY}},
                                   #vertical{#fixed{ANY}}};
REQ  Application-comments          {"Header"};
PERM Resource                      {ANY};
```

```
SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(Header)};
REQ  Subordinates                    {ANY};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"Header"};

SPECIFIC_AND_GENERIC:
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
PERM Layout-path                     {90 | 270};
}
```

16.8.3.10 Footer :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates      {any(iter(any(FrameC,FrameH)),
                                     iter(any(FrameH,FrameG)))};
REQ  Position                        {#fixed{ANY}};
REQ  Dimensions                      {ANY};
REQ  Application-comments            {"Footer"};
PERM Resource                        {ANY};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(Footer)};
REQ  Subordinates                    {ANY};
PERM Position                        {ANY};
PERM Dimensions                      {#fixed{ANY}};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"Footer"};

SPECIFIC_AND_GENERIC:
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
PERM Layout-path                     {90 | 270};
}
```

16.8.3.11 BodyFrame1 :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates      {iter(any(FrameB,FrameC,
                                     FrameD,FrameF,FrameH))};
REQ  Position                        {#fixed{ANY}};
REQ  Dimensions                      {#horizontal{#fixed{ANY}},
                                     #vertical{#fixed{ANY}}};
REQ  Application-comments            {"BodyFrame1"};
PERM Resource                        {ANY};
```

```
SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF
                                   (BodyFrame1)};
REQ  Subordinates                  {ANY};
PERM Position                      {ANY};
PERM Dimensions                    {ANY};
PERM Imaging-order                 {ANY};
PERM Application-comments          {"BodyFrame1"};

SPECIFIC_AND_GENERIC:
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
PERM Border                        {ANY};
PERM Layout-path                   {90 | 270};
}
```

16.8.3.12 BodyFrame2 :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates    {seq(iter(any(FrameA, FrameH,
                                   FrameG, BodyFrame1)))};
REQ  Position                      {#fixed{ANY}};
REQ  Dimensions                    {#horizontal{#fixed{ANY}},
                                   #vertical{#fixed{ANY}}};
REQ  Application-comments          {"BodyFrame2"};
PERM Resource                      {ANY};

SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF
                                   (BodyFrame2)};
REQ  Subordinates                  {ANY};
PERM Position                      {ANY};
PERM Dimensions                    {ANY};
PERM Imaging-order                 {ANY};
PERM Application-comments          {"BodyFrame2"};

SPECIFIC_AND_GENERIC:
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
PERM Border                        {ANY};
PERM Layout-path                   {90 | 270};
}
```

16.8.3.13 FrameA :ANY-FRAME (

```
GENERIC:
REQ  Position                        {#fixed{ANY} |
                                     #variable{ANY}};
REQ  Dimensions                      {#horizontal{ANY},
                                     #vertical{ANY} |
                                     #horizontal{fixed, default},
                                     #vertical{RuleB}};
REQ  Application-comments            {"FrameA"};
SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameA)};
REQ  Subordinates                    {OBJECT_ID_OF(Block)};
PERM Position                        {ANY};
PERM Dimensions                      {ANY};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"FrameA"};

SPECIFIC_AND_GENERIC:
PERM Permitted-categories            {ANY};
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
)
```

16.8.3.14 FrameB :ANY-FRAME (

```
GENERIC:
REQ  Generator-for-subordinates      {any(iter(FrameA),
                                     iter(any(FrameA,FrameI)))};
REQ  Position                        {#variable{ANY}};
REQ  Dimensions                      {#horizontal{fixed{ANY},
                                     default}, #vertical{RuleB}};
REQ  Application-comments            {"FrameB"};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameB)};
REQ  Subordinates                    {OBJECT_ID_OF(Block)};
PERM Position                        {ANY};
PERM Dimensions                      {ANY};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"FrameB"};

SPECIFIC_AND_GENERIC:
PERM Permitted-categories            {ANY};
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
PERM Layout-path                     {0| 90| 270};
```

```
PERM Balance                        {ANY};
}
```

16.8.3.15 FrameC :ANY-FRAME {

```
GENERIC:
REQ  Position                       {#variable{ANY}};
REQ  Dimensions                     {#horizontal{fixed{ANY},
                                    default}, #vertical{RuleB}};
REQ  Application-comments           {"FrameC"};
PERM Resource                       {ANY};
SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(FrameC)};
REQ  Subordinates                   {OBJECT_ID_OF(Block)};
PERM Position                       {ANY};
PERM Dimensions                     {#horizontal{default},
                                    #vertical{ANY}};
PERM Imaging-order                  {ANY};
PERM Application-comments           {"FrameC"};
SPECIFIC_AND_GENERIC:
PERM Permitted-categories           {ANY};
PERM Transparency                   {ANY};
PERM Colour                         {ANY};
PERM Border                         {ANY};
}
```

16.8.3.16 FrameD :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates     {any(iter(FrameE),
                                    iter(FrameG))};
REQ  Position                       {#variable{ANY}};
REQ  Dimensions                     {#horizontal{default},
                                    #vertical{RuleA}};
REQ  Layout-path                    {0 | 180};
REQ  Application-comments           {"FrameD"};
PERM Resource                       {ANY};

SPECIFIC:
REQ  Object-class                   {OBJECT_CLASS_ID_OF(FrameD)};
REQ  Subordinates                   {ANY};
PERM Position                       {ANY};
PERM Dimensions                     {#horizontal{default},
                                    #vertical{ANY}};
PERM Imaging-order                  {ANY};
PERM Application-comments           {"FrameD"};
```

```
SPECIFIC_AND_GENERIC:
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
PERM Bindings                        {MANIPULATION(PGnum)};
}
```

16.8.3.17 FrameE :ANY-FRAME {

```
GENERIC:
REQ  Position                        {#variable{ANY}};
REQ  Dimensions                      {#horizontal{RuleB, fixed,
                                     default}, #vertical{RuleB}};
REQ  Application-comments            {"FrameE"};
PERM Resource                        {ANY};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameE)};
REQ  Subordinates                    {OBJECT_ID_OF(Block)};
PERM Position                        {ANY};
PERM Dimensions                      {ANY};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"FrameE"};

SPECIFIC_AND_GENERIC:
REQ  Permitted-categories            {ANY};
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
}
```

16.8.3.18 FrameF :ANY-FRAME {

```
GENERIC:
REQ  Position                        {#variable{
                                     #fillorder{reversed}},
                                     #offset{ANY},
                                     #separation{ANY},
                                     #alignment{ANY}};
REQ  Dimensions                      {#horizontal{default},
                                     #vertical{RuleB}};
REQ  Application-comments            {"FrameF"};
PERM Resource                        {ANY};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameF)};
REQ  Subordinates                    {OBJECT_ID_OF(Block)};
PERM Position                        {ANY};
```

```
PERM Dimensions                    {#horizontal{default},
                                   #vertical{ANY}};
PERM Imaging-order                 {ANY};
PERM Application-comments          {"FrameF"};

SPECIFIC_AND_GENERIC:
REQ  Permitted-categories          {ANY};
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
PERM Border                        {ANY};
}
```

16.8.3.19 FrameG :ANY-FRAME {

```
GENERIC:
REQ  Position                      {#fixed{ANY}};
REQ  Dimensions                    {#horizontal{ANY},
                                   #vertical{ANY}};
REQ  Application-comments          {"FrameG"};
PERM Generator-for-subordinates    {seq{Block}};
PERM Resource                      {ANY};
SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(FrameG)};
REQ  Subordinates                  {OBJECT_ID_OF(Block)};
PERM Position                      {ANY};
PERM Dimensions                    {ANY};
PERM Imaging-order                 {ANY};
PERM Application-comments          {"FrameG"};

SPECIFIC_AND_GENERIC:
PERM Permitted-categories          {ANY};
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
PERM Border                        {ANY};
}
```

16.8.3.20 FrameH :ANY-FRAME {

```
GENERIC:
REQ  Position                      {#fixed{ANY} |
                                   #variable{ANY}};
REQ  Dimensions                    {#horizontal{default},
                                   #vertical{RuleB}};
REQ  Logical-source                {OBJECT_CLASS_ID_OF(
                                   CommonContent)};
REQ  Application-comments          {"FrameH"};
PERM Generator-for-subordinates    {iter(Block)};
PERM Resource                      {ANY};
```

```
SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameH)};
REQ  Subordinates                    {OBJECT_ID_OF(Block)};
PERM Position                        {ANY};
PERM Dimensions                      {#horizontal{default},
                                     #vertical{ANY}};
PERM Application-comments            {"Frame"};

SPECIFIC_AND_GENERIC:
PERM Border                          {ANY};
PERM Layout-path                     {ANY};
}
```

16.8.3.21 FrameI :ANY-FRAME {

```
GENERIC:
REQ  Generator-for-subordinates      {any(iter(any(FrameC,
                                     FrameD,FrameF)),
                                     iter(FrameG))};
REQ  Position                        {#fixed{ANY}|#variable{ANY};
REQ  Dimensions                      {#horizontal{fixed, default},
                                     #vertical{RuleB}};
REQ  Application-comments            {"FrameI"};
PERM Resource                        {ANY};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameI)};
REQ  Subordinates                    {ANY};
PERM Position                        {ANY};
PERM Dimensions                      {ANY};
PERM Imaging-order                   {ANY};
PERM Application-comments            {"FrameI"};

SPECIFIC_AND_GENERIC:
PERM Layout-path                     {90 | 270};
PERM Transparency                    {ANY};
PERM Colour                          {ANY};
PERM Border                          {ANY};
}
```

16.8.3.22 FrameJ :ANY-FRAME {

```
GENERIC:
REQ  Logical-source                  {ANY};
PERM Position                        {#fixed{ANY}};
PERM Dimensions                      {#fixed{ANY}};

SPECIFIC:
REQ  Object-class                    {OBJECT_CLASS_ID_OF(FrameJ)};
```

```
REQ  Subordinates                  {OBJECT_ID_OF(Block)};
PERM Position                      {ANY};
PERM Dimensions                    {#fixed{ANY}};
PERM Layout-path                   {270};

SPECIFIC_AND_GENERIC:
PERM Application-comments          {"FrameJ"};
}
```

16.8.3.23 FrameK :ANY-FRAME {

```
GENERIC:
PERM Position                      {#fixed{ANY}};

SPECIFIC:
REQ  Object-class                  {OBJECT_CLASS_ID_OF(FrameK)};
REQ  Subordinates                  {OBJECT_ID_OF(Block)};
PERM Position                      {ANY};
PERM Layout-path                   {270};

SPECIFIC_AND_GENERIC:
PERM Dimensions                    {#fixed{ANY}};
PERM Application-comments          {"FrameK"};
}
```

16.8.3.24 Block :ANY-LAYOUT {

```
GENERIC:
REQ  Object-type                   {block};
REQ  Content-architecture-class    {ANY};
PERM Content-generator             {$PAGENUMBER-2};
REQ  Application-comments          {"Block"};
PERM Resource                      {ANY};

SPECIFIC:
REQ  Content-architecture-class    {ANY};
PERM object-type                   {Block};
PERM Position                      {#fixed{ANY}};
PERM Dimensions                    {#horizontal{#fixed{ANY}},
                                   #vertical{#fixed{ANY}}};
PERM Initial-offset                {ANY};
PERM Formatting-indicator          {ANY};
PERM Graphic-rendition             {ANY};
PERM Graphic-character-set         {ANY};
PERM Application-comments          {"Block"};

SPECIFIC_AND_GENERIC:
PERM Content-portions              {ANY};
PERM Transparency                  {ANY};
PERM Colour                        {ANY};
```

```
PERM Border                          {ANY};
PERM Presentation-style              {STYLE_ID_OF(PStyle1 | PStyle2
                                     | PStyle3 | PStyle4};
}
```

16.8.4 Layout style constraints

16.8.4.1 Factors

```
FACTOR ANY-LAYOUT-STYLE {

REQ  Layout-style-identifier         {ANY};
PERM User-visible-name               {ANY};
PERM User-readable-comments          {ANY};
}
```

16.8.4.2 LStyle1 :ANY-LAYOUT-STYLE {

```
REQ  Layout-object-class             {OBJECT_CLASS_ID_OF(Laydoc)};
}
```

16.8.4.3 LStyle2 :ANY-LAYOUT-STYLE {

```
PERM Block-alignment                 {ANY};
PERM Concatenation                   {ANY};
PERM Indivisibility                  {ANY};
PERM Layout-category                 {ANY};
PERM Layout-object-class             {ANY};
PERM New-layout-object               {ANY};
PERM Same-layout-object              {ANY};
PERM Offset                          {ANY};
PERM Separation                      {ANY};
}
```

16.8.4.4 LStyle3 :ANY-LAYOUT-STYLE {

```
PERM Indivisibility                  {ANY};
PERM Layout-object-class             {ANY};
PERM New-layout-object               {ANY};
PERM Same-layout-object              {ANY};
PERM Synchronization                 {ANY};
}
```

16.8.4.5 LStyle4 :ANY-LAYOUT-STYLE {

```
PERM Block-alignment                {ANY};
PERM Concatenation                  {ANY};
PERM Indivisibility                 {ANY};
PERM Layout-category                {ANY};
PERM Layout-object-class            {ANY};
PERM New-layout-object              {ANY};
PERM Same-layout-object             {ANY};
PERM Offset                         {ANY};
PERM Separation                     {ANY};
PERM Synchronisation                {ANY};
}
```

16.8.4.6 LStyle5 :ANY-LAYOUT-STYLE {

```
PERM Block-alignment                {ANY};
PERM Concatenation                  {ANY};
PERM Indivisibility                 {ANY};
PERM Layout-category                {ANY};
PERM Layout-object-class            {ANY};
PERM New-layout-object              {ANY};
PERM Same-layout-object             {ANY};
PERM Offset                         {ANY};
PERM Separation                     {ANY};
PERM Synchronisation                {ANY};
PERM Fill-order                     {ANY};
}
```

16.8.4.7 LStyle6 :ANY-LAYOUT-STYLE {

```
PERM Block-alignment                {ANY};
PERM Indivisibility                 {ANY};
PERM Layout-category                {ANY};
PERM Layout-object-class            {ANY};
PERM New-layout-object              {ANY};
PERM Same-layout-object             {ANY};
PERM Offset                         {ANY};
PERM Separation                     {ANY};
PERM Synchronisation                {ANY};
}
```

16.8.5 Presentation style constraints

16.8.5.1 Macros

```
DEFINE(C-PRES-ATTR,"
PERM Alignment                      {ANY};
PERM Character-fonts                {ANY};
PERM Character-orientation          {ANY};
PERM Character-path                 {ANY};
PERM Character-spacing              {ANY};
PERM Code-extension-announcer       {ANY};
PERM First-line-offset              {ANY};
PERM Formatting-indicator           {ANY};
PERM Graphic-character-sets         {$CHAR-SET-LIST};
PERM Character-subrepertoire        { 2 -- Minimal --
                                    | 3 -- Teletex --
                                    | 5 -- ISO 646 --
                                    | 8 -- ISO 8859-1 --};
PERM Graphics-rendition             {ANY};
PERM Indentation                    {ANY};
PERM Initial-offset                 {ANY};
PERM Itemization                    {ANY};
PERM Kerning-offset                 {ANY};
PERM Line-layout-table              {ANY};
PERM Line-progression               {ANY};
PERM Line-spacing                   {100 | 150 | 200 | 300 | 400};
PERM Orphan-size                    {ANY};
PERM Pairwise-kerning               {ANY};
PERM Proportional-line-spacing      {ANY};
PERM Widow-size                     {ANY}; ")

DEFINE(R-PRES-ATTR,"
PERM Pel-path                       {ANY};
PERM Line-progression               {ANY};
PERM Pel-spacing                    {75 | 100 | 150 | 200 | 240 |
                                    300 | 400 | 600 | 1200};
PERM Spacing-ratio                  {ANY};
PERM Clipping                       {ANY};
PERM Image-dimensions               {ANY}; ")

DEFINE(G-PRES-ATTR,"
PERM Geometric-graphics-encoding-announcer
{     #VDC-type                     {ANY},
      #Integer-precision            {16 | 32},
      #Real-precision               {(0 9 23) | (1 16 16)},
      #Index-precision              {8 | 16},
      #Colour-precision             {8 | 16},
      #Colour-index-precision       {8 | 16},
      #Maximum-colour-index         {ANY},
```

```
     #Colour-value-extent          {ANY},
     #Colour-selection-mode        {ANY};
     #VDC-integer-precision        {16 | 32},
     #VDC-real-precision           {{0 9 23} | {1 16 16}} };
PERM Line-rendition                {ANY};
PERM Marker-rendition              {ANY};
PERM Text-rendition
{
     #Font-list                    {ANY},
     #Character-set-list           {$CHAR-SET-LIST},
     #Character-coding-announcer   {basic-7-bit | basic-8-bit},
     #Text-bundle-index            {ANY},
     #Text-font-index              {ANY},
     #Text-precision               {ANY},
     #Character-expansion-factor   {ANY},
     #Character-spacing            {ANY},
     #Text-colour                  {ANY},
     #Character-height             {ANY},
     #Character-orientation        {ANY},
     #Text-path                    {ANY},
     #Text-alignment               {ANY},
     #Character-set-index          {ANY},
     #Text-asf                     {ANY}
     #Text-bundle-representation   {ANY} };
PERM Filled-area-rendition
{
     #Fill-bundle-index            {ANY},
     #Interior-style               {ANY},
     #Fill-colour                  {ANY},
     #Hatch-index                  {ANY},
     #Pattern-index                {1 .. 8},
     #Fill-reference-point         {ANY},
     #Pattern-size                 {ANY},
     #Pattern-table-representation
     {    #Pattern-table-index     {1 .. 8},
          #Number-of-columns       {1 .. 16},
          #Number-of-rows          {1 .. 16},
          #Local-colour-precision  {0 | 1 | 8 | 16},
          #Colour-arry             {ANY} },
     #Fill-asf                     {ANY} };
PERM Edge-rendition                {ANY},
PERM Colour-representation         {ANY};
PERM Transparency-specification    {ANY};
PERM Transformation-specification  {ANY};
PERM Region-of-interest-specification {ANY};
PERM Picture-orientation           {ANY};
PERM Picture-dimensions            {ANY}; ")
```

16.8.5.2 Factors

```
FACTOR:   ANY-PRESENTATION-STYLE (
REQ  Presentation-style-identifier {ANY};
PERM User-readable-comments        {ANY};
PERM User-visible-name             {ANY};
PERM Border                        {ANY};
PERM Colour                        {ANY};
PERM Transparency                  {ANY};
)
```

16.8.5.3 PStyle1 :ANY-PRESENTATION-STYLE (

```
PERM Presentation-Attributes       {$C-PRES-ATTR};
)
```

16.8.5.4 PStyle2 :ANY-PRESENTATION-STYLE (

```
CASE (Document-profile(Document-characteristics
      #Content-archicture-class)) OF
$FDA:
REQ  Content-architecture-class    {$CF};
$PDA:
REQ  Content-architecture-class    {$CP};
$FPDA:
REQ  Content-architecture-class    {$CFP};
-- ENDCASE --
PERM Presentation-attributes       {$C-PRES-ATTR};
)
```

16.8.5.5 PStyle3 :ANY-PRESENTATION-STYLE (

```
REQ  Content-architecture-class    {$RFP};
PERM Presentation-attributes       {$R-PRES-ATTR};
)
```

16.8.5.6 PStyle4 :ANY-PRESENTATION-STYLE (

```
REQ  Content-architecture-class    {$GFP};
PERM Presentation-attributes       {$G-PRES-ATTR};
)
```

16.8.6 Content portion constraints

16.8.6.1 Character content portion

```
SPECIFIC_AND_GENERIC:
PERM Content-identifier-layout     {ANY};
PERM Content-identifier-logical    {ANY};
REQ  Type-of-coding                {2 8 3 6 0};
PERM Alternative-representation    {ANY};
PERM Content-information           {
     {    #Character               {ANY},
-- Shared Control Functions --
          #CR                      {},
          #GCC                     {ANY},
          #IGS                     {ANY},
          #LF                      {},
          #PLD                     {},
          #PLU                     {},
          #SCS                     {ANY},
          #SGR                     {ANY},
          #SHS                     {0 | 1 | 2 | 3},
          #SLS                     {ANY},
          #SRS                     {ANY},
          #STAB                    {ANY},
          #SUB                     {},
          #SVS                     {ANY},
          #VPB                     {ANY},
          #VPR                     {ANY},
-- Layout Control Functions --

          #HPB                     {ANY},
          #HPR                     {ANY},
          #JFY                     {ANY},
          #SACS                    {ANY},
          #SRCS                    {ANY},
          #SSW                     {ANY},
-- Logical Control Functions --
          #BPH                     {},
          #NBH                     {},
          #PTX                     {ANY},
-- Delimiter Functions --
          #SOS                     {},
          #SP                      {},
          #ST                      {} };
```

16.8.6.2 Raster graphics content portion

```
DEFINE(T6,      "(2 8 3 7 0)")
DEFINE(T41D,    "(2 8 3 7 1)")
DEFINE(T42D,    "(2 8 3 7 2)")
DEFINE(BITMAP, "(2 8 3 7 3)")

PERM Content-identifier-logical     (ANY);
PERM Content-identifier-layout      (ANY);
REQ  Type-of-coding                 ($T6 | $T41D | $T42D |
                                    $BITMAP);
PERM Alternative-representation     (ANY);
PERM Coding-attributes              (
     (    #Compression              (ANY),
          #Number-of-lines          (ANY),
          #Number-of-pels-per-line  (ANY),
                                    );
PERM Content-information            (ANY);
```

16.8.6.3 Geometric graphics content portion

```
PERM Content-identifier-logical     (ANY);
PERM Content-identifier-layout      (ANY);
REQ  Alternative-representation     (ANY);
PERM Content-information            (ANY);
```

-- Annex B.2 contains a recommended functional subset of the CGM standard for this DAP --

16.8.7 Additional usage constraints

No other usage constraints are currently defined.

16.9 INTERCHANGE FORMAT

Interchange format class "A" is to be used in this application profile, as 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.

16.9.1 ASN.1 generation constraints

There are no additional constraints, beyond ISO 8824 and ISO 8825, imposed on the ASN.1 generation.

16.9.2 ASN.1 parsing constraints

There are no additional constraints, beyond ISO 8824 and ISO 8825, imposed on the ASN.1 parsing.

16.9.3 ASN.1 generation recommendations

The focus of the ASN.1 generation recommendations is to generate ASN.1 encodings that will allow parsing by the most rudimentary of implementations.

16.9.3.1 Ordering of set members

ISO 8824 defines sets to be unordered list of values. It is the generator's option to select an order for the values of the set. Since this ordering is unpredictable from one implementation to the next, it is recommended that generators order the values in a set according to the order in which the members appear in the definition of the set. The intent of this recommendation is to reduce the possible interoperability problems associated with the unpredictable ordering of members in a set.

16.9.4 ASN.1 parsing recommendations

The overall intent of these parsing recommendations is to allow a high tolerance in the representation of the ASN.1 syntax without jeoperdizing the semantics of the information being conveyed.

16.9.4.1 Encoding of application comments

ISO 8613-5 defines 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 Application-Comments-Encoding;

Application-Comments-Encoding :=   SEQUENCE {
     Constraint-name           [0]  IMPLICIT PrintableString
```

```
        External-data               [1]  EXTERNAL}
END
```

16.10 ANNEX A IMPLEMENTATION CONFORMANCE STATEMENT

16.10.1 Generator support statement proforma

The proforma for the GSS will be provided in the next version of this document.

16.10.2 Receiver support statement proforma

The proforma for the RSS will be provided in the next version of this document.

16.11 ANNEX B INFORMATIVE RECOMMENDATIONS

16.11.1 Overview of technical specifications

FrameA is a region of the page typically representing a column. The direct subordinates are blocks of content. FrameA is at a fixed or variable position within its superior frame. The dimension orthogonal to the layout path is fixed or variable; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinates.

FrameB is a region of the page, typically representing a number of columns. The direct subordinates may only be frames of type A or I. The position is variable (i.e., determined by a rule). The dimension orthogonal to the layout path is maximum for the position; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinates.

FrameC is a region of the page, typically representing an area of variable dimension within a column, such as is needed to contain a paragraph of text or a figure. This provides for varying layout on pages (columns or pages) as the document is edited. The direct subordinates are blocks of content. The position is variable (i.e., determined by a rule). The dimension orthogonal to the layout path is maximum for the position; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinate block(s).

FrameD is a region of the page, typically representing an area of variable dimension within a column, containing a number of items side by side (e.g., text flowing around a picture). The direct subordinates may only be frames of type E. The position is variable (i.e., determined by a rule). The dimension orthogonal to the layout path is maximum for the position; in the direction of the layout path, the dimension is the minimum size needed to contain the first laid out subordinate (e.g., the picture).

FrameE is a region of the page, typically representing some text or a picture, that is side by side with the frames of this type within a superior frame of type D. The direct subordinates are blocks of content. The position is variable (i.e., determined by a rule). The dimension orthogonal to the layout path is the minimum size needed to contain the subordinate block(s). In the direction of the layout path, the dimension is either a fixed dimension or is a computed dimension equal to either the minimum size needed to contain the subordinate block(s) (e.g., picture), or the maximum size for the position (e.g., text).

Frame F is a region of the page typically used to contain a footnote. The direct subordinates are blocks of content. The frame is positioned within its superior in reverse order at a fixed position from its superior in the direction orthogonal to the layout path and at a variable position in the direction of the layout path. The dimension orthogonal to the layout path is maximum for the position; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinate blocks.

FrameG is a region of the page, typically representing a set of pieces of content placed at defined positions. This provides for complex, fixed, relatively positioned pieces of content, including overlapping pieces of content (e.g., overlapping pictures or pictures and text). The direct subordinates are blocks of content. The position relative to the superior is fixed; the dimensions are also fixed.

FrameH is a region of the page, typically representing a variable piece of logical information in the header or footer area of a page (e.g., current chapter number). The direct subordinates are blocks of content. The frame, in all cases, specifies that its content is derived from the use of the attribute

"logical source" referring to a logical object of "header or footer content." The position is fixed or variable (i.e., determined by a rule.) The dimension orthogonal to the layout path is maximum for the position; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinate blocks.

Frame I: A region of the page typically representing a column. The frame may contain any substructure of further frames of any of the types C, F, or G. Blocks are not permitted directly subordinate to FrameI. FrameI is at a fixed or variable position within its superior frame. The dimension orthogonal to the layout path is fixed; in the direction of the layout path, the dimension is the minimum size needed to contain the subordinates.

Frame J is a single basic frame to contain header or footer contents. This is provided for compatibility with the CCITT Recommendation T.502.

Frame K is a single basic frame to represent the body area of a page. This is provided for compatibility with the CCITT Recommendation T.502.

16.11.2 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 Annex, in addition to the constraints imposed by ISO 8613-8. It is believed that this subset of the CGM is sufficiently widely implemented to enable interworking of geometric graphics for application conforming this DAP.

The content information of a content portion description that conforms to this content architecture is an ASN.1 octet string representing a Computer Graphics Metafile (CGM) conforming to the following constraints:

a) Conform to part 1 of the ISO 8632 standard;
b) Conform to the binary encoding defined in part 3 of the ISO 8632 standard;
c) Consist of a single picture;
d) Conform to the ISO pdISP FCG13, except as noted with respect to font and colour table support;

e) Generalized Drawing Primitives are ignored;
f) ESCAPE Elements are ignored;
g) External Elements may be ignored.

The following list is a description of the constraints for each of the CGM elements. 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.

16.11.2.1 Delimiter elements

No-Op	See Note 1
Begin Metafile	See Note 2
End Metafile	--
Begin Picture	See Note 2
Begin Picture Body	--
End Picture	--

16.11.2.2 Metafile description elements

Metafile Version	1
Metafile Description	See Notes 2, 3
VDC Type	--
Integer Precision	8, 16
Real Precision	(0,9,23), (1,16,16)
Index Precision	16
Colour Precision	8, 16
Colour Index Precision	8, 16
Maximum Colour Index	--
Colour Value Extent	--
Metafile Element List	--
Metafile Defaults Replacement	See Note 4
Font List	--
Character Set List	See Note 5
Character Coding Announcer	basic-7-bit, basic-8-bit

16.11.2.3 Picture descriptor elements

Scaling Mode	See Note 6
Colour Selection Mode	--
Line Width Specification Mode	--
Marker Size Specification Mode	--
Edge Width Specification Mode	--

VDC Extent	--
Background Colour	--

16.11.2.4 Control elements

VDC Integer Precision	16, 32
VDC Real Precision	(0,9,23), (1,16,16)
Auxiliary Colour	--
Transparency	--
Clip Rectangle	--
Clip Indicator	--

16.11.2.5 Graphical primitive elements

Polyline	See Note 7
Disjoint Polyline	See Note 7
Polymarker	See Note 7
Text	See Note 2
Restricted Text	See Notes 2, 8
Append Text	See Notes 2, 8
Polygon	See Note 7
Polygon Set	See Note 7
Cell Array	See Note 9
Rectangle	--
Circle	--
Circular Arc 3 Point	--
Circular Arc 3 Point Close	--
Circular Arc Centre	--
Circular Arc Centre Close	--
Ellipse	--
Elliptical Arc	--
Elliptical Arc Close	--

16.11.2.6 Attribute elements

Line Bundle Index	--
Line Type	--
Line Width	--
Line Colour	--
Marker Bundle Index	--
Marker Type	--
Marker Size	--
Marker Colour	--
Text Bundle Index	--
Text Font Index	--
Text Precision	--
Character Expansion Factor	--
Character Spacing	--
Text Colour	--

Character Height	--
Character Orientation	--
Text Path	--
Text Alignment	--
Character Set Index	--
Alternate Character Set Index	--
Fill Bundle Index	--
Interior Style	--
Fill Colour	--
Hatch Index	--
Pattern Index	1 .. 8
Edge Bundle Index	--
Edge Type	--
Edge Width	--
Edge Colour	--
Edge Visibility	--
Fill Reference Point	--
Pattern Table	See Notes 10, 11
Pattern Size	--
Colour Table Specification	See Notes 12, 13
Aspect Source Flags	--

16.11.2.7 External Elements

Message	No action
Application Data	See Note 2

Note 1: An arbitrary sequence of n octets. Where n=0, 1, .., 32767. The sequence of zero or more octets is for padding purposes.

Note 2: Support will be provided for strings with a length up to 256 octets, except for data records which will support strings with a length up to 32767 octets.

Note 3: The METAFILE DESCRIPTION string parameter will be used to include the sub-string "ISO FCG13" to label the content information as conforming to this agreement. In addition, generator of content are encouraged to append a sub-string that identifies the company and product that produced the CGM.

Note 4: The METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. No part of the element will be partitioned. Multiple occurances of the MDR element may be used to avoid the need for partitioning. The MDR element must appear in the CGM to establish the defaults for TEXT PRECISION and any other elements whose defaults are different than those specified in ISO 8632-1 and -3.

Note 5: Only those character set that are permitted in character content can be specified in geometric graphics content. Refer to 7.1, Document profile, for constraints on character sets in character content.

Note 6: The Scale Factor parameter of SCALING MODE element is always a 32-bit floating point value, even when the REAL PRECISION has selected fixed point for other real numbers. It is not apparent in ISO 8632 what the precision of this floating point value is when fixed point has been selected. Its precision shall be (0,9,23).

Note 7: The minimum support for the length of point lists is 1024 elements.

Note 8: The complete restricted text string, including appended text, shall be included in a metafile conforming to this agreement. The complete restricted text string shall be scaled isotropically such that the specified aspect ratio for the text is not distorted and the string fits into the text extent parallelogram.

Note 9: The minimum support for the length of colour lists parameter for the CELL ARRAY element is 1,048,576. This supports a 1024x1024 image.

Note 10: The PATTERN TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The PATTERN TABLE element shall appear prior to any graphical primitive element to assure that interpreting systems without dynamic pattern update can render the intended effect.

Note 11: The minimum support for the length of the Colour Array parameter for the PATTERN TABLE element is 2048. This will support 8 patterns of 16x16.

Note 12: The COLOUR TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The COLOUR TABLE element shall appear prior to any graphical primitive elements to assure that interpreting systems without dynamic colour update can render the intended effect.

Note 13: The minimum support for the length of the Colour List parameter in the COLOUR TABLE element is 61. This will support a 63 entry colour table.

16.11.3 Interoperability with SGML Applications

The recommended method for the exchange of documents between Standard Generalized Markup Language (ISO 8879, SGML) based systems and systems based on this ODA document application profile is by means of exchanging a document representation conforming to these agreements in an encoded form of the SGML language known as the Office Document Language (ODL). ODL is a standardized SGML application for representing documents conforming to the ODA base standard. Such a representation can be converted into the Office Document Interchange Format (ODIF) supported by these agreements.

17. FUTURE OFFICE DOCUMENT ARCHITECTURE (ODA)

Editor's Note: This section will contain the new text relating to Office Document Architecture (ODA) Agreements.

18 NETWORK MANAGEMENT

18.1 INTRODUCTION

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

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

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

b. keep network resources operational,

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

d. manage the accounting of their usage, and

e. manage their protection from unauthorized use/tampering.

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

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

18.1.1 References

The following documents are referenced in the statements of the agreements relating to OIW 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.

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

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

[CMIP] ISO/IEC 9596-2, Information Processing Systems - Open Systems Interconnection - Management Information Protocol Specification - Part 2: Common Management Information Protocol, 6 December I989.

[CMIS] ISO/IEC 9595-2, Information Processing Systems - Open Systems Interconnection - Management Information Service Definition - Part 2: Common Management Information Service, 6 December 1989.

[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.

Other OSI References:

[ASN1] ISO 8824, Information Processing Systems - Open System Interconnection - Specification of Abstract Syntax Notation One (ASN.1), 19 May 1987.

18.2 SCOPE AND FIELD OF APPLICATION

(Refer to the Working Implementation Agreements Document.)

18.3 STATUS

(Refer to the Working Implementation Agreements Document.)

18.4 ERRATA

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

(Refer to the Working Implementation Agreements Document.)

18.5 MANAGEMENT FUNCTIONS AND SERVICES

(Refer to the Working Implementation Agreements Document.)

18.6 MANAGEMENT COMMUNICATIONS

18.6.1 Association Policies

(Refer to the Working Implementation Agreements Document.)

18.6.2 General Agreements on Users of CMIS

The following agreements pertain to the first phase of network management implementors agreements (IAs). These agreements are based on the standard defined in [CMIS], [CMIP], [ADDRMVP] and [ADDRMVS] constrain the users of CMIS services and not the implementation of CMIP itself.

18.6.2.1 Object Naming

Object Naming will be accomplished using Distinguished Name or Local Distinquished Name. Use of nonspecific form is outside the scope of these agreements.

18.6.2.2 Multiple Object Selection

Multiple Object Selection applies to all management operations except M-EVENT-REPORT, M-CREATE and M-CANCEL-GET.

18.6.2.2.1 Scoping

These network management IAs specify that scoping will be used as specified in [CMIS] 6.5.1, 8.3.1.1.5, 8.3.2.1.6, 8.3.3.1.6, 8.3.5.1.5.

18.6.2.2.2 Filtering

These network management IAs specify that filtering will be used as specified in [CMIS] 6.5.2, 8.3.1.1.6, 8.3.2.1.7, 8.3.3.1.7, 8.3.5.1.6.

If a system receives a filter parameter that it is unable to process, it must return the error "complexity limitation", including the CMISFilter requested.

If, in the process of filtering from a set of selected entities, there are no managed objects selected, the system must return an empty response consisting of an Invoke ID and no response argument.

18.6.2.2.3 Synchronization

In order to support interoperability between managing systems and managed systems, these network management IAs define that the default synchronization (i.e., BestEffort) must be supported by all conforming systems. Atomic synchronization may also be supported as an option.

If a performer is unable to comply with a synchronization request specified by an invoker, the performer must return the error "synchronization not supported" with the parameter indicating the synchronization not supported.

18.6.2.2.4 Multiple Replies

These network management IAs specify the use of multiple replies as specified in [CMIS] 7.1, 7.2.3.

18.6.2.3 Current/Event Time

The time value that is reported to CMIS, if provided, should be as close as possible to, but not before, the actual time that the operation to which the reported time value applies occurred. This constraint is stipulated to provide the most accurate timestamp for temporal ordering of operations and events on a single open system.

For these network management IAs, the encoding of the Current Time parameters is ASN.1 Generalised Time, UTC Type, as specified in [ASN1] clause 32.3, b) and c), with the precision of the time representation indicating the granularity of the time measurement. For example, the string 19890613123012.333-0500 represents a local time of 12:30:12 (and 333 msecs) on 13th June 1989, in a time zone which is 5 hours behind GMT.

18.6.2.4 Access Control

Conformant implementations are not required to use this field. The Access Control field, if provided by the invoker in CMIS request indicators, may be ignored by responding systems which do not support access control. These systems must not reject a request on the basis of the presence of access information. The invoker may interpret this as acceptance of the access control parameter.

18.6.2.5 CMIS Functional Units

Only the Kernel Functional Unit must be supported. Other functional units except Extended Service are optional and their use must be negotiated as specified in [CMIS]. Extended Service is not within the scope of these agreements. Negotiation for its "non use" must be supported.

18.6.2.6 CMIS Parameters

The CMIS globalForm must be used for the following parameters:

action type id
attribute id
event type id
object class

Use of localForm is outside the scope of these agreements.

18.6.3 Specific Agreements on Users of CMIS

These agreements pertain to the first phase of network management implementors agreements (IAs). These agreements are based on the standard defined in [CMIS] and [ADDRMVS]. The agreements in this clause have been defined in terms of those capabilities necessary to support the functions and services defined in clause 18.5 (Management Functions and Services) of these agreements. These agreements constrain the users of CMIS services and not the implementation of CMIP itself.

The parameter presence information in the tables in this clause are repeated from [CMIS] and have the same meaning as in the standard. They are repeated for reader convenience. In addition, these tables provide references to text clauses where agreements are stated relative to parameters in each table. The lack of a reference signifies no agreement beyond the base standard.

18.6.3.1 M-Event-Report

The following agreements and clarifications, pertinent to section 8.2 of the base standard [CMIS] and section 6.3 of the base standard [CMIP] and regarding the M-EVENT-REPORT service, are included within these network management IAs.

Clause 18.5 (Management Functions and Services) of these agreements defines some of the types of Event Reports that may be sent.

18.6.3.1.1 Event Argument

All arguments defined for the particular event type of the managed object class (see sec. 18.7, Management Information Agreements) for the M-EVENT-REPORT must be supplied in the Event Argument parameter. The Event Information and Event Reply parameters shall be supplied as specified for the event type.

18.6.3.1.2 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions defined in [CMIS] unless those conditions are refined by the referenced agreement.

Table 1: M-EVENT REPORT Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Mode	M	-	
3	Managed Object Class	M	U	18.6.2.6
4	Managed Object Instance	M	U	18.6.2.1
5	Event Type	M	C=	18.6.2.6
6	Event Time	U	-	18.6.2.3
7	Event Information	U	-	18.6.3.1.1
8	Current Time	-	U	18.6.2.3
9	Event Reply	-	C	
10	Errors	-	C	18.6.4.4

18.6.3.2 M-Get

The following agreements and clarifications, pertinent to section 8.3.1 of the base standard [CMIS] and section 6.4 of the base standard [CMIP] and regarding the M-GET service, are included within these network management IAs.

18.6.3.2.1 Successful Response

For a successful M-GET operation, the performer shall return (in the Attribute List parameter) either the attribute values for all attributes explicitly requested (in the Attribute Identifier List parameter), or the attribute values for all attributes defined for the managed object(s) selected (if the Attribute Identifier List is omitted).

18.6.3.2.2 Partially Successful or Unsuccessful Response

For a partially successful M-GET operation, where only some attribute values were retrieved, the performer shall return (in the Errors parameter, specifically encoded as GetListError) all attribute ids and their corresponding values that were successfully retrieved from the set of attributes selected as described above, together with all attribute ids, and the corresponding error codes, for each of the attributes for which errors were detected. All attributes requested by the invoker must be processed, with either a value or an error code returned for each.

18.6.3.2.3 Multiple Replies

For the final reply of a series of multiple relies or the single reply where no objects were selected when filtering has been specified, the GetResult is omitted. Hence Managed Object Class, Managed Object Instance, Current Time and Attribute List are all omitted in these cases.

18.6.3.2.4 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions defined in [CMIS] unless those conditions are refined by the referenced agreement.

Table 2: M-GET Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Linked Identifier	-	C	18.6.4
3	Base Object Class	M	-	18.6.2.6
4	Base Object Instance	M	-	18.6.2.1
5	Scope	U	-	18.6.2.2.1
6	Filter	U	-	18.6.2.2.2
7	Access control	U	-	18.6.2.4
8	Synchronization	U	-	18.6.2.2.3
9	Attribute Identifier List	U	-	18.6.2.6
10	Managed Object Class	-	C	18.6.2.6
11	Managed Object Instance	-	C	18.6.2.1
12	Current Time	-	U	18.6.2.3
13	Attribute list	-	C	18.6.2.6, 18.6.3.2.1, 18.6.3.2.3
14	Errors	-	C	18.6.3.2.2, 18.6.4.4

18.6.3.3 M-Set

The following agreements and clarifications, pertinent to section 8.3.2 of the base standard [CMIS] and section 6.5 of the base standard [CMIP] and regarding the M-SET service, are included within these network management IAs.

18.6.3.3.1 Successful Response

For a successful M-SET confirmed operation, the performer shall return (in the Attribute List parameter) the attribute values for all attributes explicitly specified (in the Attribute List parameter) indicating their new values.

18.6.3.3.2 Partially Successful or Unsuccessful Response

For a partially successful M-SET operation, where only some attribute values were modified, the performer shall return (in the Errors parameter, specifically encoded as SetListError) all attribute ids and their corresponding values that were successfully modified from the set of attributes ids and values supplied, and all attribute ids and the corresponding error codes for each of the attributes for which errors were detected. All attributes requested by the invoker must be processed, with either a value or an error code returned for each.

18.6.3.3.3 Multiple Replies

For the final reply of a series of multiple relies or the single reply where no objects were selected when filtering has been specified, the SetResult is omitted. Hence Managed Object Class, Managed Object Instance, Current Time and Attribute List are all omitted in these cases.

18.6.3.3.4 Add/Remove Response

Where multi-valued attributes are involved in an M-SET operation [ADDRMVS], the values returned after any modification operation must be the full set of values of that attribute and not just the values that were modified (e.g., added or removed).

18.6.3.3.5 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions specified in [CMIS] unless those conditions are modified by the referenced agreements.

Table 3: M-SET Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Linked Identifier	-	C	18.6.4
3	Mode	M	-	
4	Base Object Class	M	-	18.6.2.6
5	Base Object Instance	M	-	18.6.2.1
6	Scope	U	-	18.6.2.2.1
7	Filter	U	-	18.6.2.2.2
8	Access control	U	-	18.6.2.4
9	Synchronization	U	-	18.6.2.2.3
10	Managed Object Class	-	C	18.6.2.6
11	Managed Object Instance	-	C	18.6.2.1
12	Modification List	M	-	18.6.2.6, 18.6.3.3.1, 18.6.3.3.3, 18.6.3.3.4
13	Attribute List	-	U	18.6.2.6, 18.6.3.3.1, 18.6.3.3.3
14	Current Time	-	U	18.6.2.3
15	Errors	-	C	18.6.3.3.2, 18.6.4.4

18.6.3.4 M-Action

The following agreements and clarifications, pertinent to section 8.3.3 of the base standard [CMIS] and section 6.6 of the base standard [CMIP] and regarding the M-ACTION service, are included within these network management IAs.

18.6.3.4.1 Multiple Objects

When multiple objects are selected for an M-ACTION operation, there is no ordering implied between selected objects. If the ordering is important, the requesting system may use separate operations, for individual object instances, in the desired order.

18.6.3.4.2 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions defined in [CMIS] unless those conditions are modified by the referenced agreements.

Table 4: M-ACTION Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Linked Identifier	-	C	18.6.4
3	Mode	M	-	
4	Base Object Class	M	-	18.6.2.6
5	Base Object Instance	M	-	18.6.2.1
6	Scope	U	-	18.6.2.2.1
7	Filter	U	-	18.6.2.2.2
8	Managed Object Class	-	C	18.6.2.6
9	Managed Object Instance	-	C	18.6.2.1
10	Access control	U	-	18.6.2.4
11	Synchronization	U	-	18.6.2.2.3
12	Action Type	M	C(=)	18.6.2.6
13	Action Information	U	-	
14	Current Time	-	U	18.6.2.3
15	Action Reply	-	C	
16	Errors	-	C	18.6.4.4

18.6.3.5 M-Create

The following agreements and clarifications, pertinent to section 8.3.4 of the base standard [CMIS] and section 6.7 of the base standard [CMIP] and regarding the M-CREATE service, are included within these network management IAs.

18.6.3.5.1 Managed Object Instance

The Managed Object Instance request parameter may be present or absent depending on whether the invoker supplies the instance name or the performer assigns the instance name automatically. The definition

of each Managed Object Class shall define whether the instance name may be supplied by the invoker, or must be assigned by the performer. This definition shall apply to every management-initiated creation of instances of that managed object class.

18.6.3.5.2 Attribute Values

The values of each of the attributes of the newly created object are derived as defined in [MIM] 5.2.2.1. If none of these methods provides a value for any one attribute, then the operation shall be considered to have failed, i.e., no new instance is created, and the error code 'Missing Attribute Value' shall be returned.

18.6.3.5.3 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions defined in [CMIS] unless those conditions are modified by the referenced agreement.

Table 5: M-CREATE Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Managed Object Class	M	C	18.6.2.6
3	Managed Object Instance	U	C	18.6.2.1, 18.6.3.5.1
4	Superior Object Instance	U	-	18.6.2.1
5	Access Control	U	-	18.6.2.4
6	Reference Object Instance	U	-	18.6.2.1
7	Attribute List	U	C	18.6.2.6, 18.6.3.5.2
8	Current Time	-	U	18.6.2.3
9	Errors	-	C	18.6.3.5.2, 18.6.4.4

18.6.3.6 M-Delete

The following agreements and clarifications, pertinent to section 8.3.5 of the base standard [CMIS] and section 6.8 of the base standard [CMIP] and regarding the M-DELETE service, are included within these Phase 1 network management IAs.

18.6.3.6.1 Deletion of Objects Containing Objects

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

18.6.3.6.2 Parameter Agreements

Conditional parameters (C) shall be included according to the conditions defined in [CMIS] unless those conditions are modified by the referenced agreements.

Table 6: M-DELETE Parameters

Item	Parameter Name	Req/Ind	Rsp/Conf	Text Reference
1	Invoke Identifier	M	M(=)	18.6.4
2	Linked Identifier	-	C	18.6.4
3	Base Object Class	M	-	18.6.2.6
4	Base Object Instance	M	-	18.6.2.1
5	Scope	U	-	18.6.2.2.1
6	Filter	U	-	18.6.2.2.2
7	Access control	U	-	18.6.2.4
8	Synchronization	U	-	18.6.2.2.3
9	Managed Object Class	-	C	18.6.2.6
10	Managed Object Instance	-	C	18.6.2.1
11	Current Time	-	U	18.6.2.3
12	Errors	-	C	18.6.3.6.1, 18.6.4.4

18.6.4 Specific Agreements on CMIP

These agreements pertain to the first phase of network management implementors agreements (IAs). These agreements are based on the standard defined in [CMIP] and [ADDRMVP]. The agreements in this clause have been defined in terms of those capabilities necessary to support the functions and services defined in clause 18.5 (Management Functions and Services) of these agreements.

These network management IAs make no agreements beyond the specifications in [CMIP] and [ADDRMVP], except the following:

18.6.4.1 Invoke/Linked Identifier Size

Invoke Identifiers and Linked Identifiers must be encoded in an integer of four (4) octets maximum length.

18.6.4.2 Version

Protocol Version 2 (only) is supported.

18.6.4.3 Linked Reply Values

Responders must send a linked reply value that corresponds to the original invoke operation value.

18.6.4.4 Error Codes

Responders must send error types that correspond to the operation definition for the original invoke.

18.6.5 Services Required by CMIP

(Refer to the Working Implementation Agreements Document.)

18.7 MANAGEMENT INFORMATION

(Refer to the Working Implementation Agreements Document.)

19. REMOTE DATABASE ACCESS (RDA)

Editor's Note: This section serves as a placeholder for text provided by the newly-formed Remote Database Access (RDA) Special Interest Group.

Table of Contents

List of Tables

Manufacturing Message Specification (MMS)

1 Introduction

(Refer to Working Agreements, Dated September, 1990.)

2 Scope and Field of Application

(Refer to Working Agreements, Dated September, 1990.)

3 References

(Refer to Working Agreements, Dated September, 1990.)

4 Definitions

(Refer to Working Agreements, Dated September, 1990.)

5 Errata

(Refer to Working Agreements, Dated September, 1990.)

6 Status

(Refer to Working Agreements, Dated September, 1990.)

7 General Agreements

(Refer to Working Agreements, Dated September, 1990.)

8 Service-Specific Agreements

(Refer to Working Agreements, Dated September, 1990.)

A Backwards Compatibility Agreements

(Refer to Working Agreements, Dated September, 1990.)

B DIS 9506 Modifications Required For Backwards Compatibility

B.1 Introduction

This annex is an integral part of this part. It documents the modifications to DIS 9506 required to describe implementations for which the agreements of this part provide backwards compatibility. This annex as applied to DIS 9506 is referred to as Version 0.

B.2 References

[1] MMS/1 Manufacturing Message Specification - ISO DIS 9506 - Service Definition, December 1987

[2] MMS/2 Manufacturing Message Specification -ISO DIS 9506 - Protocol Specification, December 1987

[3] NBS OSI Implementors Workshop Agreements - December 1987

B.3 General

B.3.1 Implementation Base

Version 0 is based upon Reference [3] in B.2 as it applies to MMS.

B.3.2 Rules of Extensibility

The following sentence is appended to the last paragraph in section 8.2.1.1.5.2 Proposed Parameter CBB and the last paragraph in section 8.2.1.2.5.2 Negotiated Parameter CBB of DIS 9506-1.

"Any additional bits shall be ignored."

B.4 Modifications to the Protocol definitions

B.4.1 Page 39, Section 7.5.2 of DIS 9506-2

CHANGE

```
reportEventEnrollmentStatus   [60] IMPLICIT ReportEventEnrollmentStatus-Request,
```

TO

```
reportEventEnrollmentStatus   [60] ReportEventEnrollmentStatus-Request,
```

B.4.2 Page 49, Section 7.6.4, DIS 9506-2

CHANGE

```
ApplicationReference ::= SEQUENCE {
 ap-title              ISO-8650-ACSE-1.AP-title OPTIONAL,
 ap-invocation-id      ISO-86 50-ACSE-1.AP-invocation-id OPTIONAL,
 ae-qualifier          ISO-8650-ACSE-1.AE-qualifier OPTIONAL,
 ae-invocation-id      ISO-8650-ACSE-1.AE-invocation-id OPTIONAL
    }
```

TO

```
ApplicationReference ::= SEQUENCE {
 ap-title              [0] OBJECT IDENTIFIER OPTIONAL,
 ap-invocation-id      [1] INTEGER OPTIONAL,
 ae-qualifier          [2] INTEGER OPTIONAL,
 ae-invocation-id      [3] INTEGER OPTIONAL
 }
```

B.4.3 Page 95, Section 12.2.1 of DIS 9506-2

CHANGE

```
structure   [2] IMPLICIT SEQUENCE OF SEQUENCE {
```

TO

```
structure   [2] IMPLICIT SEQUENCE {
```

B.4.4 Page 96, Section 12.3.1 of DIS 9506-2

CHANGE

```
named    [4] IMPLICIT SEQUENCE {
```

TO

```
named    [5] IMPLICIT SEQUENCE {
```

B.4.5 Page 98, Section 12.4.2 of DIS 9506-2

CHANGE

```
generalized-time   [10] IMPLICIT GeneralizedTime,
```

TO

```
generalized-time   [11] IMPLICIT GeneralizedTime,
```

B.4.6 Page 138, Section 15.14 of DIS 9506-2

CHANGE

```
additionalDetail      [9] IMPLICIT EE-Additional-Detail OPTIONAL
```

TO

```
additionalDetail      [9] EE-Additional-Detail OPTIONAL
```

B.4.7 Page 166, Section 17.10 of DIS 9506-2

CHANGE the transfer syntax object identifier value from

```
{ iso  asn1(1)  basic-encoding(1) }
```

TO

```
{ joint-iso-ccitt  asn1(1)  basic-encoding(1) }
```

B.5 Behavioral Requirements

B.5.1 Filenames

File Names are specified in accordance with the NBS Implementors'agreements for FTAM Reference [3] in B.2.

B.5.2 Identify Service

In the Identify service, the vendor, model and revision fields may be of any length, but only the first 64, 16, and 16 octets respectively are treated as significant.

B.5.3 Initiate Service

An MMS Client will:

1. propose 1 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 1 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 Server will:

1. propose 1 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 1 or greater for the value of the Negotiated Max Serv Outstanding Called parameter in the Initiate service when receiving the application association initiation (called).

B.5.3.1 Minimum Segment Size

MMS implementations are able to parse and process 512 octets of MMSpdu as they are encoded in ASN.1 basic encoding rules.

B.5.3.2 Maximum Segment Size

The Max Segment Size is defined as the maximum number of octets in an MMSpdu encoded using the negotiated transfer syntax. This size will apply to all MMSpdu's with the exception of the initiate-Request PDU, initiate-Response PDU, and the initiate-Error PDU. The max segment size will be negotiated during connection initiation using the Proposed Max Segment Size and Negotiated Max Segment Size parameters of the MMS initiate service.

The Max Segment Size will be applied as follows:

Any received MMSpdu which is less than or equal to the Max Segment Size will be properly parsed and processed.

An MMS implementation will not send an MMSpdu whose size exceeds the Max Segment Size.

B.5.4 Abstract Syntax Name

The ASN.1 object identifier value for the abstract syntax name will be the same as specified on page 166, section 17.10 of DIS 9506-2.

B.5.5 Application Context Name

The ASN.1 object identifier value for the application context name will be the same as specified on page 166, section 17.11 of DIS 9506-2.

An MMS implementation ignores the Application Context Name in the A-Associate indication and the A-Associate confirm.

B.5.6 Minor Version Number

The Minor Version Number is zero.

B.6 Parameter CBB Subset

The following subset of MMS Parameter CBBs were considered during preparation of this appendix.

```
STR1,
NEST,
VADR,
VNAM
```

B.7 Service Subset

The following subset of MMS services were considered during preparation of this appendix.

Initiate	Output
Conclude	TakeControl
Cancel	RelinquishControl
Status	ReportSemaphoreStatus
GetNameList	ReportPoolSemaphoreStatus
Identify	ReportSemaphoreEntryStatus
UnsolicitedStatus	CreateProgramInvocation
GetCapabilityList	DeleteProgramInvocation
InitiateDownloadSequence	Start
DownloadSegment	Stop
TerminateDownloadSequence	Resume
InitiateUploadSequence	Reset
UploadSegment	Kill
TerminateUploadSequence	GetProgramInvocationAttributes
RequestDomainDownload	ObtainFile
RequestDomainUpload	GetEventConditionAttributes
LoadDomainContent	ReportEventConditionStatus
StoreDomainContent	GetAlarmSummary
DeleteDomain	ReadJournal
GetDomainAttributes	WriteJournal
Read	InitializeJournal
Write	CreateJournal
InformationReport	DeleteJournal
GetVariableAccessAttributes	ReportJournalStatus
Input	

Table MMS Service Subset

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990 — 281-557/20736

21. REFERENCES

Selected references are grouped by organization publishing the documents and by Reference Model layer to aid relating standards to the OSI Basic Reference Model and to aid relating equivalent standards published by different standards organizations.

21.1 CCITT

General

Application Layer - MHS

CCITT Recommendation X.121 (1988), International Numbering Plan.

CCITT Recommendation I.231, (Blue Book, 1988), Circuit-Mode Bearer Service Categories.

CCITT Recommendation I.232, (Blue Book, 1988), Packet-Mode Bearer Services Categories.

Physical Layer

CCITT Recommendation I.431, (Blue Book, 1988), Primary Rate User-Network Interface-Layer 1 Specification.

Data Link Layer

CCITT Recommendation Q.921 (I.441), (Blue Book, 1988), ISDN User-Network Interfaces: Data Link Layer Specification.

Network Layer

CCITT Recommendation Q.931 (I.451), (Blue Book, 1988), ISDN User-Network Interface Layer 3 Specification for Basic Call Control.

CCITT Recommendation X.25, (Yellow Book, 1980), Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks.

CCITT Recommendation X.25, (Blue Book, 1988), Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit.

CCITT Recommendation X.31, (Blue Book, 1988), Support of Packet Mode Terminal Equipment by an ISDN.

CCITT REcommendation X.213, (Blue Book, 1988), Network Service Definition for Open Systems Interconnection for CCITT Applications.

Transport Layer

CCITT Recommendation X.214, (Blue Book, 1988), Transport Service Definition for Open Systems Interconnection for CCITT Applications.

CCITT Recommendation X.224, (Blue Book, 1988), Transport Protocol Specification for Open Systems Interconnection for CCITT Applications.

Session Layer

CCITT Recommendation X.215, (Red Book, 1984), Session Service Definition for Open Systems Interconnection for CCITT Applications.

CCITT Recommendation X.225, (Red Book, 1984), Session Protocol Profile for Open Systems Interconnection for CCITT Applications.

Presentation

CCITT Recommendation T.50, (Red Book, 1984), International Alphabet No. 5.

CCITT Recommendation T.61, (Red Book, 1984), Graphic Character Sets-Basic Teletex Profiles.

Application Layer -- MHS

CCITT Recommendation X.400, (Red Book, 1984), Message Handling Systems: System Model-Service Elements.

CCITT Recommendation X.401, (Red Book, 1984), Message Handling Systems: Basic Service Elements and Optional User Facilities.

CCITT Recommendation X.408, (Red Book, 1984), Message Handling Systems: Encoded Information Type Conversion Rules.

CCITT Recommendation X.409, (Red Book, 1984), Message Handling Systems: Presentation Transfer Syntax and Notation.

CCITT Recommendation X.410, (Red Book, 1984), Message Handling Systems: Remote Operations and Reliable Transfer Server.

CCITT Recommendation X.411, (Red Book, 1984), Message Handling Systems: Message Transfer Layer.

CCITT Recommendation X.420, (Red Book, 1984), Message Handling Systems: Interpersonal Messaging User Agent Layer.

CCITT Recommendation X.430, (Red Book, 1984), Message Handling Systems: Access Protocol for Teletex Terminals.

CCITT Recommendation X.400 (1988), Message Handling, System and Service Overview.

CCITT Recommendation X.402 (1988), Message Handling Systems, Overall Architecture.

CCITT Recommendation X.407 (1988), Message Handling Systems, Abstract Service Definition Conventions.

CCITT Recommendation X.411 (1988), Message Handling Systems, Message Transfer System: Abstract Service Definition and Procedures.

CCITT Recommendation X.413 (1988), Message Handling Systems, Message Store: Abstract Service Definition.

CCITT Recommendation X.419 (1988), Message Handling Systems, Protocol Specifications.

CCITT Recommendation X.420 (1988), Message Handling Systems, Interpersonal Messaging System.

Application Layer - ODA

CCITT Recommendations

T.6 - Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus 1984

T.4 - Standardization of Group 3 Facsimile Apparatus for Document Transmission 1984

T.411 - Open Document Architecture (ODA) and Interchange Format - Introduction and General Principles 1988

T.412 - Open Document Architecture (ODA) and Interchange Format - Document Structures 1988

T.414 - Open Document Architecture (ODA) and Interchange Format - Document Profile 1988

T.415 - Open Document Architecture (ODA) and Interchange Format - Document Interchange Format (ODIF) 1988

T.416 - Open Document Architecture (ODA) and Interchange Format - Character Content Architectures 1988

T.417 - Open Document Architecture (ODA) and Interchange Format - Raster Graphics Content Architectures 1988

T.418 - Open Document Architecture (ODA) and Interchange Format - Geometric Graphics Content Architectures 1988

T.501 - Document Application Profile MM for the Interchange of Formatted Mixed Mode Documents 1988

T.502 - Document Application Profile PM1 for the Interchange of Processable Form Documents 1988

T.503 - Document Application Profile for the Interchange of Group 4 Facsimile Documents 1988

CCITT documents may be obtained from:
International Telecommunications Union
Place des Nations, CH 1211,
Geneva 20 SWITZERLAND

21.2 ISO

Status of ISO work can be determined by the reference number; working drafts are referenced by committee and number; e.g., TC 97/SC 6 Nxxxx.

Standards are cited by either ISO xxxx or IS xxxx; DIS and DPs are cited in similar form. Note: ISO TC 97 is now called ISO/IEC JTC1.

Information Processing Systems - Open Systems Interconnection - Basic Reference Model. ISO/IS 7498. First Edition - Oct. 15, 1984. Ref. No. ISO 7498-1984(E).

OSI Basic Reference Model - Part 2: Security Architecture. ISO/IS 7498-2-1988(E).

OSI Basic Reference Model - Part 3: Naming and Addressing. ISO/IS 7498-3-1988(E).

Data Interchange - Structure for the identification of organizations. ISO 6523. 1984-02-01.

Physical Layer

Information Processing Systems - Local Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Physical Layer Specification ISO/DIS 8802/3

Information Processing Systems - Local Area Networks - Part 4: Token-Passing Bus Access Method and Physical Layer Specification, ISO/DIS 8802/4

ISO 8802-5 Final Text of ISO/DIS 8802-5: Info proc systems - Local Area Nets -Part 5: Token ring access method and physical layer specification, ISO/TC 97/SC 6 N4477,1987

Information Processing Systems - Fiber Distributed Data Interface (FDDI) - Part 1: Token Ring Physical Layer Protocol (PHY) Requirements; ISO 9314-1 (for Working Agreements Only).

Information Processing Systems - Fiber Distributed Data Interface (FDDI) - Part 2: Token Ring Media Access Control (MAC) Requirements; ISO 9314-2 (for Working Agreements Only).

Information Processing Systems - Fiber Distributed Data Interface (FDDI) - Part 3: Physical Layer Medium Dependent (PMD) Requirements; ISO DIS 9314-3 (for Working Agreements Only).

Data Link Layer

ISO 8802-2 Text for ISO/DIS 8802-2.2, Logical Link Control, ISO/TC 97/SC 6 N4609, 1987

Information Processing Systems - Data Communication - High-Level Data Link Control Procedures - Description of the X.25 LAPB-compatible DTE Data Link Procedures, ISO 7776.

Network Layer

Information Processing Systems - Data Communications - Network Service Definition, ISO 8348.

Information Processing Systems - Data Communications - Network Service Definition: Addendum 1: Connectionless mode Data Transmission, ISO 8348/Add. 1.

Information Processing Systems - Data Communications - Network Service Definition: Addendum 2: Network Layer Addressing, ISO 8348/Add. 2.

Information Processing Systems - Data Communications - Internal Organization of the Network Layer, ISO 8648.

Information Processing Systems - Data Communications - Protocol for Providing the Connectionless-mode Network Service, ISO 8473.

Information Processing Systems - Telecommunications and Information Exchange Between Systems - End System to Intermediate System Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-Mode Network Service (ISO 8473), ISO 9542.

Information Technology - Data Communications - X.25 Packet Layer Protocol for Data Terminal Equipment, ISO 8208 (2nd Edition).

Information Processing Systems - Data Communications - Use of X.25 to provide the OSI Connection-mode Network Service, ISO 8878

Information Processing Systems - Data Communications - Use of the X.25 Packet Layer Protocol in local area networks, ISO 8881.

Information Technology - Telecommunications and Information Exchange Between Systems - Provision of the OSI Connection-Mode Network Service by Packet Mode Terminal Equipment Connected to an Integrated Services Digital Network (ISDN), ISO/IEC 9574.

Information Technology - Telecommunications and Information Exchange Between Systems - Protocol Identification in the Network Layer, ISO/IEC DTR 9577.

Information Processing Systems - Data Communications - Operation of an X.25 Interworking Unit, ISO TR 10029.

Transport Layer

Information Processing Systems - Open Systems Interconnection - Transport Service Definition, ISO 8072.

Information Processing Systems - Open Systems Interconnection - Connection-oriented Transport Protocol Specification, ISO 8073, (2nd Edition).

Information Processing Systems - Open Systems Interconnection - Transport Service Definition - Addendum 1: Connectionless-Mode Transmission, ISO 8072/Addendum 1.

Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification -Addendum 1: Network Connection Management Subprotocol, ISO 8073/Addendum 1.

Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification -Addendum 2: Class Four Operation Over Connectionless Network Service, ISO 8073/Addendum 2.

Information Processing Systems - Open Systems Interconnection - Protocol for Providing the Connectionless-Mode Transport Service, ISO 8602.

Session Layer

Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Service Definition, ISO 8326: 1987 (E).

Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification, ISO 8327 : 1987 (E).

Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Service Definition-AD 2 to ISO 8326 to Incorporate Unlimited User Data, ISO/IEC JTC1/SC21 N2494.

Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification - AD 2 to ISO 8327 to Incorporate Unlimited User Data, ISO/IEC JTC1/SC21 N2495.

Information Processing Systems - Open Systems Interconnection-Session Service Definition: Addendum 3 Covering Connectionless-Mode Session Service, ISO/AD3 8326.

Information Processing Systems - Open Systems Interconnection-Connectionless Session Protocol to Provide the Connectionless-Mode Session Service, ISO/IS 9548.

Presentation Layer

Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Service Definition, ISO 8822: 1988 (ISO/IEC JTC1/SC21 N2335).

Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Protocol Specification, ISO 8823: 1988 (ISO/IEC JTC1/SC21 N2336).

Interim Revised Text of ISO 8825/PDAD1, ASN.1 Extensions (ISO/IEC JTC1/SC21 N2342).

Interim Revised Text of ISO 8824/PDAD1, ASN.1 Extensions (ISO/IEC JTC1/SC21 N2341).

Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), ISO 8824: 1987 (E).

Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation (ASN.1), ISO 8825: 1987 (E).

Information Interchange - Representation of Local Time Differentials, ISO 3307.

Information Processing Systems - Open Systems Interconnection - Presentation Service Definition: Draft Addendum 1 Covering Connectionless-Mode Presentation Service, ISO/DAD1 8822: 1989-02-15(e) (ISO/IEC JTC1/SC21 N 3171).

Information Processing Systems - Open Systems Interconnection - Connectionless Presentation Protocol to Provide the Connectionless-Mode Presentation Service, ISO/IS 9576: 1989-02-25 5(E) (ISO/IEC JTC1/SC21 N 3172).

Application Layer

Information Processing Systems - Open Systems Interconnection - Application Layer Structure, ISO/DP 9545, ISO/TC97/SC21/N1743. July 24, 1987. Revised November 1987.

ISO 10021-1 Information Processing Systems - Text Communication - MOTIS - System and Service Overview.

ISO 10021-2 Information Processing Systems - Text Communication - MOTIS - Overall Architecture.

ISO 10021-3 Information Processing Systems - Text Communication - MOTIS - Abstract Service Definition Conventions.

ISO 10021-4 Information Processing Systems - Text Communication - MOTIS - Message Transfer System: Abstract Service Definition and Procedures.

ISO 10021-5 Information Processing Systems - Text Communication - MOTIS - Message Store: Abstract Service Definition.

ISO 10021-6 Information Processing Systems - Text Communication - MOTIS - Protocol Specifications.

ISO 10021-7 Information Processing Systems - Text Communication - MOTIS - Interpersonal Messaging System.

Application Layer -- FTAM

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part I: General Introduction, ISO 8571-1: 1988(E).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 2: Virtual Filestore Definition, ISO 8571-2: 1988(E).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 3: The File Service Definition, ISO 8571-3: 1988(E).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 4: File Protocol Specification, ISO 8571-4: 19878(E).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 5: Protocol Information Conformance Statement Proforma, ISO 8571-5: 1990(E).

Application Layer -- ASE/ACSE

Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element, ISO 8649: 1987 (E) (ISO/IEC JTC1/SC21 N2326)

Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element, ISO 8650: 1987 (E) (ISO/IEC JTC1/SC21 N2327)

Information Processing System - Open Systems Interconnection - ACSE Service Definition: Draft Addendum 2 Covering Connectionless-Mode ACSE Service, ISO 8649/DAD2.

Information Processing Systems - Open Systems Interconnection - Connectionless ACSE Protocol to Provide the Connectionless-Mode ACSE Service, ISO IS 10035: 1989-02-25 (ISO/IEC JTC1/SC21 N 3456).

Application Layer -- VTP

Information Processing Systems - Open Systems Interconnection - Virtual Terminal Service - Basic Class, IS 9040.

Information Processing Systems - Open Systems Interconnection - Virtual Terminal Protocol - Basic Class, IS 9041.

Application Process -- Office Document Interchange -- ODA

ISO 8613/1 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 1: Introduction and General Principles

ISO 8613/2 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 2: Document Structures

ISO 8613/4 - Information processing : Text and Office Systems; Office

Document Architecture (ODA) and Interchange Format Part 4: Document Profile

ISO 8613/5 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 5: Office Document Interchange Format

ISO 8613/6 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 6: Character Content Architectures

ISO 8613/7 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 7: Raster Graphics Content Architectures

ISO 8613/8 - Information processing : Text and Office Systems; Office Document Architecture (ODA) and Interchange Format Part 8: Geometric Graphics Content Architectures

Application Process -- Computer Graphics -- CGM/GKS

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 1; Functional Specification, IS 8632/1

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 2; Character Encoding, IS 8632/2

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 3; Binary Encoding, IS 8632/3.

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 4; Clear Text Encoding, IS 8632/4.

Information Processing Systems - Font and Character Information Interchange, IS 9541.

Information Processing Systems - 8-Bit Single Byte Coded Graphic Character Sets, Part 1; Latin Alphabet Part 1, IS 8859/1.

Information Processing Systems - Computer Graphics Functional Specification of the Graphical Kernel System (GKS), IS 7942.

Information Processing Systems - Computer Graphics - Graphical Kernel System for Three Dimensions (GKS-3D), Functional Description, DIS 8805.

Information Processing Systems - Computer Graphics - Programmers Hierarchical Interactive Graphics System (PHIGS), DP 9592.

Information Processing Systems - Computer Graphics - Interfacing Techniques for Dialogues with Graphical Devices (CGI),ISO TC 97/SC 21 N 1179.

Application Layer -- Directory Services

The Directory--Overview of Concepts, Models, and Services (CCITT Recommendation X.500, ISO 9594)

The Directory--Information Framework (CCITT Recommendation X.501, ISO 9594)

The Directory--Access and System Services Definition (CCITT Recommendation X.511, ISO 9594)

The Directory--Procedures For Distributed Operation (CCITT Recommendation X.518, ISO 9594)

The Directory--Access and System Protocols Specification (CCITT Recommendation X.519, ISO 9594)

The Directory--Selected Attribute Types (CCITT Recommendation X.520, ISO 9594)

The Directory--Selected Object Classes (CCITT Recommendation X.521, ISO 9594)

The Directory--Authentication Framework (CCITT Recommendation X.509, ISO 9594)

Remote Operations-Part 1: Model, Notation and Service Definition (CCITT Recommendation X.219, ISO 9072 Version 5)

Remote Operations-Part 2: Protocol Specification (CCITT Recommendation X.229, ISO 9072 Version 5)

Association Control-Service Definition (CCITT Recommendation X217, ISO 8649)

Association Control-Protocol Definition (CCITT Recommendation X.217, ISO 8650)

Note: ISO 9594, 9072 are preferred texts (over the CCITT counterparts) and are to be taken as of Gloucester, Nov. 1987.

Additional ISO References

ISO 8859-1 - Information Processing - 8-bit Single-byte Coded Graphic Character Sets - Part 1: Latin Alphabet No. 1

ISO 8859-2 - Information Processing - 8-bit Single-byte Coded Graphic Character Sets - Part 2: Latin Alphabet No. 2

DIS 8859-3 - Information Processing - 8-bit Single-Byte Coded Graphic Character Sets - Part 3: Latin Alphabet No. 3

ISO 6937-1 - Information Processing - Coded Character Sets for Text Communication - Part 1: General Introduction

ISO 6937-2 - Information Processing - Coded Character Sets for Text Communication - Part 2: Latin Alphabetic and Non-Alphabetic Graphic Characters

ISO 2022 - Information Processing - ISO 7-Bit and 8-Bit Coded Character Sets - Code Extension Techniques

7-Bit Coded Character Set for Information Processing Interchange, ISO 646.

ISO 6429: 1988 Information Processing - ISO 7-Bit and 8-Bit Coded Character Sets - Additional Control Functions for Character-Imaging Devices.

ISO/IEC ISP AFTnn-Information Technology ~~Processing-Systems~~ - International Standardized Profiles AFTnn - File Transfer, Access and Management.

Part 1 Specification of ACSE, Presentation and Session Protocols for the use by FTAM PDISP ~~1989-06-01~~. 1990-01-10.

Part 2 Definition of Document Types, Constraint Sets and Syntaxes PDISP ~~1989-06-01~~. 1990-01-10.

Part 2 Definition of Document Types, Constraint Sets and Syntaxes, Addendum 1: Additional Defintions Working Draft ~~1989-08-15~~. 1990-01-10.

Part 3 AFT11-Simple File Transfer Service (Unstructured) PDISP ~~1989-06-01~~. 1990-01-10.

Part 4 AFT12-Positional File Transfer Service (Flat) Working Draft ~~1989-08-15~~. 1990-01-10.

Part 5 AFT22-Positional File Access Service (Flat) Working Draft ~~1989-08-15~~. 1990-01-10.

Part 6 AFT3-File Management Service Working Draft ~~1989-08-15~~. 1990-01-10.

ISO documents may be obtained from:

Frances E. Schrotter
ANSI
ISO TC 97/SC 6 Secretariat
1430 Broadway
New York, NY. 10018

21.3 Additional References

INTAP-S007-03 - Multi-media (ODA/ODIF) Functional Standard Detail Description (AE.111n-J), Small Function Set Profiles of Document Interchange Format with Layout Information

INTAP-S007-04 - Multi-media (ODA/ODIF) Functional Standard Detail Description (AE.112n-J), Medium Function Set Profiles of Document Interchange Format with Layout Information

21.4 IEEE

Physical Layer

IEEE Standard for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Physical Layer Specification, ANSI/IEEE Standard 802.3 1985, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

IEEE Standard for Local Area Networks: Supplements to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification ANSI.IEEE Standard 802.3a, b, c, e -1988

Supplement a, MAU and Baseband Medium Specification, Type 10BASE2 (Section 10)

Supplement a, Broadband MAU & Medium Specification, Type 10BROAD36 (Section 11)

Supplement c, Repeater Set and Repeater Unit Specification for Use with 10BASE5 and 10 BASE2 Networks

Supplement e, Physical Signaling, Medium Attachment, and Baseband Medium Specification, Type 10BASE5

IEEE Standard for Local Area Networks: Token-Passing Bus Access Method and Physical Layer Specification, ANSI/IEEE Standard , 802.4 Draft 1987, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

IEEE Standard for Local Area Networks: Token-Ring Access Method, ANSI/IEEE Standard 802.5- 1986, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

Data Link Layer

IEEE Standard for Local Area Networks: Logical Link Control, ANSI/IEEE Standard 802.2 - 1987, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

21.5 NBS

Implementation Agreements for Open Systems Interconnection Protocols: NBS Workshop for Implementors of Open Systems Interconnection, National Bureau of Standards, NBSIR 86-3385-6, Robert Rosenthal, Editor, Revised July 1987.

U. S. Government Open Systems Interconnection Profile (GOSIP), National Bureau of Standards, Institute for Computer Sciences and Technology, 1987.

NBS documents may be obtained from:

NTIS
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA. 22161.

21.6 MAP

Manufacturing Automation Protocol, General Motors Corporation,Manufacturing Engineering and Development, Advanced Product and Manufacturing Engineering Staff (APMES), APMES A/MD-39, GM Technical Center, Warren, MI. 48090-9040.

21.7 TOP

Technical and Office Protocols Specification Version 3.0, MAP/TOP Users Group, Attention TOP 3.0 Document. One SME Drive, P.O. Box 930, Dearborn Mi. 48121.

21.8 CEN/CENELEC

ENV 41204: 1988(E) ENV 41204 (revised): 1990(E) FTAM Simple File Transfer (Unstructured), June Jan. 1990 (proposed revision June 1989).

prENV 41205 FTAM File Management, March Nov. 1989.

prENV 41206 FTAM Positional File Transfer (Flat), June 1989 Jan. 1990.

prENV 41207 FTAM Postional File Access (Flat), June 1989 Jan. 1990.

ENV 41201 "Private Message Handling Systems"

ENV 41202 "Public Message Handling Systems"

21.9 SPAG

Guide to the Use of Standards, Revision 4.0 1989.

21.10 ANSI

Physical Layer

Fiber Distributed Data Interface (FDDI) - Token Ring Media Access Control (MAC); ANS X3.139-1987 (for Working Agreements only).

Fiber Distributed Data Interface (FDDI) - Token Ring Physical Layer Protocol (PHY); ANS X3.148-1988 (for Working Agreements only).

Fiber Distributed Data Interface (FDDI) - Physical Layer Medium Dependent (PMD); ANS X3.166-1989 (for Working Agreements only).

Carrier-to-Customer Installation - DS1 Metallic Interface, ANS T1.403 - 1989.

Integrated Services Digital Network - Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT-Layer 1 Specification, ANS T1.601 - 1988.

Integrated Services Digital Network - Basic Access Interface at S and T Reference Points - Layer 1 Specification, ANS T1.605 - 1988.

Data Link Layer

Telecommunications - Integrated Services Digital Network (ISDN) - Data-Link Layer Signaling Specification for Application at the User-Network Interface, ANS T1.602 - 1989.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990—261-913 20604

12-4-90 MMW

March 1990 (Stable)

CHANGE PAGE INDEX
VERSION 3 EDITION 1, MARCH 1990 (STABLE) CHANGE PAGES
ISSUED JUNE 1990
OUTPUT FROM MARCH 1990 OSI WORKSHOP

SUPPLEMENT TO STABLE IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS, VERSION 3, EDITION 1, DECEMBER 1989 NIST SPECIAL PUBLICATION 500-177

This document records, in replacement page format, all changes to stable material current (according to Version and Edition number) as of the previous Workshop (December 15, 1989). In this case, that would be NIST SP 500-177, Version 3, Edition 1. By following the instructions below, and replacing or inserting the indicated pages, a new "Edition" will be created which reflects new the current status of relevant stable material as of March 16, 1990.

If there are sidebars in the Table of Contents, then that document should contain the replacement pages for March 1990. **THIS INSERT SET SHOULD BE SAVED.**

The following table gives all necessary information. Entries in the first column indicate that those pages are to be replaced with the pages referenced in the second column.

The instructions below are given by Chapter and Page number, referring to NIST SP 500-177. Page numbers with extra digits are "overflow" pages to be inserted immediately after the previously-mentioned page. The changes made are indicated by vertical bars on the left margins of replacement pages, and all the replacement pages are dated in the upper right-hand corner for easy identification.

Changes on replacement pages are Errata (technical, alignment, editorial, other). The term "other" may include addition of new stable functionality. All changes apply to the previous version and edition. Replacements reflect the following types of changes:

1. deletion of material (blank lines)
2. insertion of new material (may lead to "overflow" pages), and
3. replacement of old text with new text (combination of (1) and (2) above.

Technical errata are changes from implementor experience which materially affect the meaning or semantics of a section, and editorial changes do not change semantics. Alignment changes are material changes made in the interest of international harmonization and evolving base standards.

In general, technical errata are more "severe" than editorial errata, and alignment errata are more "severe" than technical errata. If a

particular replacement has a combination of errata, the most "severe" errata will be checked in the replacement page table. Readers should keep in mind that these errata may be further categorized in future editions.

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
iii-xxvii	iii-xxvii			X	
1-1, 1-2	1-1, 1-2			X	
2-1 - 2-2	2-1, 2-2				X
Chapter 5	Chapter 5			X	
6-3 thru 6-6	6-3 thru 6-6			X	
8-5, 8-6	8-5, 8-6			X	
8-29, 8-30	8-29, 8-30		X		
8-33, 8-34	8-33, 8-34		X		
8-45, 8-46	8-45, 8-46		X		
9-3, 9-4	9-3, 9-3-1			X	
insert	9-4			X	
9-7 thru 9-12	9-7 thru 9-12			X	
9-31, 9-32	9-31, 9-31-1			X	
insert	9-32				
9-41, 9-42	9-41, 9-42	X			
insert	9-42-1	X			
9-43, 9-44	9-43, 9-44	X			
9-47 thru 9-50	9-47 thru 9-50	X			
9-53 thru 9-56	9-53 thru 9-56	X			
9-59 thru 9-62	9-59 thru 9-62	X			
insert	9-62-1	X			
9-63, 9-64	9-63, 9-64	X			
9-65, 9-66	9-65, 9-65-1	X			

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
insert	9-66	X			
9-67, 9-68	9-67, 9-68	X			
10-1, 10-2	10-1, 10-2			X	
10-3, 10-4	10-3, 10-4			X	
10-5, 10-6	10-5, 10-6			X	
10-11, 10-12	10-11, 10-12	X			
10-13, 10-14	10-13, 10-14			X	
10-15, 10-16	10-15, 10-16			X	
10-19, 10-20	10-19, 10-20	X			
10-49, 10-50	10-49-10-50	X			
10-63, 10-64	10-63, 10-64	X			
10-65 thru 10-67	10-65 thru 10-67	X			
10-67, 10-68	10-67, 10-68			X	
10-71, 10-72	10-71, 10-72	X			
10-73, 10-74	10-73, 10-74			X	
10-77, 10-78	10-77, 10-78			X	
10-83, 10-84	10-83, 10-84			X	
10-85 thru 10-86	10-85 thru 10-86			X	

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
21-5, 21-6	21-5, 21-6			X	
21-7, 21-8	21-7, 21-8			X	
21-9, 21-10	21-9, 21-10			X	
21-11, 21-12	21-11, 21-12			X	
21-13 thru 21-15	21-13 thru 21-15			X	

Please retain this change page index in the back of your document.

Thank you.

12-5-90
NWW

June 1990 Stable

CHANGE PAGE INDEX
VERSION 3, JUNE 1990 (STABLE) CHANGE PAGES
ISSUED SEPTEMBER 1990
OUTPUT FROM JUNE 1990 OSI WORKSHOP

SUPPLEMENT TO STABLE IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS, VERSION 3, MARCH 1990 NIST SPECIAL PUBLICATION 500-177

This document records, in replacement page format, all changes to stable material current (according to Version and Edition number) as of the previous Workshop (March 16, 1990). In this case, that would be NIST SP 500-177, Version 3, Edition 1 with previous Change Pages incorporated. By following the instructions below, and replacing or inserting the indicated pages, text will be created which reflects the current status of relevant stable material as of June 22, 1990.

If there are sidebars in the Table of Contents, then that document should contain the replacement pages for June 1990. **THIS INSERT SET SHOULD BE SAVED.**

The following table gives all necessary information. Entries in the first column indicate that those pages are to be replaced with the pages referenced in the second column.

The instructions below are given by Chapter and Page number, referring to NIST SP 500-177. Page numbers with extra digits are "overflow" pages to be inserted immediately after the previously-mentioned page. The changes made are indicated by vertical bars on the left margins of replacement pages, and all the replacement pages are dated in the upper right-hand corner for easy identification.

Changes on replacement pages are Errata (technical, alignment, editorial, other). The term "other" may include addition of new stable functionality. All changes apply to the previous version and edition. Replacements reflect the following types of changes:

1. deletion of material (blank lines)
2. insertion of new material (may lead to "overflow" pages), and
3. replacement of old text with new text (combination of (1) and (2) above.

Technical errata are changes from implementor experience which materially affect the meaning or semantics of a section, and editorial changes do not change semantics. Alignment changes are material changes made in the interest of international harmonization and evolving base standards.

In general, technical errata are more "severe" than editorial errata, and alignment errata are more "severe" than technical errata. If a

particular replacement has a combination of errata, the most "severe" errata will be checked in the replacement page table. Readers should keep in mind that these errata may be further categorized in future editions.

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
iii-xxvii	iii-xxviii			X	
1-1, 1-2	1-1, 1-2			X	
3-1 thru 3-18	3-1 thru 3-19	X			
5-3, 5-4	5-3, 5-4		X		
5-7, 5-8	5-7, 5-8		X		
5-11, 5-12	5-11, 5-12		X		
7-1 thru 7-4	7-1 thru 7-4			X	
Blank-9-4	Blank-9-4			X	
9-9, 9-10	9-9, 9-10			X	
insert	9-10-1				
9-41, 9-42	9-41, 9-42			X	
9-63, 9-64	9-63, 9-64			X	
9-65, 9-65-1	9-65, 9-65-1			X	
11-25, 11-26	11-25, 11-25-1			X	
insert	Blank, 11-26			X	
14-1 thru 14-75	14-1 thru 14-77		X		
15-1, blank	15-1, blank			X	
18-1, blank	18-1 thru 18-13				X
21-7, 21-8	21-7, 21-8		X		
21-9, 21-10	21-9, 21-10		X		

Please retain this change page index in the back of your document.

Thank you.

12-5-90
MAN

September 1990 Stable

CHANGE PAGE INDEX
VERSION 3, JUNE 1990 (STABLE) CHANGE PAGES
ISSUED DECEMBER 1990
OUTPUT FROM SEPTEMBER 1990 OSI WORKSHOP

SUPPLEMENT TO STABLE IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS, VERSION 3, SEPTEMBER 1990 NIST SPECIAL PUBLICATION 500-177

This document records, in replacement page format, all changes to stable material current (according to Version and Edition number) as of the end of the June Workshop. In this case, that would be NIST SP 500-177, Version 3, Edition 1 with previous Change Pages incorporated. By following the instructions below, and replacing or inserting the indicated pages, text will be created which reflects the current status of relevant stable material as of September 14, 1990.

If there are sidebars in the Table of Contents, then that document should contain the replacement pages for September 1990. **THIS INSERT SET SHOULD BE SAVED.**

The following table gives all necessary information. Entries in the first column indicate that those pages are to be replaced with the pages referenced in the second column.

The instructions below are given by Chapter and Page number, referring to NIST SP 500-177. Page numbers with extra digits are "overflow" pages to be inserted immediately after the previously-mentioned page. The changes made are indicated by vertical bars on the left margins of replacement pages, and all the replacement pages are dated in the upper right-hand corner for easy identification.

Changes on replacement pages are Errata (technical, alignment, editorial, other). The term "other" may include addition of new stable functionality. All changes apply to the previous version and edition. Replacements reflect the following types of changes:

1. deletion of material (blank lines)
2. insertion of new material (may lead to "overflow" pages), and
3. replacement of old text with new text (combination of (1) and (2) above.

Technical errata are changes from implementor experience which materially affect the meaning or semantics of a section, and editorial changes do not change semantics. Alignment changes are material changes made in the interest of international harmonization and evolving base standards.

In general, technical errata are more "severe" than editorial errata, and alignment errata are more "severe" than technical errata. If a

particular replacement has a combination of errata, the most "severe" errata will be checked in the replacement page table. Readers should keep in mind that these errata may be further categorized in future editions.

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
Table of Contents	Table of Contents			X	
Chapter 1	Chapter 1				X
2-1 -2-2	2-1, 2-2			X	
3-1, 3-2	3-1, 3-2			X	
4-1, 4-2	4-1, 4-2			X	
5-3, 5-4	5-3, 5-4			X	
5-13, 5-14	5-13, 5-14			X	
	insert 5-14-1			X	
5-25, 5-26	5-25, 5-26			X	
5-27, 5-28	5-27, 5-28			X	
8-3, 8-4	8-3, 8-4			X	
	insert 8-4-1			X	
8-9, 8-10	8-9, 8-10			X	
8-21, 8-22	8-21, 8-22			X	
8-59, 8-60	8-59, 8-60			X	
8-65, 8-66	8-65, 8-66			X	
blank, 9-4	blank, 9-4			X	
11-25, 11-25-1	11-25, 11-25-1	X			
11-61, 11-62	11-61, 11-62	X			
11-63, 11-64	11-63, 11-64				X
11-65, 11-66	11-65, 11-66	X			
11-67, 11-68	11-67, 11-68				X
11-69, 11-70	11-69, 11-70	X			

DELETE PAGE NUMBER	INSERT PAGE NUMBER	TECHNICAL	ALIGNMENT	EDITORIAL	OTHER
11-71, 11-72	11-71, 11-72				X
14-3, 14-4	14-3, 14-4			X	
14-13-14-20	14-13-14-20		X		
	insert 14-20-1				
14-53, 14-54	14-53, 14-54		X		
	insert 14-54-1		X		
14-57-14-60	14-57-14-60			X	
	insert 14-60-1			X	
14-65, 14-66	14-65, 14-66			X	
Chapter 20	Chapter 20				X

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-500/177

2. PERFORMING ORGANIZATION REPORT NUMBE

3. PUBLICATION DATE
March 1990

4. TITLE AND SUBTITLE

STABLE IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS
Version 3 Edition 1 December 1989

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
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Same as Item #6

10. SUPPLEMENTARY NOTES

Supersedes NIST/SP-500/162

[ ] 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 OI LITERATURE SURVEY, MENTION IT HERE.)

This document records current Stable Agreements for Open Systems Interconnection Protocols among the organizations participating in the NIST/OSI Workshop Series for Implementors of OSI Protocols.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Local Area Networks; Network Protocols; NIST/OSI Workshop; Open Systems Interconnection; Testing Protocols

13. AVAILABILITY

[XX] UNLIMITED

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

[XX] 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
679

15. PRICE

ELECTRONIC FORM

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990 — 261 - 913/ 20518

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899