NIST Special Publication 500-206

Computer Systems Technology

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

NIST

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 6 Edition 1 December 1992

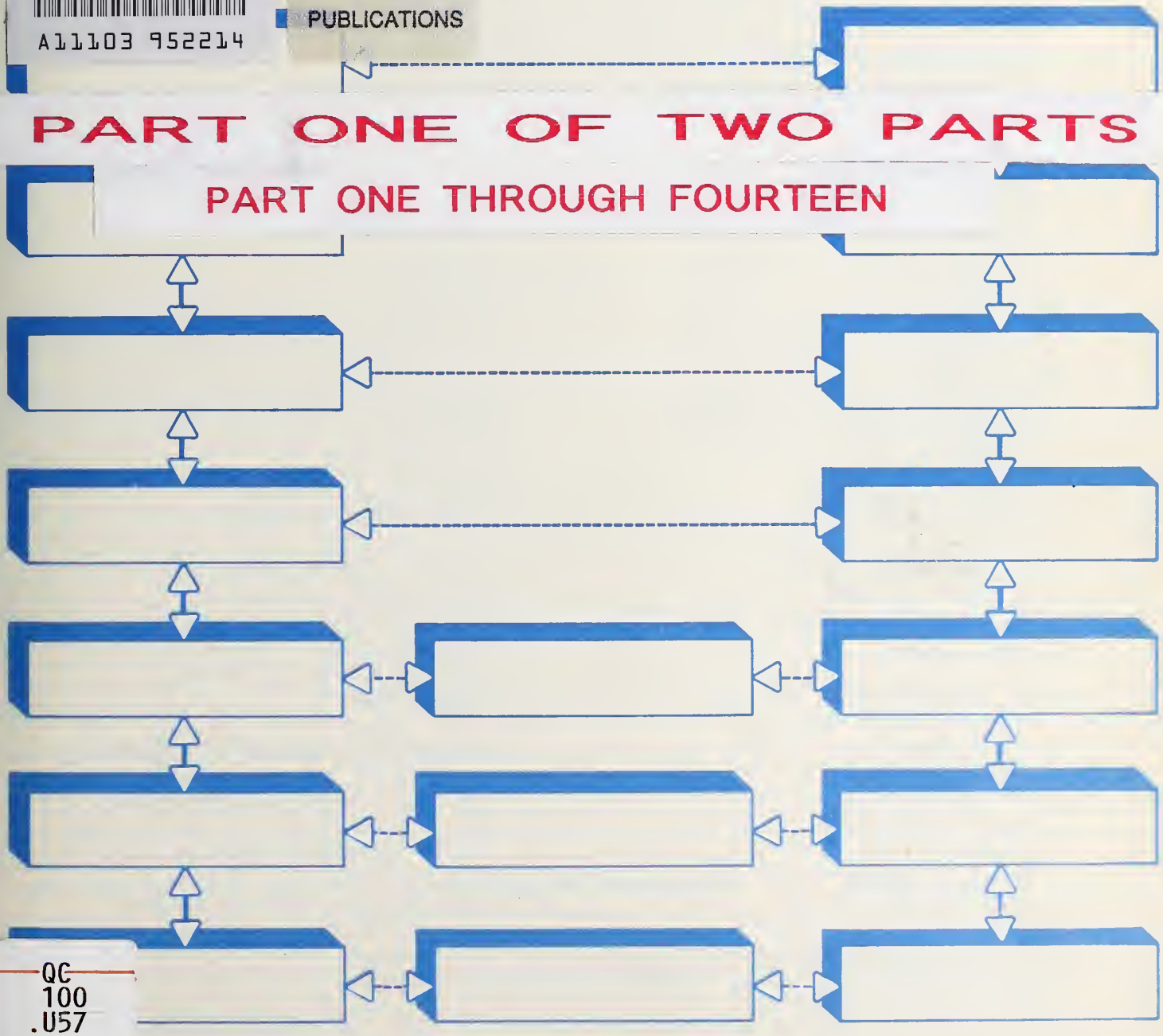
Based on the Proceedings of the OSE Implementors' Workshop (OIW)

Workshop Chairman/Technical Editor
Tim Boland, NIST

Workshop Editor
Brenda Gray

PART ONE OF TWO PARTS

PART ONE THROUGH FOURTEEN



The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering and performs related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Public Inquiries Desk, 301-975-3058.

Technology Services
- Manufacturing Technology Centers Program
- Standards Services
- Technology Commercialization
- Measurement Services
- Technology Evaluation and Assessment
- Information Services

Electronics and Electrical Engineering Laboratory
- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]

Chemical Science and Technology Laboratory
- Biotechnology
- Chemical Engineering[1]
- Chemical Kinetics and Thermodynamics
- Inorganic Analytical Research
- Organic Analytical Research
- Process Measurements
- Surface and Microanalysis Science
- Thermophysics[2]

Physics Laboratory
- Electron and Optical Physics
- Atomic Physics
- Molecular Physics
- Radiometric Physics
- Quantum Metrology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Manufacturing Engineering Laboratory
- Precision Engineering
- Automated Production Technology
- Robot Systems
- Factory Automation
- Fabrication Technology

Materials Science and Engineering Laboratory
- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- Reactor Radiation

Building and Fire Research Laboratory
- Structures
- Building Materials
- Building Environment
- Fire Science and Engineering
- Fire Measurement and Research

Computer Systems Laboratory
- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

Computing and Applied Mathematics Laboratory
- Applied and Computational Mathematics[2]
- Statistical Engineering[2]
- Scientific Computing Environments[2]
- Computer Services[2]
- Computer Systems and Communications[2]
- Information Systems

[1] At Boulder, CO 80303.
[2] Some elements at Boulder, CO 80303.

NIST Special Publication 500-206

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 6 Edition 1 December 1992

Based on the Proceedings of the OSE Implementors' Workshop (OIW)

Workshop Chairman/Technical Editor
Tim Boland, NIST

Workshop Editor
Brenda Gray

Special Interest Group Contacts

Conformance Testing	Eva Kuiper	Hewlett Packard
Directory Services	Kenneth J. Rossen	SHL Systemhouse
File Transfer Access Mgmt.	Joe Mohen	Proginet
Health Care	John J. Harrington	Hewlett Packard
Library Applications	Ray Denenberg	Library of Congress
Lower Layers	Fred Burg	AT&T
Message Manufacturing Spec.	John Baier	Allen-Bradley
Network Management	Paul J. Brusil	The Mitre Corp.
	George Mouradian	AT&T Bell Labs
Office Document Architecture	Jim Wing	IBM
Open Systems Environment/TLC	Jerry Johnson	State of Texas
Remote Database Access	Peter Eng	IBM Canada
Security	James M. Galvin	Trusted Info. Systems
Technical Liaison Committee	Einar Stefferud	Network Management Associates, Inc.
Transaction Processing	Jeff Hildebrand	Boeing Computer Services
Upper Layers	Jim Quigley	Hewlett Packard
Virtual Terminals	Luke Lucas	Control Data Systems, Inc.
X.400	Neil K. Koorland	Microsoft Corp.

Supersedes NIST/SP-500/202

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Issued March 1993

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) has a unique responsibility for computer systems technology within the Federal government. NIST's Computer Systems Laboratory (CSL) 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. CSL'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. CSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports CSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-206
Natl. Inst. Stand. Technol. Spec. Publ. 500-206, 1767 pages (Mar. 1993)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1993

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

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 1 - General Information

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

Chair/Editor, OIW Workshop: **Tim Boland, NIST**
Workshop Editor: **Brenda Gray, NIST**

Foreword

This part of the Stable Implementation Agreements was prepared by the Chair of the Open Systems Environment Implementors' Workshop (OIW).

This part replaces the previously existing chapter on this subject. There is no significant technical change from this text as previously given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

Part 1 - General Information

0 Introduction

This document records current stable implementation agreements of OSI protocols among the organizations participating in the Open Systems Environment Implementors' Workshop (OIW). Stable in the context of this document means the following:

a) 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;

b) the agreements have been approved by the OSE Implementors' Workshop (OIW) 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 OSE Implementors' Workshop (OIW) 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 designated portions of each Part.

1 Scope

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 part 1). Modifications to 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 part of the document covers a different subject area, and the parts are presented so as to present a consistent and unified approach. The format of this version follows the ISO directives whenever possible.

The corresponding and aligned document, "Working Implementation Agreements for OSI Protocols," 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. These errata will only be applied to the current version.

Currently efforts are under way to define worldwide technically harmonized profiles. The goal is to create a consolidated global market for OSI products. This means that vendors can sell to a larger market, and users can procure products from a variety of vendors around the world. Agreements in this document are likely to be used in these alignment efforts.

This version is backwards compatible with the previous version to the maximum extent possible. This version includes all of the material from the previous version (modified by errata) as well as new stable material from the previous year.

2 Normative References

See succeeding parts.

3 Definitions

See succeeding parts.

4 Purpose of the Workshop

In February, 1983, at the request of industry, NIST organized the OSI Implementors' Workshop (OIW) 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 and OSE protocols and promote interoperability of independently manufactured data communications equipment. Recently the Workshop was renamed the Open Systems Environment Implementors' Workshop (OIW).

5 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
OSE Implementors' Workshop (OIW)
Building 225, Room B-217
Gaithersburg, Maryland 20899

6 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 clause of the Working Agreements Document for more on this subject.

7 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.

8 Structure and Operation of Workshop

8.1 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 clause of the Working Agreements Document.

8.2 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 clause of the Working Agreements Document.

9 Points of Contact

For information concerning the workshop, write to:

Chair, OSE Implementors' Workshop (OIW), at the address given in 1.3.

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

10 Profile Conformance

NOTE - SIG text relating to text below may be given in some succeeding parts.

This clause presents general concepts for profile conformance. These concepts shall be observed when writing implementation Agreements.

10.1 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.

10.2 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.

10.2.1 Sending/Encoding Entity

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

10.2.2 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).

10.3 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:

a) m mandatory;

b) o optional;

c) c conditional;

d) x excluded;

e) i out of scope;

f) - not applicable.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 2 - Subnetworks

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Fred Burg, AT&T**
SIG Editor: **Brenda Gray, NIST**

Foreword

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

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

Annexes A and B are for information only.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 2 - Subnetworks

0 Introduction

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

1 Scope

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

2 Normative References

2.1 CCITT

[1] Recommendation I.231 (Blue Book, 1988), *Circuit-Mode Bearer Service Categories.*

[2] Recommendation I.232 (Blue Book, 1988), *Packet-Mode Bearer Service Categories.*

[3] Recommendation I.431 (Blue Book, 1988), *Primary Rate User-Network Interface - Layer 1 Specification.*

[4] Recommendation Q.921 (I.441) (Blue Book, 1988), *ISDN User-Network Interface, Data Link Layer Specification.*

[5] CCITT Recommendation Q.922, *ISDN Data Link Layer Specification for Frame Mode Bearer Services, ITU*, Geneva, (1991).

[6] Recommendation Q.931 (I.451) (Blue Book, 1988), *ISDN User-Network Interface Layer 3 Specification for Basic Call Control.*

[7] CCITT Recommendation Q.933, *ISDN Signaling Specification for Frame Mode Bearer Services, ITU*, Geneva (1992).

[8] 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.*

[9] 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.*

[10] Recommendation X.31 (Blue Book, 1988), *Support of Packet Mode Terminal Equipment by an ISDN.*

2.2 ISO

[11] ISO 7776, *Information processing systems - Data communication - High-level Data Link Control Procedures - Description of the X.25 LAPB-compatible DTE data link procedures.*

[12] ISO/IEC 8208 Edition 2, *Information technology - Data communications - X.25 Packet layer protocol for data terminal equipment.*

[13] ISO 8802-2, *Information processing systems - Local area networks - Part 2: Logical link control.*

[14] ISO DIS 8802-3, *Information processing systems - Local area networks - Part 3: Carrier sense multiple access method with collision detection (CSMA/CD) and physical layer specifications.*

[15] ISO DIS 8802-4, *Information processing systems - Local area networks - Part 4: Token-passing bus access method and physical layer specifications.*

[16] ISO 8802-5, *Information processing systems - Local area networks - Part 5: Token ring access method and physical layer specifications.*

[20] ISO 10039, *Information Technology - Local Area Networks - MAC Service Definition*

2.3 ANSI

[21] ANSI/IEEE 802.2, *Logical Link Control*, 1987.

[22] ANSI/IEEE 802.3, *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Physical Layer Specifications*, 1985.

[23] ANSI/IEEE 802.3 Supplement a, *MAU and Baseband Medium Specification, Type 10BASE2 (Section 10)*, 1988.

[24] ANSI/IEEE 802.3 Supplement b, *Broadband MAU and Broadband Medium Specifications, Type 10BROAD36 (Section 11)*, 1988.

[25] ANSI/IEEE 802.3 Supplement c, *Repeater Unit for 10 Mb/s Baseband Networks (Section 9)*, 1988.

[26] ANSI/IEEE 802.3 Supplement e, *Physical Signaling, Medium Attachment, and Baseband Medium Specifications, Type 1BASE5 (Section 12)*, 1988.

[27] ANSI/IEEE 802.4, *Token-Passing Bus Access Method and Physical Layer Specifications*, Draft 1987.

[28] ANSI/IEEE 802.5, *Token-Ring Access Method and Physical Layer Specifications*, 1986.

[29] ANS T1.408-1990, *ISDN Primary Interface - Customer installation Metallic Interfaces - Layer 1 Specification.*

[30] ANS T1.601, *Integrated Services Digital Network (ISDN) - Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT (Layer 1 Specification).*

[31] ANS T1.602, *Integrated Services Digital Network (ISDN) - Data-Link Layer Signaling Specification for Application at the User-Network interface.*

[32] ANS T1.605, *Integrated Services Digital Network (ISDN) - Basic Access Interface for S and T Reference Points - Layer 1 Specification.*

[33] ANS T1.606 - *Frame Relay Bearer Service - Architectural Framework and Service Description*, 1990.

[34] ANS T1.606 - Addendum 1 - Frame Relaying Bearer Service - Architectural Framework and Service Description. (Final text may be found in T1S1/91-454.)

[35] ANS T1.607, *Telecommunications - Digital Subscriber Signaling System #1 - Layer 3 Signaling Specification for Circuit Switched Bearer Service.*

[36] ANS T1.608, *Telecommunications - Digital Subscriber Signaling System #1 - Layer 3 Signaling Specification for Packet Switched Bearer Service.*

[37] ANS T1.617 - DSS1-*Core Aspects of Frame Protocol for Use with Frame Relay Bearer Service*, 1991.

[38] ANS T1.618 - DSS1-*Signaling Specification for Frame Relay Bearer Service*, 1991.

[39] ANS X3.139, *Fiber distributed data interface (FDDI) - Token ring media access control (MAC)*, 1987.

[40] ANS X3.148, *Fiber distributed data interface (FDDI) - Token ring physical layer protocol (PHY)*, 1988.

[41] ANS X3.166, *Fiber distributed data Interface (FDDI) - Physical layer medium dependent (PMD)*, 1989.

3 Status

This version was completed in December 1990.

4 Errata

NOTE - This clause may contain "defect report" and resolutions material, and the versions of Implementor's Agreements to which this material applies.

5 Local Area Networks

5.1 IEEE 802.2 Logical Link Control

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

a) Link Service Access Point (LSAP):

1) 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;

2) The most significant bit is bit 7, thus this bit pattern represents hexadecimal FE (see figure 1 below);

b) Type and Class:

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

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

Figure 1 - LSAP bit pattern

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:

a) The address length shall be 48 bits;

b) For a data packet of LLC data length of n octets, the length of the pad field is as follows:

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

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

a) Single Cable Networks:

1) The translator frequency shall be 192.25 Mhz;

2) The channel allocations are as follows:

a) Reverse Channels:

1) T12, T13, T14

2) T13, T14, 2'

3) T14, 2', 3'

4) 2', 3', 4'

5) 3', 4', 4A'

6) 4', 4A', 5'

b) Forward Channels:

1) L, M, N

2) M, N, O

3) N, O, P

4) O, P, Q

5) P, Q, R

6) Q, R, S

b) Dual Cable Networks:

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

1) T12, T13, T14

2) T13, T14, 2'

3) T14, 2', 3'

4) 2', 3', 4'

5) 3', 4', 4A'

6) 4', 4A', 5'

7) L, M, N

8) M, N, O

9) N, O, P

10) O, P, Q

11) Q, R, S

c) When both IEEE 802.4 and IEEE 802.3 10 BROAD 36 networks coexist on the broadband cable system the preferred channel allocations are as follows:

1) Reverse:

a) T12, T13, T14 - IEEE 802.3;

b) 6', FM1' - IEEE 802.4;

c) 3', 4'- Channels reserved for future use;

d) 4A', 5' - Channels reserved for future use;

2) Forward:

a) L, M, N - IEEE 802.3;

b) T, U - IEEE 802.4;

c) P, Q - Channels reserved for future use;

d) R, S - Channels reserved for future use.

The following implementation agreements have been reached with respect to 10 BASE-T networks:

a) Auto-partition: A repeater port which connects 10 BASE-T links either through an external or internal MAU shall implement the auto-partition and reconnections algorithm as defined in IEEE 802.3, Chapter 9.

5.3 IEEE 802.4 Token Bus Access Method

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

a) Data Rate:

1) 10 Mb (Broadband);

2) 5 Mb (Carrierband);

b) Addressing: 48 bit;

c) The ImeOption, Priority Mechanism, shall be Implemented;

d) Broadband Channel Assignments are as follows:

1) Forward:

a) P

b) Q

c) R

d) S

e) T

f) U

2) Reverse:

a) 3'

b) 4'

c) 4A'

d) 5'

e) 6'

f) FM1'.

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 Physicai Layer specification:

a) The data signalling rate shali be 4 Mbit/s;

b) The address length shall be 48 bits;

c) The message priority (PM) of the AMP data unit shall be 7;

d) The ALL_STATIONS_THIS_RING_ADDRESS shall be X'C000FFFFFFFF';

e) The TRR value shall be 4 millIseconds;

f) The THT value shall be 8.9 millIseconds;

g) The TQP value shall be 20 milliseconds;

h) The TVX value shall be 10 milliseconds;

i) The TNT value shall be 2.6 milliseconds;

j) The TAM value shall be 7 seconds;

k) The TSM value shall be 15 seconds;

l) The MAC information field (I-field) shall be defined as follows in figure 2 below. Sequences are as follows:

1) Starting Sequence includes: SD, AC, FC, DA, SA;

2) Ending Sequence includes: FCS, ED, FS;

m) With the above timer and MAC i-field definitions, the following limits are defined:

1) Protocol limits the I-field to a maximum of 4425 bytes;

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

Figure 2 - I-Field format

All taken token ring interfaces should support the use of group MAC addressing (in addition to functional addressing) in accordance with ISO 8802-5. It is strongly recommended that group addresses be used to support all OSI uses of multicast on token ring networks.

NOTE - It is expected that support for group MAC addressing for OSI usage shall be mandatory in the future.

In some deployment scenarios, it may be necessary to use functional addresses so as to interoperate with existing installed systems that have limited, or no, ability to support group addresses. Refer to the appropriate base standards and implementors agreements for specific restrictions and recommendations regarding these issues.

5.5 Fiber Distributed Data Interface (FDDI)

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

The following are implementation agreements with respect to FDDI MAC:

a) The address length shall be 48 bits;

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

c) The default value of T_Max shall be at least 165 milliseconds and not more than 200 milliseconds;

d) The default value of T_Req shall be equal to T_MAX_LB.[1] ;

e) All FDDI stations shall process Info_Fields of 0 to 4478 bytes. The frame is defined below in Figure 3;

f) Stations shall not use restricted token service.

Figure 3 - FDDI frame format

P: Preamble
- 16 Idle Symbols for Transmitting
- >= 12 Idle Symbols for Copying
- >=2 Idle Symbols for Repeating

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

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

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

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

- 1 microsecond in a station, and

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

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

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

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

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

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

6 Wide Area Networks

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.)

6.2 ISO 7776

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

a) Address assignment information is as follows:

1) DTE = A (=11000000 binary);

2) DCE = B (=10000000 binary);

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

b) The modulus shall be 8;

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

supported;

d) The Multilink Procedures are excluded.

6.3 ISO 8208

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

a) Virtual Call Service;

b) 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;

c) A DTE must be capable of receiving the Flow Control Parameter Negotiation Facility and responding appropriately (per ISO 8208);

d) 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.

7 Integrated Services Digital Networks (ISDN)

7.1 Introduction

This clause 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 4 and 5 give the agreed stacks for X.25 packet transfer over D and B channels, respectively. Some particular aspects are given below:

a) 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;

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

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

d) 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;

e) Refer to part 3 for the specification of methods for providing OSI Network Services.

7.2 Implementation Agreements

This clause gives implementation Agreements for individual ISDN-related protocols. The relevant protocol stacks are given in figures 4 and 5.

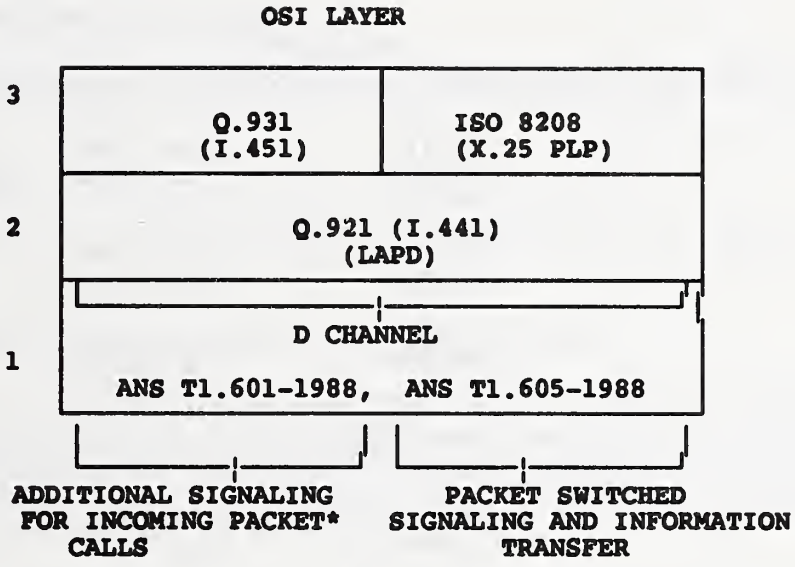


Figure 4 - Protocol Layers at S, T and U reference points when D Channel is used in ISDN

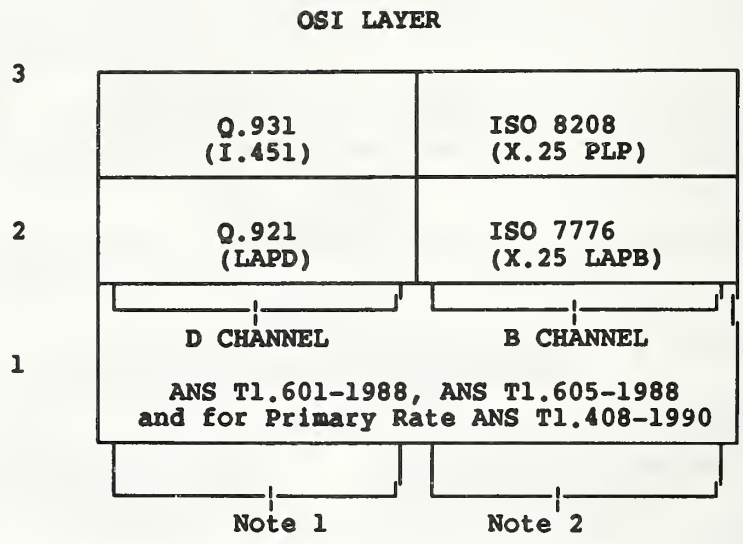


Figure 5 - Protocol Layers at S, T and U reference points when B Channel is used in ISDN

NOTES

1 This refers to signaling for circuit switched access (may be null), as well as additional signaling for incoming packet calls (may be null).

2 This refers to packet switched signaling and information transfer.

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.

7.2.2 Physical Layer, Basic Access at S and T

ANS T1.605-1988, "Integrated Services Digital Network-Basic Access Interface for S and T Reference Points-Layer 1 Specification" applies.

7.2.3 Physical Layer, Primary Rate at S, T, and U

ANS T1.408-1990, "ISDN Primary Rate - Customer Installation Metallic Interfaces - Layer 1 Specification" applies.

7.2.4 Data Link Layer, D-Channel

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

7.2.5 Signaling

CCITT Recommendation Q.931 (I.451), "ISDN User-Network Interface Layer 3 Specification for Basic Call Control" applies.

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

a) 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;

b) 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;

c) 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.

7.2.6 Data Link Layer B-Channel

The agreements on ISO 7776 specified in 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:

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

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

7.2.7 Packet Layer

The agreements on ISO 8208 specified in 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.

8 Frame Relay Subnetworks

8.1 Introduction

The following implementation agreements pertain to Frame Relay subnetworks. These agreements are to guide implementors on optional aspects of the Frame Relay protocol standards. The following terminology is used in this clause:

- must, shall or mandatory - the item is an absolute requirement of this implementation agreement,
- should - the item is highly desirable,
- may or optional - the item is not compulsory, and may be followed or ignored according to the needs of the implementor.

8.2 Relevant Standards

Normative references relevant to Frame Relay are ANS T1.606, and its addendum, ANS T1.607, ANS T1.617, ANS T1.618, CCITT Q.922, and CCITT Q.933.

Additional information relevant to Frame Relay is found in CCITT I.122, and CCITT I.233.1.

8.3 Implementation Agreements

8.3.1 Physical Layer Interface

The recommended physical layer interfaces supported by the Frame Relay network equipment are based on American National Standards and CCITT (International Telegraph and Telephone Consultative Committee) Recommendations. This clause provides a description of the recommended physical layer interfaces that may be supported by a Frame Relay equipment. Interfaces other than those listed below may be used where appropriate (e.g., ISDN, etc.). If the recommended interfaces are used, they should be used as follows:

8.3.1.1 ANS T1.403-1989

The ANS T1.403-1989, "Carrier to Customer Installation DS1 Metallic Interface" document is applicable with the following exceptions:

a) Section 2.2 - Other Publications: The reference to CCITT, Red Book Q921 Recommendation is replaced by "CCITT, Blue Book, Volume VI - Fascicle VI.10, Recommendation Q.921, Digital Subscriber Signalling System No. 1 (DSS 1), Data Link Layer."

b) Section 5.3.1 - Transmission Rate: The rate variation up to +/-200 bit/s is not applicable.

c) Section 6.1 - Framing Format General: The Superframe (SF) format is not applicable.

d) Section 6.3 - Superframe Format: This section is not applicable.

e) Section 7 - Clear Channel Capability: The text in this section is replaced by the following: To provide DS1 Clear Channel Capability (CCC), a DS1 signal with unconstrained information bits is altered to meet the pulse density requirement of 5.6. The method is used to provide DS1 CCC is B8ZS. This method shall be used in both directions of transmission.

f) Section 8 - Maintenance: The mention of SF format and the associated note 4 is not applicable.

g) Section 8.1 - Yellow Alarm: Item 1 of the list (Superframe format) and associated note 5 are not applicable. In the same section; item 3 of the list, is applicable to ESF only.

h) Section 8.3.1.1 - Line Loopback Using the SF Format: This section including note 6 is not applicable.

i) Section 8.4.3.3 - Format of Message-Oriented Performance Report: The sentence before last: "Throughput of the data link may be reduced to less than 4 Kbit/s in some cases" is not applicable.

j) Section 8.4.5 - Special Carrier Applications: Item 3 of the list and note 12 are not applicable.

k) Table 2 - Superframe Format: This table is not applicable.

l) Table 3 - Extended Superframe Format: The portion of the table "Signaling Bit Use Options" and notes related to Option T. Option 2, Option 4, and Option 16 are not applicable.

8.3.1.2 CCITT Recommendation V.35

The interface specifications are:

a) Electrical characteristics according to CCITT Recommendation V.35 Annex 1 and V.28;

b) Connector and pin assignment according to ISO 2983;

c) Interchange circuit defintions according to CCITT Recommendation V.24.

8.3.1.3 CCITT Recommendation G.703 (2048 kbit/s)

Applicable sections of this specification are:

a) Introduction. Except those references which are to 1544 kbit/s;

b) Section 6. Interface at 2048 kbit/s;

c) Annex A. Definition of codes;

d) Annex B. Specification of the overvoltage protection requirement. In addition, when the 75 ohm interface is implemented, the transmit BNC connector shall be labeled TFC OUT and the receive BNC connector shall be labeled TFC IN.

8.3.1.4 CCITT Recommendation G.704 (2048 kbit/s)

Applicable sections of this specification are:

a) General;

b) Section 2.3. Basic frame structure at 2048 kbit/s;

c) Section 5. Characteristics of frame structures carrying channels at various bit rates in 2048 kbit/s interfaces;

d) Annex A.3. CRC-4 procedure for Interface at 2048 kbit/s.

NOTE - Section 1, "General," specifies the electrical interface characteristics to be CCITT Recommendation G.703.

8.3.1.5 CCITT Recommendation X.21 (non-switched operation)

This unstructured interface uses the leased line (i.e., non-switched point to point) subset of the X.21 Recommendation. The interface specifications are:

a) Electrical characteristics according to CCITT Recommendation X.27 (V.11);

b) Connector and pin assignment according to ISO 4903;

c) Interchange circuit definitions according to CCITT Recommendation X.24.

8.3.2 Data Transfer

Implementations shall be based on ANS T1.618. Implementation agreements on the optional parts of ANS T1.618 are proposed as follows:

8.3.2.1 Section 2.2 Flag sequence

Interframe time fill shall be accomplished by transmitting one or more contiguous HDLC flags with the bit pattern '01111110' when the data link layer has no frames to send.

8.3.2.2 Section 2.5 Frame relay information field

A maximum frame relay information field size of 1600 octets shall be supported by the network and the user. In addition, maximum information field sizes less than or greater than 1600 octets may be agreed to between networks and user at subscription time.

8.3.2.3 Section 3.3 Address field variables

See the following:

a) Section 3.3.1 Length of address field;

1) An address field of 2 octets shall be supported. (All frames must have the EA bit set to 0 in the first octet of the address field and the EA bit set to 1 in the second octet of the address field);

2) The 3- and 4-octet address formats are outside the scope of this agreement;

b) Section 3.3.6 Data link connection identifier;

1) The 2 octet address format shall be supported with DLCI values as defined in table 1(a);

c) Section 3.3.6.2 DLCI on the D-channel;

1) This section is not applicable for Permanent Virtual Connections (PVCs);

d) Section 3.3.7 DLCI or DL-CORE control indicator (D/C);

1) This section is not applicable.

Other address structure variables and their usage are as specified in ANS T1.618.

8.3.2.4 Section 5 Congestion control

Congestion control strategy for Frame Relay is defined in ANS T1.606 Addendum 1. The following implementation agreements apply to network equipment and user equipment respectively:

a) Section 5.1 Network response to congestion

1) Mandatory procedures of ANS T1.606 Addendum 1 shall be implemented. When implemented, rate enforcement using the DE indicator, and/or setting of the FECN and BECN indicators, should be implemented according to T1.606 Addendum 1.

b) Section 5.2 User response to congestion

1) User equipment reaction is dependent on the protocols operating over the Data Link Core sublayer. It is recommended that the procedures of ANS T1.618 Annex A should be implemented where appropriate.

8.3.2.5 Section 6 Consolidated link layer management (CLLM) message

Use of the CLLM message is not required.

8.3.3 Control (Signaling) procedures

8.3.3.1 Permanent Virtual Connections (PVC) procedures

a)

a) Managing PVCs on a bearer channel that only supports PVCs: User devices (and the network) shall implement the mandated procedures of Annex D of ANS T1.617. In addition, Annex B and optional procedures of Annex D of ANS T1.617 may also be implemented. By bilateral agreement:

1) Optional procedures of Annex D of ANS T1.617 may be used;

2) Annex B may be used instead of Annex D.

Implementation Note - The number of PVCs that can be supported by Annex D is limited by the maximum frame size that can be supported by the user equipment and the network on the bearer channel (e.g., when the maximum frame relay information field size is 1600 octets then a maximum of 317 PVC status information elements may be encoded in a STATUS message).

b)

b) Managing PVCs on a bearer channel where switched virtual connections (SVCs) and PVCs co-exist: User devices (and the network) shall implement Annex B of ANS T1.617.

8.3.3.2 Switched Virtual Connections (SVCs) procedures

The implementation agreements for SVCs will be provided later.

Annex A (informative)

Cross Reference Between CCITT and ANSI Text Relating to ISDN Agreements

This annex provides a cross-reference listing between those sections of the CCITT ISDN Recommendations given in clause 7 of this document and the sections of the corresponding ANSI ISDN Standards. This appendix does not supersede clause 7 but merely provides a pointer to those who wish to implement the ISDN procedures based on ANSI Standards.

A.1 Data Link Layer, D-Channel

CCITT Recommendation Q.921, which is referenced in 7.2.4 of this document, is identical to ANSI Standard T1.602.

A.2 Signaling

The following table provides the cross-references between those sections of CCITT Recommendation Q.931 referenced in 7.2.5 of this document and the corresponding ANSI ISDN Standards.

Table 1 - ANSI-CCITT cross-references

CCITT RECOMMENDATION Q.931	ANS T1.608
Section 2.1	Section 4.1 (refers to sec. 2.1.1 of ANS T1.607)
Section 2.2	Section 4.2
Section 3.1	Section 5.1 (refers to sec. 3 of ANS 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

Annex B (informative)

Bibliography

B.1 ANSI

ANS T1.607-1989, Telecommunications - Digital Subscriber Signaling System #1 - Layer 3 Signaling Specification for Circuit Switched Bearer Service.

ANS T1.608-1989, Telecommunications - Digital Subscriber Signaling System #1 - Layer 3 Signaling Specification for Packet Switched Bearer Service.

B.2 CCITT

CCITT Recommendation I.122, *Framework for Providing Additional Packet Mode Bearer Services,* 1988.

CCITT Recommendation I.233.1, *ISDN Frame Mode Bearer Services (FRBS)-ISDN Frame Relaying Bearer Service,* (proposed 1991).

CCITT Recommendation I.430 (Blue Book), Basic User-Network Interface - Layer 1 Specification.

B.3 ISO

ISO 9314-1:1989, *Information processing systems - Fibre distributed data interface (FDDI) - Part 1: Token ring physical layer protocol (PHY).*

ISO 9314-3: 1990, *Information processing systems - Fibre distributed data interface (FDDI) - Part 2: Token ring media access control (MAC).*

ISO 9314-3: 1990, *Information processing systems - Fibre distributed data interface (FDDI) - Part 3: Physical layer medium dependent (PMD).*

B.4 OTHER

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

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 3 - Network Layer

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Fred Burg, AT&T**
SIG Editor: **Brenda Gray, NIST**

Foreword

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

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

Annex A is for information only.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 3 - Network Layer

0 Introduction

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

1 Scope

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

2 Normative References

2.1 CCITT

[1] Recommendation X.213 (Blue Book, 1988), *Network Service Definition for Open Systems interconnection for CCITT Applications.*

2.2 ISO

[2] ISO 8348, *information processing systems - Data communications - Network service definition.*

[3] ISO 8348 Addendum 1, *Information processing systems - Data communications - Network service definition - Addendum 1: Connectionless-mode transmission.*

[4] ISO 8348 Addendum 2, *information processing systems - Data communications - Network service definition - Addendum 2: Network layer addressing.*

[5] ISO 8473, *information processing systems - Data communications - Protocol for providing the connectionless-mode network service.*

[6] ISO 8648, *information processing systems - Open systems interconnection - Internal organization of the Network Layer.*

[7] ISO 8878, *Information processing systems - Data communications - Use of X.25 to provide the OSI connection-mode network service.*

[8] ISO 8881, *information processing systems - Data communications - Use of the X.25 packet level protocol in local area networks.*

[9] ISO 9542, *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 service (ISO 8473).*

[10] ISO/IEC 9574, *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).*

[11] ISO/IEC TR 9577, *Information technology - Telecommunications and information exchange between systems - Protocol identification in the network layer.*

[12] ISO/IEC TR 10029, *Information technology - Telecommunications and information exchange between systems - Operation of an X.25 interworking unit.*

[13] ISO/IEC 10030, *Information processing systems - Telecommunications and information exchange between systems - End system routeing information exchange protocol for use in conjunction with ISO 8878.*

[14] ISO/IEC 10589, *Information technology - Telecommunications and information exchange between systems - Intermediate system to intermediate system intro-domain routeing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473).*

[15] ISO/IEC DIS 11577, *Information Technology - Telecommunications and Information Exchange Between Systems - Network Layer Security Protocol*

3 Status

This version of the agreements was completed in December 1990.

4 Errata

This clause may contain "Defect Report" and resolutions material, and the versions of implementor agreements to which this material applies.

The following defects are being progressed in ISO:

a) ISO 9542, defect 1, Parts 1-13;

b) ISO 9542, defect 2;

c) ISO 8473, defects 1 through 11, and technical corrigendum 1;

d) ISO/IEC 10589, defect 1.

5 Connectionless-Mode Network Service (CLNS)

5.1 ISO 8473

NOTE - Defect reports upon the base standard have been issued. See clause 4 of this part for further information.

5.1.1 Subsets of the Protocol

Agreements on subsets of the protocol are as follows:

a) Implementations will not transmit PDUs encoded using the inactive subset. Received PDUs encoded using the inactive subset will be discarded;

b) 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.

5.1.2 Mandatory Functions of ISO 8473

Agreements on Mandatory Functions of ISO 8473 are as follows:

a) 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 ms), whichever is greater;

b) 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;

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

d) 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;

e) For the purposes of relaying and routing, a protocol entity need not verify the correctness of ISO 8348/Add. 2 semantics carried in NPAI of received PDUs.

5.1.3 Optional Functions of ISO 8473

Agreements on Optional Functions of ISO 8473 are as follows:

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

b) Partial and complete source routing will not be supported;[1]

c) Partial record of route will be supported by intermediate systems;

d) ISO 8473 will be followed with respect to QOS;

e) For systems implementing the congestion notification function, the following applies:

1) 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;

2) 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 figure 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;

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

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

a) 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);

b) 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;

5) An embodiment of the averaging algorithm is shown in figure 1;

f) Refer to the Working Implementation Agreements document for additional optional functions.

---

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

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:

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

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

3. Average Queue Length Update:

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

Figure 1 - Queue length averaging algorithm

5.2 Provision of CLNS over Local Area Networks (LANS)

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

a) The definition of CLNS shall be as specified in ISO 8348/Add. 1;

b) The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 5.1;

c) The necessary subnetwork dependent convergence function shall be as defined in ISO 8473 - clause 8.4.2, "SNDCF used with ISO 8802/2 sub-networks."

5.3 Provision of CLNS over X.25 Subnetworks

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

a) The definition of CLNS shall be as specified in ISO 8348/Add. 1;

b) The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 5.1;

c) The necessary subnetwork dependent convergence function shall be as defined in ISO 8473 - clause 8.4.3, "SNDCF used with ISO 8208 subnetworks for operation over X.25 subnetworks," and the default throughput class shall be used if this facility is available;

d) The X.25 PLP shall be as specified in part 2 clause 6.3.

5.4 Provision of CLNS over ISDN

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

a) The definition of CLNS shall be as specified in ISO 8348/Add. 1;

b) The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 5.1;

c) The necessary Subnetwork Dependent Convergence function shall be as defined in:

1) ISO 8473 for operation of CLNP over X.25 with agreements as specified in 5.3;

2) ISO 9574 for control of the B and D channels;

3) The X.25 PLP shall be as specified in part 2, clause 6.3;

4) The agreements for the ISDN-related protocols are specified in part 2, clause 7.

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.

5.5 Provision of CLNS over Point-to-Point Links

Refer to the Working Implementation Agreements document.

6 Connection-Mode Network Service (CONS)

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

6.1 Mandatory Method of Providing CONS

6.1.1 General

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

a) The definition of the CONS is as specified in ISO 8348, Network Service Definition;

b) The mapping of the elements of the CONS to the elements of the X.25 Packet Layer Protocol (PLP) is as specified in 6.3.1;

c) 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.

6.1.2 X.25 WAN

No provisions additional to those in 6.1.1 apply in an X.25 WAN.

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 6.1.1, shall apply:

a) 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.

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 6.1.1, shall apply.

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:

a) The definition of the CONS is as specified in ISO 8348, Network Service Definition;

b) 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 6.3.2.

6.3 Agreements on Protocols

6.3.1 ISO 8878

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

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

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.

6.4 Interworking

Interworking between subnetworks whose End Systems use ISO 8208 to provide the CONS as specified in 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.

7 Addressing

NSAP address formats supported will conform to Addendum 2 of ISO 8348 as follows:

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

b) 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.

c) In routing environments in which systems support NSAP addresses containing selectors as specified in b), the corresponding Network Entity Titles shall have the same format with the NSAP selector set to zero.

d) End Systems and intermediate Systems operating in routing domains that employ the ISO 10589 intradomain Routing Protocol shall meet the NSAP/NET addressing requirements specified in ISO 10589 (clause 7.1) and clause 8.3 of these agreements.

NOTE - This may be incompatible with systems implemented according to previous versions of these agreements.

8 Routing

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

a) The global OSI environment will consist of a number of Administrative Domains, An Administrative Domain consists of a collection of End Systems (ESs) 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;

b) 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;

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

1) End System (ES) to Intermediate System (IS) routing within a single subnetwork;

2) IS to IS routing within a single routing domain (Intra-domain);

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

8.1 ISO 9542 End System to Intermediate System Routing

NOTE - Defect reports upon the base standard have been issued. See clause 4 of this part for further information.

For use in conjunction with ISO 8473, 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:

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

b) For LANs, implementors shall support both Configuration Information and Route Redirection information; no subsets are permitted. For X.25 subnetworks, Route Redirection information shall be supported;

c) All timer values shall be configurable;

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

e) 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;

f) If the configuration notification function described in 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. Alternate mechanisms for ISs and ESs are described in 8.1.1 and 8.1.2.

g) For LANs, this protocol employs the same LSAP as ISO 8473;

h) 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 encoded as specified in clause 12.2.1.4 of ISO 10039. On X.25 subnetworks, it is a DTE address, each digit being binary coded in a semi-octet, and, if there are an odd number of digits, an additional semi-octet set to the value 1111 shall be added at the end;

i) 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:

1) ALL_ESN = 09-00-2B-00-00-04, ALL_ISN = 09-00-2B-00-00-05;

2) 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;

j) when operating on a specific IEEE 802.5 subnetwork all ESs and ISs, shall use exclusively either Funcational Addresses or Group Addresses for the operation of ISO 9542. It is strongly recommended that Group Addressing be used where possible.

For IEEE 802.5 LANs in which Group Addressing is supported by all ESs and ISs, the multicast addresses corresponding to "All Intermediate Systems on the Network" (ALL_ISN) and "All End Systems on the Network" (ALL_ESN) shall be as follows:

1) ALL-ESN = 09-00-2B-00-00-04, ALL_ISN = 09-00 2B-00-00-05;

For IEEE 802.5 LANs in which Functional Addressing must be used, the ISO 9542 multicast shall be as follows:

1) ALL-ESN = 03-00-00-00-02-00, ALL_ISN = 03-00-00-00-01-00.

Editor's Note - When transmitted onto the medium, the above addresses would be represented as follows:

1) ALL_ESN = C0-00-00-00-40-00, ALL_ISN = C0-00-00-00-80-00.

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

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

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

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

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

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

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

a) An End System may choose to ignore an RD PDU received for a destination to which the ES has not sent traffic for some period of time. An ES must record redirection information only for those other systems with which it is in active communication;

b) A holding time value of zero is permitted. When configuration and/or redirection information with a zero holding time is received, prior information shall be replaced, thus causing the system to set its holding timer to zero and discard the corresponding information;

c) If one or more ISs suggested an ESCT, the minimum of the non-zero suggested values replaces the current value of the ES's CT.

8.1.1 Alternative Configuration Mechanism - IS Actions

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.

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:

a) Delay a random interval between 0 and PollESHelloRate seconds. (This is to avoid synchronization with other ISs which have detected a change.);

b) 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;

c) 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;

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

8.1.2 Alternate Configuration Mechanism - 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:

a) SaveCT(currentSuggestedHelloTimer);

b) Transmit an ES Hello with Holding Time field set to MinCT * HoldingMultiplier;

c) 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):

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

b) 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.

8.2 ISO 10030 End System to Intermediate System Routing

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

The following agreements apply to the use of ISO 10030:

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

b) The multicast addresses corresponding to "All CONS End Systems" and "All CONS SNAREs" shall default to the following on IEEE 802.3 and IEEE 802.4 subnetworks:

1) All CONS End Systems = 01-80-C2-00-00-16

2) All CONS SNAREs = 01-80-C2-00-00-17

8.3 Intra-Domain Intermediate Systems to Intermediate Systems Routing

8.3.1 Static Intra-Domain Routing

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

8.3.2 Dynamic Intra-Domain Routing

NOTE - Defect reports upon the base standard have been issued. See clause 4 of this part for further information.

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

The following agreements apply to the use of ISO 10589:

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

b) The implementation shall support a system identifier (ID) length of 6 octets and shall use this value as a default.

8.4 Inter-Domain Intermediate Systems to Intermediate Systems Routing

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

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

9 Procedures for OSI Network Service/Protocol Identification

9.1 General

The Protocol Identifiers specified in ISO TR 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);

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 1 - 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

9.2 Processing of Protocol Identifiers

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

a) the OSI Network Service to be provided;

b) the protocol to be used in providing this service;

c) the role the protocol is to be used in (per the Internal Organization of the Network Layer);

d) the type of subnetwork to which the system is connected.

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 clause 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:

a) 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;

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

c) 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 TR 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.

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 TR 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:

a) 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;

b) 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.

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.

9.3 Applicable Protocol Identifiers

The protocol identifiers applicable to these agreements are given in table 2 and table 3.

Table 2 - 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
1	0	0	0	0	0	1	1	ISO/IEC 10589
1	0	0	0	0	1	1	0	ISO/IEC 11577
x	x	0	1	x	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 - SPI Values

Bit Pattern[1] 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	0	1	1	ISO/IEC 10589
1	0	0	0	0	1	0	0	ISO 8878/Annex A
1	0	0	0	0	1	1	0	ISO/IEC 11577

NOTES

1 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 are outside the scope of these agreements.

10 Migration Considerations

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

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 therefrom 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 6.2.

11 Use of Priority

Refer to the Working Implementation Agreements document.

11.1 Introduction

Refer to the Working Implementation Agreements document.

11.2 Overview

Refer to the Working Implementation Agreements document.

12 Security

~~Refer to the Working Implementation Agreements document.~~

12.1 ISO/IEC DIS 11577 Network Layer Security Protocol (NLSP)

ISO/IEC DIS 11577 describes both a connction oriented and connectionless security protocol that can be used in conjunction with OSI Network Layer Protocols. Before secure communication can be accomplished, a security association (in band or out of band) shall have been established with agreement on all attributes associated with this association.

Managed objects are not yet specified by this standard and therefore the security domain/administrative authority shall determine the procedures and policies that govern this information with other security information.

All mandatory functions are supported by these implementation agreements.

12.2 Services

If access control service is selected and the label mechanism is used then integrity shall also be selected.

12.3 Mechanisms

To optimize efficiency and assist in the interoperability of secure implementations, it is useful to specify which mechanisms and algorithms apply. This specification shall allow implementations to know the exact encapsulation format used including what fields are required, their length, and order. A set of applicable profiles (mechanisms and algorithms) shall be specified within the Implementation Agreements to insure this efficient interoperability.

12.4 Protocol Data Unit

Although the standard has the option of all type-length-value (tlv) fields being in any order, for efficiency, the encapsulation format depicted in the standard shall be used. If the tlv fields are not in order, undefined (type field has not been allocated a value in the NLSP standard), or the PDU fails one of the NLSP Security checks, the secure encapsulated PDU should be discarded. The reporting of this situation is a local matter. If shared knowledge of this event is required, a possible technique would be to use the system management to report the error.

The Security Association-Identification field should be no more than twenty octets.

12.5 Functional Security Sequence Ordering

If Access control is implemented using labels, the label function is first applied followed by the integrity function. If confidentiality has also been selected, then that function is performed after the integrity function.

If integrity and confidentiality have been selected, then the integrity function is performed before the confidentiality function.

13 Conformance

Refer to the Working Implementation Agreements document.

Annex A (informative)

Bibliography

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

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

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

ISO/IEC 8880-2, 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 8880-3, 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 TR 9575, Information Technology - Telecommunications and Information Exchange Between Systems - OSI Routing Framework.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 4 - Transport

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Fred Burg, AT&T**
SIG Editor: **Brenda Gray**

Foreword

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

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

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 4 - Transport

0 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.

1 Scope

This part presents agreements for providing the OSI Transport layer services over both connection mode and connectionless mode services.

2 Normative References

2.1 CCITT

[1] Recommendation X.214 (Blue Book, 1988), *Transport Service Definition for Open Systems Interconnection for CCITT Applications.*

[2] Recommendation X.224 (Blue Book, 1988), *Transport Protocol Specification for Open Systems interconnection for CCITT Applications.*

2.2 ISO

[3] ISO 8072, *information processing systems - Open systems interconnection - Transport service defintion.*

[4] ISO 8072 Addendum 1, *Information processing systems - Open systems interconnection - Addendum 1: Transport service definition - Connectionless-mode transmission.*

[5] ISO 8073 Edition 2, *Information processing systems - Open systems interconnection - Connection oriented transport protocol specification.*

[6] ISO 8073 Addendum 1, *information processing systems - Open systems interconnection - Connection oriented transport protocol specification - Addendum 1: Network connection management subprotocol.*

[7] ISO 8073 Addendum 2, *information processing systems - Open systems interconnection - Connection oriented transport protocol specification - Addendum 2: Class four operation over connectionless network service.*

[8] ISO 8602, *Information processing systems - Open systems interconnection - Protocol for providing*

the connectionless-mode transport service.

[9] *ISO/IEC 10736 Information Technology - Telecommunications and Information Exchange Between Systems -Transport Layer Security Protocol*

3 Status

Completed December 1990.

4 Errata

NOTE - This clause may contain "defect report" and resolutions material, and the versions of implementor agreements to which this material applies.

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).

5.1 Transport Class 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.).

---

[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).

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.

5.1.2.1 General Rules

The rules are as follows:

a) 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;

b) 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;

c) Implementations shall be capable of proposing and accepting the non-use of checksums;

NOTE - See clause 8.2 for more information on checksums when the Transport Protocol and the Transport Layer Security Protocol are both implemented.

d) 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;

e) 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;

f) 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;

g) 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;

h) An unknown parameter in any received CR TPDU shall be ignored;

i) 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;

j) 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;

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

1) Parameter:	2) Action:
a) TSAP id	a) Send DR TPDU
b) TPDU size	b) ignore parameter, use default
c) Version	c) ignore parameter, use default
d) Checksum	d) discard CR TPDU
e) Alternate Protocol Classes	e) Protocol Error

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

m) it is recommended that the capability of request acknowledgments be supported and proposed in CR TPDUs. If request acknowledgments are supported, then if the implementation delays acknowledgments it shall:

1) request use of request acknowledgments in the CR TPDU;

2) accept the use of request acknowledgments in the CC TPDU if it was proposed in the CR TPDU.

n) it is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU.

o) It is recommended that inactivity timer values be exchanged during connection establishment. This may be mandatory in the future. If the "exchange of inactivity timers" capability is supported, the implementation shall send its minimum inactivity timer in the CR TPDU. If a CR TPDU is received with this timer value and the capability is supported, the responding CC TPDU shall contain the inactivity time.

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

$i_R/N > W \qquad N \geq 2$

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.

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. Example techniques for maintaining the estimate and calculating the retransmission timer are described below. Example 1 represents a simple retransmission strategy and example 2 is particularly suitable for networks subject to high traffic loads.

Example 1

The value of the retransmission timer may be calculated according to the following formula:

$$T1 \leftarrow\!- kE + AR.$$

In this formula, E is the current estimate of the round-trip delay on the transport connection, AR is the value of the acknowledgment 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:

$$E \leftarrow\!- E + (1 - \alpha)(S - E).$$

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.

Example 2

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

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

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

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

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

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

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

5.1.2.4 Keep-Alive Function

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:

a) 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;

b) 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 1;

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

1) The window timer must be greater than the round-trip delay. See 5.1.2.3;

2) 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 figure 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.

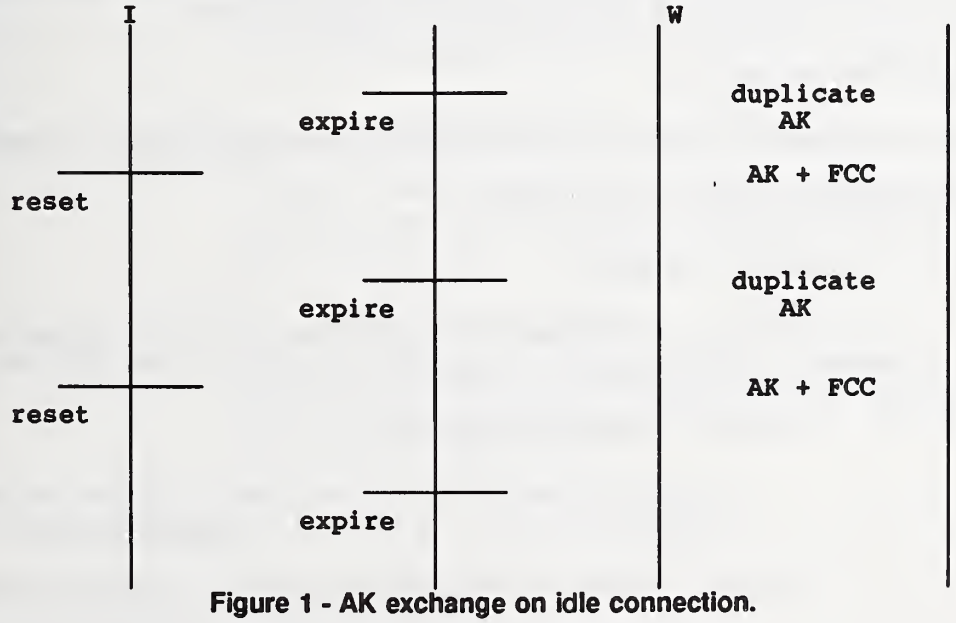


Figure 1 - AK exchange on idle connection.

5.1.2.5 Congestion Avoidance Policies

This clause 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 are as follows:

a) 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;

b) 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;

c) 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;

d) 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 are as follows:

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

1. RECEIVING TRANSPORT ENTITY (RTE) RULES:

a) Rule 1 - Initialization of Window:

1) 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;

a) Rule 2 - Required Sampling Period:

1) 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;

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

1) 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;

c) Rule 4 - Required Action upon the end of a Sampling Period: All RTEs shall take the following actions at the end of each sampling period:

1) 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);

2) Reset N and NC to zero;

3) Transmit the new window WR in the next CDT field sent to the sending transport entity;

2) SENDING TRANSPORT ENTITY (STE) RULES:

a) Rule 1: Initialization of Window:

1) 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;

b) Rule 2: Required Action on a Timeout;

1) 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;

c) Rule 3: Required Counting of Acknowledged TPDU:

1) 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;

d) Rule 4: Increase Window Policy:

1) 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.

5.1.2.6 Use of Priority

(Refer to the Working Implementation Agreements).

5.2 Transport Class 0

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).

5.2.2 Protocol Agreements

5.2.2.1 General Rules

Transport Class 0 agreements are as follows:

a) 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;

b) The allowed values for the maximum TPDU size are 128, 256, 512, 1024, and 2048;

c) The Class 0 protocol does not support multiplexing. At any instant, one Transport corresponds to one Network connection;

d) 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;

e) An unlimited TSDU length must be supported.

f) It is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU.

5.2.2.2 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.

5.2.3 Rules for Negotiation

The rules for class negotiation shall be used.

5.3 Transport Class 2

5.3.1 Transport Class 2 Overview

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

5.3.2 Protocol Agreements

Transport Class 2 agreements follow:

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

1) TS1: 60 seconds;

2) TS2: 60 seconds;

b) 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;

c) The rules for class negotiation shall be used;

d) 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 - 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.

e) It is recommended that implementations send both the preferred and existing TPDU size parameters in the CR TPDU.

6 Provision of Connectionless Transport Service

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

6.1 Connectionless Transport Overview

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

6.2 Protocol Agreements

6.2.1 General Rules

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

a) The optional elements of procedure for use of CLTS over CONS (i.e., clause 6.3 of ISO 8602) will not be supported;

b) A Unitdata TPDU that is received that contains a protocol error or an unknown destination TSAP ID shall be discarded.

6.2.2 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.

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 1. These TPDUs, when used, are conveyed as N-connect user data.

Table 1 - Protocol Identification TPDU Values

TPDU Value	Protocol
03 01 01 00 * (see note 1) 03 01 02 00 ** (see note 2)	ISO 8073/Add. 1 ISO 8602

NOTES

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

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

The following agreements apply:

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

b) When using ISO 8208, usage of a Protocol Identification TPDU not corresponding to those listed in table 1 is outside the scope of these agreements.

8 Security

~~Refer to the Working Implementation Agreements document.~~

8.1 ISO/IEC 10736 Transport Layer Security Protocol (TLSP)

ISO/IEC 10736 describes both a connection oriented and connectionless security protocol that can be used in conjunction with OSI Transport Layer Protocols (ISO/IEC 8073 and ISO/IEC 8602). Before secure communication can be accomplished, a security association (in band or out of band) shall have been established with agreement on all attributes associated with this association.

Managed objects are not yet specified by this standard and therefore the security domain/administrative authority shall determine the procedures and policies that govern this information with other security information.

All mandatory functions are supported by these implementation agreements.

8.2 Services

If access control service is selected and the labels mechanism is used, then integrity shall also be selected.

The Transport (Class 4) initiator shall propose the non-use of checksums if TLSP is also invoked with connection integrity selected (as this would be redundant functionality). The integrity mechanism selected shall be one of the recommended algorithms (a signed MD5 or SHA for public key systems or DES MAC for secret key systems to name just a few) in part 12 (OS Security) of these agreements or a private algorithm that both communicating parties have agreed to use.

8.3 Mechanisms

To optimize efficiency and assist in the interoperability of secure implementations, it is useful to specify which mechanisms and algorithms apply. This specification shall allow implementations to know the exact encapsulation format used including what fields are required, their length, and order. A set of applicable profiles (mechanisms and algorithms) shall be specified within the Implementation Agreements to insure this efficient interoperability.

8.4 Protocol Constraints

Although the standard has the option of all type-length-value (tlv) fields being in any order, for efficiency, the encapsulation format depicted in the standard shall be used. If the tlv fields are not in order, undefined (type field has not been allocated a value in the TLSP Standard), or the SE TPDU fails one of the TLSP Security checks, the secure encapsulated PDU should be discarded. The reporting of this situation is a local matter. If shared knowledge of this event is required, a possible technique would be to use the system management to report the error.

The Security Association-Identification field should be no more than 20 octets.

8.5 Functional Security Sequence Ordering

If Access control is implemented using labels, the label function is first applied followed by the integrity function. If confidentiality has also been selected, then that function is performed after the integrity function.

If integrity and confidentiality have been selected, the integrity function is performed before the confidentiality function.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 5 - Upper Layers

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Jim Quigley, Hewlett Packard**
SIG Editor: **Debbie Britt, NCTS**

Foreword

This part of the Stable Implementation Agreements was prepared by the Upper Layers Special Interest Group (ULSIG) of the Open Systems Environment Implementors' Workshop (OIW). The charter for the OIW is located in the Procedures Manual.

The text in this part has been approved by the Plenary of the OIW. This part replaces the previously existing part on the Upper Layers.

Annex B is for information purposes only. Annex A forms an integral part of these implementor Agreements.

Future changes and additions to these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

List of Tables

Part 5 - Upper Layers

0 Introduction

In this portion of the Implementors' Agreements, the 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.

1 Scope

The agreements in this part 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" clause of this part.

2 Normative References

2.1 Session Layer

[1] ISO 8326: 1987 (E), *Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Service Definition.*

[2] ISO 8327: 1987 (E), *Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification.*

[3] ISO/IEC JTC1/SC21 N2494, *Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Service Definition-AD 2 to ISO 8326 to incorporate Unlimited User Data.*

[4] ISO/IEC JTC1/SC21 N2495, *Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification - AD 2 to ISO 8327 to incorporate Unlimited User Data.*

[5] ISO/AD3 8326, *Information Processing Systems - Open Systems Interconnection-Session Service Definition: Addendum 3 Covering Connectionless-Mode Session Service.*

[6] ISO/IS 9548, *Information Processing Systems - Open Systems Interconnection-Connectionless Session Protocol to Provide the Connectionless-Mode Session Service.*

2.2 Presentation Layer

[7] ISO 8822: 1988 (ISO/IEC JTC1/SC21 N2335), *Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Service Definition.*

[8] ISO 8823: 1988 (ISO/IEC JTC1/SC21 N2336), *Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Protocol Specification.*

[9] ISO 8824: 1990 (E), *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

[10] ISO 8825: 1990 (E), *Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

[11] ISO/DAD1 8822: 1989-02-15(e) (ISO/IEC JTC1/SC21 N 3171), *Information Processing Systems - Open Systems Interconnection - Presentation Service Definition: Draft Addendum 1 Covering Connectionless-Mode Presentation Service.*

[12] ISO/IS 9576: 1989-02-25 5(E) (ISO/IEC JTC1/SC21 N 3172), *Information Processing Systems - Open Systems Interconnection - Connectionless Presentation Protocol to Provide the Connectionless-Mode Presentation Service.*

2.3 Application Layer

[13] ISO/DP 9545, ISO/TC97/SC21/N1743, July 24, 1987, revised November 1987, *Information Processing Systems - Open Systems Interconnection - Application Layer Structure.*

2.4 Application Layer - ASE/ACSE

[14] ISO 8649: 1987 (E) (ISO/IEC JTC1/SC21 N2326), *Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element.*

[15] ISO 8650: 1987 (E) (ISO/IEC JTC1/SC21 N2327), *Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element.*

[16] ISO 8649/DAD2, *Information Processing System - Open Systems Interconnection - ACSE Service Definition: Draft Addendum 2 Covering Connectionless-Mode ACSE Service.*

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

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

[19] ISO 8649/Cor.1: 1991 (E) (ISO/IEC JTC1/SC21 N5630), *Information Processing Systems - Open Systems Interconnection - Technical Corrigendum 1 to ACSE Service* (ISO 8649: 1988) Covering Defects 8649/001, 8649/002 and 8649/003.

[20] ISO 8650/Cor.1: 1991 (E) (ISO/IEC JTC1/SC21 N5631), *Information Processing Systems - Open Systems Interconnection - Technical Corrigendum 1 to ACSE Protocol* (ISO 8650: 1988) Covering Defects 8650/001, 8649/004.

[20] ISO IS 10035: 1989-02-25 (ISO/IEC JTC1/SC21 N 3456), *Information Processing Systems - Open Systems Interconnection - Connectionless ACSE Protocol to Provide the Connectionless-Mode ACSE Service.*

3 Status

This text is stable.

NOTE - Changes due to errata are summarized in clause 4

4 Errata

4.1 ISO Defect Solutions

This clause lists the defect solutions from ISO which are currently recognized to be valid for the purposes of conformance.

ISO 8326 defect solutions:

023, 024

ISO 8327 defect solutions:

037, 038

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.

4.3 Approved Errata

Errata to this part are marked with change bars; deleted text is marked with strikeouts and new text is indicated by shading.

5 Association Control Service Element

5.1 Introduction

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

5.2 Services

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

5.3 Protocol Agreements

5.3.1 Application Context

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

5.3.2 AE Title

AE-titles shall be implemented as specified in ISO 8650/ Corr.1.

5.3.3 Peer Entity Authentication

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

5.4 ASN.1 Encoding Rules

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 Connectionless

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

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

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 part have been made regarding this standard.

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 part have been made regarding this standard.

8 Presentation

8.1 Introduction

This clause 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 clause. This clause 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.

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.

8.3 Protocol Agreements

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.

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.

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.

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 a workshop-conformant system normally results in either a P-CONNECT indication or a P-CONNECT confirmation being issued.

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.

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.

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, clause 6.2.5.3).

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.

8.3.9 RS-PPDU

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

8.4 Presentation ASN.1 Encoding Rules

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.

8.5 General

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.

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.

8.7 Connectionless

The connectionless Presentation protocol shall be implemented as specified in ISO 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 part have been made regarding this standard.

9 Session

9.1 Introduction

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

9.2 Services

The following functional units are within the scope of a workshop-conformant system:

a) Kernel;

b) Duplex;

c) Expedited Data;

d) Resynchronize;

e) Exceptions;

f) Activity Management;

g) Half-duplex;

h) Minor Synchronize;

i) Major Synchronize;

j) Typed Data.

9.3 Protocol Agreements

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.

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.

9.3.3 Reuse of Transport Connection

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

9.3.4 Use of Transport Expedited Data

The Session use of Transport expedited service is optional.

> **NOTE** - A referencing ASE may require that this feature shall be offered by an initiating implementation if it is available, and that it shall be accepted by a responding implementation if it is available and was offered.

9.3.5 Use of Session Version Number

Session Versions 1 and 2 are recognized. Each relevant SIG chooses the version or versions of Session which it requires for a particular implementation phase, and these choices are documented in clause 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. Workshop-conformant Session Version 2 implementations need only support the maximum data lengths specified in the Specific ASE Requirements section.

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.

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.

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.

9.3.8 S-Selectors

S-selectors shall be a maximum of 16 octets.

9.4 Connectionless

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

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

10 UNIVERSAL ASN.1 ENCODING RULES

10.1 TAGS

The maximum value of an ASN.1 basic encoding tag that need be handled by a workshop-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.

10.2 Definite Length

The maximum value of an ASN.1 length octets component that need be handled by an workshop-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.

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.

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.

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

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 Workshop agreement to the contrary is made for a specific INTEGER type.

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.

10.6 Bit String

A responding implementation must be able to accept all valid encodings of the BIT STRING type.

NOTES:

1. Local restraints may restrict the length of BIT STRING which can be decoded. However, the minimum length supported shall be at least equal to the currenly defined number of bits of the referencing specification.

2. For implementation efficiency, it is recommended that initiating applications should encode as many bits as are defined at the time the implementation is released, but no more.

11 Character Sets

See Part 21 of Working Implementation Agreements.

12 Conformance

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

a) A conformant implementation must meet all of the requirements of this specification. All documents referenced in the Upper Layers part 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 clause 2.

b) Workshop-conformant implementations shall be ISO conformant. PICS may contain limitations on length or value aspects of a protocol. PICS of workshop-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.

13 Specific ASE Requirements

The following list for each ASE the corresponding 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 Implementor's agreements.

13.1 FTAM Phase 2

13.1.1 ACSE Requirements

ACSE Functional Requirements: Kernel

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.

13.1.2 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: At least 3 Presentation Contexts must be supported.

Abstract Syntaxes:

a) Abstract Syntaxes for conformant Implementations

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

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

3) "FTAM unstructured binary abstract syntax" { iso(1) standard(0) 8571

abstract-syntax(2) unstructured-binary(4) }

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

b) Abstract Syntaxes Depending on Implementation Profile

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

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

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

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

c) Associated Transfer Syntax:

1) "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.

13.1.3 Session Requirements

Session Functional Units:

a) kernel

b) duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240

13.1.4 Session Options

Session Functional Units:

a) resynchronize - only a Resynchronize Type value of "abandon"

b) minor synchronize

NOTES

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

2 The default value for Minor Sync Point Sync type item PV-field shall be absent if explicit confirmation is required (per ISO 8327, 8.3.20.3) (SIA->value of $).

13.1.5 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 part.

13.2 MHS

13.2.1 Phase 1 (1984 X.400) Session Requirements

Session Functional Units:

a) kernel

b) half-duplex

c) exceptions

d) activity management

e) minor synchronize

Version Number: 1

Maximum size of User Data parameter field: 512

NOTES

1 Restricted use is made by the RTS of the Session services implied by the functional units selected. Specifically, 1) No use is made of S-TOKEN-GIVE, and 2) S-PLEASE-TOKENS only asks for the data token.

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

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

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

13.2.2.1 ROSE Requirements

ROSE is not used.

13.2.2.2 RTSE Requirements

The RTSE requirements are:

a) Monologue

b) TWA - optional

c) checkpointing:

1) minimum checkpointsize = 1

2) minimum windowsize = 1

d) no checkpointing

For the Monologue Association:

a) initiator keeps initial turn

b) APDUs are transferred from initiator to responder only

c) no turn passing

d) only the initiator effects the orderly release of an association

For the two way alternate Association

a) the initiator may keep or pass the initial turn, at binding

b) APDUs are transferred by the holder of the turn

c) only the initiator effects the orderly release of an association, when it possesses the turn

13.2.2.3 ACSE Requirements

As per Phase 2, Protocol P7.

Application Contexts:

a) "MTS-transfer-protocol-1984" - mandatory

b) "MTS-transfer-protocol" - mandatory

c) "MTS-transfer" - mandatory

13.2.2.4 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: at least 3 must be supported

Abstract Syntaxes:

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

b) "MTS-RTSE"

c) "MTSE"

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

13.2.2.5 Session Requirements

As per Phase 2, Protocol P7.

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

13.2.3.1 ROSE Requirements

Operation and association classes are used as per the standard.

13.2.3.2 RTSE Requirements

The RTSE requirements are:

a) TWA

b) normal-mode

c) checkpointing

d) minimum checkpointsize = 1

e) minimum windowsize = 1

f) no checkpointing

For the Monologue Association:

a) initiator keeps initial turn

b) APDUs are transferred from initiator to responder only

c) no turn passing

d) only the initiator effects the orderly release of an association

For two way alternate Association:

a) the initiator may keep or pass the initial turn, at binding

b) APDUs are transferred by the holder of the turn

c) only the initiator effects the orderly release of an association, when it possesses the turn

13.2.3.3 ACSE Requirements

ACSE Functional Requirements: Kernel

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:

a) "MS-access" - mandatory; normal mode

b) "MS-reliable-access" - optional; normal mode

13.2.3.4 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: at least 5

Abstract Syntaxes:

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

b) MSBind/MSUnbind (with or without RTSE)

c) MSSE (Message Submission)

d) MASE (Message Administration)

e) MRSE (Message Retrieval)

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

13.2.3.5 Session Requirements

Session Functional Units:

a) kernel

b) half-duplex (if RTSE is supported)

c) full-duplex (if RTSE is not supported)

d) exceptions

e) activity management

f) minor synchronize

Version Number: 2

Maximum size of User Data parameter field: 10,240

NOTES

1 MHS proposes both versions 1 and 2 for pass through mode (X.410 mode), but only version 2 for normal mode.

2 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.

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

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

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

13.2.4.1 ROSE Requirements

As per Phase 2, P7.

13.2.4.2 RTSE Requirements

As per Phase 2, P7.

13.2.4.3 ACSE Requirements

As per Phase 2, P7.

Application Contexts:

a) "MTS-access" - mandatory

b) "MTS-reliable-access" - optional

c) "MTS-forced-access" - mandatory

d) "MTS-forced-reliable-access" - optional

13.2.4.4 Presentation Requirements

As per Phase 2, P7.

13.2.4.5 Session Requirements

As per Phase 2, P7.

13.3 DS Phase 1

13.3.1 ACSE Requirements

ACSE Functional Requirements: Kernel

Application Contexts:

a) "id-ac-directoryAccessAC" { joint-iso-ccitt(2) ds(5) 3 1 }

b) "id-ac-directorySystemAC" { joint-iso-ccitt(2) ds(5) 3 2 }

13.3.2 Presentation Requirements

Presentation Functional Units: kernel

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

Abstract Syntaxes:

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

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

c) "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) }

13.3.3 Session Requirements

Session Functional Units:

a) kernel

b) duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240

13.4 Virtual Terminal

13.4.1 Phase Ia

13.4.1.1 ACSE Requirements

ACSE Functional Requirements: Kernel

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

13.4.1.2 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: at least 2 must be supported

Abstract Syntaxes:

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

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

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

13.4.1.3 Session Requirements

Session Functional Units:

a) kernel

b) duplex

c) expedited data

d) major synchronize

e) resynchronize - only a Resynchronize Type value of "restart"

f) typed data

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options: expedited data

13.4.2 Phase Ib and Phase II

13.4.2.1 ACSE Requirements

ACSE Functional Requirements: Kernel

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

13.4.2.2 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: at least 2 must be supported

Abstract Syntaxes:

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

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

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

13.4.2.3 Session Requirements

Session Functional Units:

a) kernel

b) duplex

c) half-duplex

d) expedited data

e) major synchronize

f) resynchronize - only a Resynchronize Type value of "restart"

g) typed data

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options: expedited data

13.5 MMS

13.5.1 ACSE Requirements

ACSE Functional Units: Kernel

Application Context: "ISO MMS" { iso(1) standard(0) 9506 part(2) mms-application-context-version1(3)} - implies use of ACSE and MMS ASE

13.5.2 Presentation Requirements

Presentation Functional Units: Kernel

At ieast 2 Presentation Contexts must be supported.

Abstract Syntaxes:

a) "mms-abstract-syntax-major-version1" { iso(1) standard(0) 9506 part(2) mms-abstract-syntax-major-version1 (1)}

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

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

13.5.3 Session Requirements

Session Functional Units:

a) Kernel

b) Duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240

13.6 Transaction Processing

See Working Implementation Agreements Document.

13.7 Network Management

13.7.1 ROSE Requirements

The Rose requirements are as specified in ISO 9596 section 5.2: Underlying Services, and section 6.2 Remote Operations.

Operations Classes: 1, 2, and 5

Association Classes: 3

13.7.2 ACSE Requirements

ACSE Functional Units: kernel

Application Contexts: as defined by [SMO]

AE-Title: The association responder shall support both forms of the AE-Title. The association requestor may use either form of the AE-Title.

13.7.3 Presentation Requirements

Presentation Functional Units: kernel

Presentation Contexts: At least 2 must be supported.

Abstract Syntaxes:

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

b) "CMIP-PCI" {joint-iso-ccitt(2) ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}

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

13.7.4 Session Requirements

Session Functional Units:

a) kernel

b) duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240.

13.8 Remote Database Access

13.8.1 ACSE Requirements

ACSE Functional Units: Kernel

Application Contexts:

a) "RDA-SQL-BASIC-APPL-CONTEXT-V1" {iso(1) standard(0) rda(9579) part-2(2) basic-ac(2) version-1(1)} implies use of the ACSE and RDA SQL ASEs;

b) "RDA-SQL-TP-APPL-CONTEXT-V1" {iso(1) standard(0) rda(9579) part-2(2) tp-ac(3) version-1(1)} implies use of the ACSE, RDA SQL, TP, and optionally CCR ASEs.

13.8.2 Presentation Requirements

Presentation Functional Units: Kernel

13.8.2.1 Presentation Contexts for the RDA Basic Application Context

At least 2 presentation contexts must be supported;

Abstract Syntaxes:

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

b) "RDA-SQL-ABSTRACT-SYNTAX-V1" {iso(1) standard(0) rda(9579) part-2(2) abstract-syntax(1) version-1(1)};

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

13.8.3 Session Requirements

Session Functional Units:

a) Kernel;

b) Duplex;

Version: 2:

Maximum size of User Data parameter field: 10,240.

Annex A (normative)

Object Identifier Register

Editor's Note - Annexes A and B have been switched to place the informative annex after the normative annex.

A.1 Register Index

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

a) { iso(1) identified-organization(3) oiw(14) uisig(8) application-context(1) nil(1) } is defined in 14.2;

b) { iso(1) identified-organization(3) oiw(14) uisig(8) abstract-syntax(2) octet-string(1) } is defined in 14.2.

A.2 Object Identifier Descriptions

{ iso(1) identified-organization(3) oiw(14) uisig(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.

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

```
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.

Annex B (informative)

Recommended Practices

Editor's Note - Annexes A and B have been switched to place the informative annex after the normative annex.

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 1.

The second octet encodes the incoming event as a number relative to 0 as detailed in table 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 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 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 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 Paramet r
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
OA7	80	VERFLOW ACCEPT SPDU
PR-AB	79	PREPARE (ABORT) SPDU

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 6 - Registration Authority Procedures for the OSE Implementors Workshop (OIW)

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Einar Stefferud, Network Management Associates, Inc.**
SIG Editor: **Brenda Gray, NIST**

Foreword

This part of the Stable Implementation Agreements was prepared by the Registration Special Interest Group (RSIG) of the Open Systems Environment Implementors' Workshop (OIW).

This part replaces the previously existing chapter on this subject.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 6 - Registration Authority Procedures for the OSE Implementors' Workshop (OIW)

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 OSE 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.

0 Introduction

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.

1 Scope

This part defines registration procedures for OSE 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 part, 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 part is structured as follows: 6.2 describes the information objects that need to be registered, and 6.3 describes a registration procedures for OIW object identifiers. Annex A lists the object identifier component values assigned to the OIW and the SIGs. Annex B discusses object identifiers used in the 1987 and 1988 Stable Implementation Agreements. Annex C presents guidelines for registering changes to technical objects. The appendices are integral parts of this specification.

2 Normative References

3 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:

a) Application-process-titles;

b) Application-entity-titles;

c) Abstract syntaxes;

d) Transfer syntaxes;

e) Application-contexts;

f) MHS;

1) ADMD names;

2) PRMD names;

3) Organization names;

4) Encoded information types;

5) Extended body part types;

6) Extensions;

7) etc.;

g) Object Identifier values;

h) ASN.1 modules;

i) Directory;

1) Relative distinguished names;

2) Attribute types;

3) Attribute syntaxes;

4) Object classes;

5) Encryption algorithms;

6) etc.;

j) VT;

1) Profiles;

2) Reference Information objects;

3) etc.;

k) Network management objects;

l) Network layer addresses;

m) System titles;

n) FTAM;

1) Document types;

2) Constraint sets;

3) etc.;

o) 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 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 1 - Structure of Object Identifier for OIW.

As an example figure 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)

                                          rasig(13)

                                                   example(0)
```

Figure 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 Annex A. The assignment of values below each SIG in the object identifier tree is the responsibility of that SIG.

4 Registration Procedures for Object Identifiers

This clause 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.

4.1 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.

4.2 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.

4.3 Requirements for Information Object Registration

4.3.1 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)}

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

4.3.2 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.

4.3.3 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 to the "Stable Implementation Agreements" document.

4.3.4 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.

4.4 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 part annexes of the OIW Agreements Document.

Table 1 - Index Entry Example

Object Identifier	Reference
iso identified-organization	4.3.1
oiw(14) rasig(13) example(0)	

Annex A (normative)

Assignments to Workshop Organizations

Table 2 - Identifier Assignments

Identifier	Value	Assigned To	Assigned By
oiw	14	OIW	ISO 6523 RA
llsig	1	SIG	OIW
nmsig	2	SIG	OIW
secsig	3	SIG	OIW
tpsig	4	SIG	OIW
ftamsig	5	SIG	OIW
mhsig	6	SIG	OIW
dssig	7	SIG	OIW
ulsig	8	SIG	OIW
rdasig	9	SIG	OIW
mmssig	10	SIG	OIW
odasig	11	SIG	OIW
vtsig	12	SIG	OIW
rasig	13	SIG	OIW
hcsig	14	SIG	OIW
ctsig	15	SIG	OIW

Annex B (normative)

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.

Annex C (informative)

Guidelines for Registering Changes to Technical Objects

Part 6 of the OIW Agreements document describes the process for registering technical information objects that are defined in the development of OIW implementation agreements. However, the process does not describe a criteria for determining when a change to an object definition is of sufficient magnitude to require registration of a new object with a new OID. Such criteria would be useful when changes are proposed to technical object definitions that have already been registered.

The registration procedures for technical information objects in OIW Implementation Agreements assumes that each object is uniquely different in particular ways from all other registered technical information objects, and requires that there is exactly one definition for each registered object identifier (OID). Therefore, when an object definition is to be changed, it must receive a new OID if the change is "sufficiently significant," in order to signal to all concerned parties that something significant has been changed.

The purpose of this tutorial section is to provide a guide for the evaluation of proposed changes to the definition of a technical object. Many of the changes will fall in a gray area between an obvious "editorial change" with no change to the operational characteristics of the registration and "significant change" that will require the requested change to be registered as a new technical object with a new Object Identifier (OID).

These guidelines are presented to assist in providing a consistent approach for reviewing requests and making changes to any registered technical object.

C.1 Evaluating Registered Technical Objects

Technical objects in the OIW registers include a functional definition describing the object, its states and the actions that can change the object's states. The definition and actions of the object are usually presented as descriptive text, while the state may be defined by a data structure and a set of values such as constants or ranges, having a particular syntax. It should be recognized as stated above that modifying or deleting one or more parts of the definition may not be of sufficient importance to require the registration of the registered technical object under another identifier (OID).

For example, we should note that sometimes a change may be desired to specifically reduce confusion that stems from different interpretations of a given definition. In this case, the change might require some implementations to be modified to conform to a chosen interpretation, but who is to say that the definition was changed, versus saying that the original intent was finally made clear? It is a matter of judgement by the responsible OIW SIG to decide whether a new OID should be assigned in this case, or not.

When a change is sufficiently significant to require a new OID, then the old object must remain unchanged with its old OID.

Review criteria must consider the relative importance of the parts of the technical object definition that are affected by a change before determining whether to approve the proposed change. Deciding not to accept a proposed change may result in a need to create a new technical object very similar to the one being reviewed.

C.2 A Registration Review Criteria

The significant components of the technical definition, to which a criteria can be applied include the text description of the technical object, the definitions of the state values, and the definitions of the data structures. The criteria is not intended to be regulatory in nature, but to provide some direction in reviewing each of the three components when evaluating change requests for registered technical objects.

C.2.1 The Technical Object Description

Does the proposed definition change or require a uniquely different set of functions or state conditions, or does the proposed change alter the relationship between functions of the registered object?

If the proposed definition change adds another function or creates another state, or modifies the relationship between functions or state conditions of the object, or deletes a defined function or state condition then the proposed change should be registered as a new technical object with a new OID, provided that the proposed change would have a significant impact upon the implementation or operation of the technical object when changed.

Editorial changes can be made to correct grammatical errors or to improve clarity, without changing the definition. For changes that require additions or deletions of text to the definition, an evaluation must be made to determine if the changes when applied are optional or will apply only to a local implementation, or are extensive enough to require a new implementation.

Deciding what change means with respect to the functional definition is a subjective view and will require that each SIG establish some guidelines for its particular object types. Consistency of application of a registration policy within each SIG would be most helpful to the process.

One approach is to rule that any change, other than a spelling or grammar change requires a new technical object registration. However, the consequence of such a rule would be a large number of registered technical objects with very similar definitions that can create considerable confusion for implementors. The opposite position is to treat any change to the functional description as an editorial change, and only changes to the other criteria like the state values or data structures are evaluated to make a decision about registration of the changed object as a new technical object.

Between the two is a more acceptable view that provides for the evaluation of the proposed change to decide whether the change is editorial only, NOT affecting implementation; or if it is a change in functionality that MAY affect implementation. If it does NOT affect implementation, then it is an editorial change. If it MAY affect the implementation, then the change requested should be evaluated for registration as a new technical information object with a new OID.
An Example:

Change #1. A given registered object includes a range of values for a particular attribute called TIMEOUT. It includes the following two facts:

a) a definition of the TIMEOUT attribute;

b) the range of values for the TIMEOUT attribute.

If the range of the TIMEOUT attribute is changed from 10..100 to be 10..1000, it is possible that the change is not significant enough to warrant a new registration, if the parameter is only applied locally. (We will assume that this is the case for this example.)

Change#2. Suppose the same attribute is to be deleted. Then some assessment is needed, regarding the impact of the change to the global operational environment in which the technical object is to function, to determine if a new registration is required with a new OID.

The relative significance of the two changes to the operational requirements are clear (given our assumptions here) in these two cases. Changing the values of the range of the TIMEOUT parameter is a relatively minor change which affects only local operation. Depending on other operational considerations, and the relation of the TIMEOUT to other facts about the technical object it could be changed without a new registration. But the elimination of the TIMEOUT attribute altogether would be much more significant, and more than likely require a new registration, since current implementations would be expecting the existence of such an attribute in any operating environment, and future implementations would not include it.

C.2.2 Evaluating the State Values

Within the registered technical object description there may be a number of constants, ranges of values, and syntaxes specifically defined for the object. They are all subject to change. The evaluation criteria applied to requests for changes to state values has to consider the kind of operation that the technical object is performing.

EXAMPLE: The range of accepted values is changed from 1-128 to 0-127, or the default value of a parameter is changed from 32 to 128.

Understanding the implications of what is changing helps to measure the impact of the proposed changes. The shift of the range from 1-128, to 0-127 could be trivial, depending on the scope of its use and would not alone necessarily warrant a new registration. However to change a default value from 32 to 128 (if the attribute applies to the availability or limit of some external system or network resource) would clearly be cause for much concern over how the change impacts implementations of the technical object.

C.2.3 Evaluating the Data Structure

Each technical information object may have one or more data structures defined within the description of the object. Changes can be made to the data structures in a number of ways. Data field sizes can change, and the number of data fields can change. As with the state values, they should be considered in a very broad sense that is within the definition and the extent of the use of these data structures beyond local system usage.

One must be aware that ali syntacticai changes in a technical definition need not be mandatory; they may be optlonal. Given that the changes are mandatory, they are most iikely to affect every implementation, and are going to impact the functioning of the object. Such a case wouid warrant that the change be registered as a new technical object.

EXAMPLE: A defined data fleld Is changed from 3 octets to 4 and another field Is reduced from 2 octets to 1 octet.

Changing the data structure Is probably the ciearest case of a requirement to change an implementation. One must be aware that aii syntacticai changes in technical definitions need not be mandatory, they may be optlonai. But if the changes are mandatory, they will most iikely affect every Implementatlon, and In such a way that they will not interoperate properly with old implementations. Such cases warrant that the change be registered as a new technical object with a new OID.

With respect to applying these criteria, it shouid be emphasized that it is most important to be consistent in making subjective judgments concerning changes to registered technicai objects, rather than being "correct."

There may be more than one interpretation or "correct" view regarding any proposed change, but if the application of the guideiines are consistent, then the Implementations are more iikeiy to be consistent.

C.3 The Change Process

Responsibiiity for evaiuating the change requests is assigned to each SIG. The SiG makes its determinations by voting on changes to each registered object as it is defined in the SiG text in the OiW impiementation Agreements Documents. Any SIG approved changes must also be voted in the OIW Pienary using the ruies of the SIG and the Plenary.

An object definition in a Working Agreement text is not registered untii it has been voted into the Stabie Agreements Document, so it Is possible to modify an as yet "unregistered" object in the Working Agreements Document.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 7 - 1984 Message Handling Systems

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Neil Koorland (Microsoft)**
SIG Editor: **Rich Ankney (Fischer International)**

Foreword

This part of the Stable Implementation Agreements was prepared by the Message Handling Systems Special Interest Group (X.400 SIG) of the Open Systems Environment Implementors' Workshop (OIW).

Text in this part has been approved by the Plenary of the X.400 SIG. This part replaces the previously existing chapter on this subject. There is no significant technical change from this text as previously given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 7 CCITT 1984 X.400 Based Message Handling System

NOTE - The classification schema used in this chapter (see table 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.

0 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 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 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 clause 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 clause 5.9

Figure 1 - The layered structure of this implementation agreement.

1 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 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 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.**

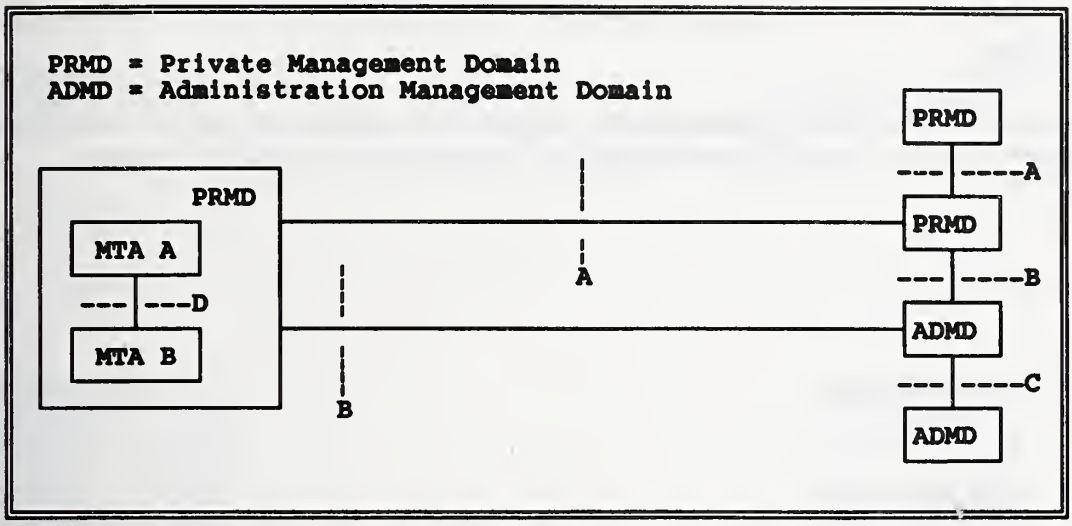


Figure 2 - This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD; and (D) MTA and MTA.

2 Normative references

CCITT Recommendation X.121 (1988), International Numbering Plan.

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), MessageHandling Systems: Interpersonal Messaging User Agent Layer.

CCITT Recommendation X.430, (Red Book, 1984), Message Handling Systems: Access Protocol for Teletex Terminals.

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 clause--Errata.

4 Errata

5 PRMD to PRMD

5.1 Introduction

This clause is limited in scope to issues arising from the **direct** connection (interface A in figure 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 3 schematically represents the scope of this clause.

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 clause addresses the P1 and P2 protocols and their related services in a direct connection environment. This clause 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 clause.

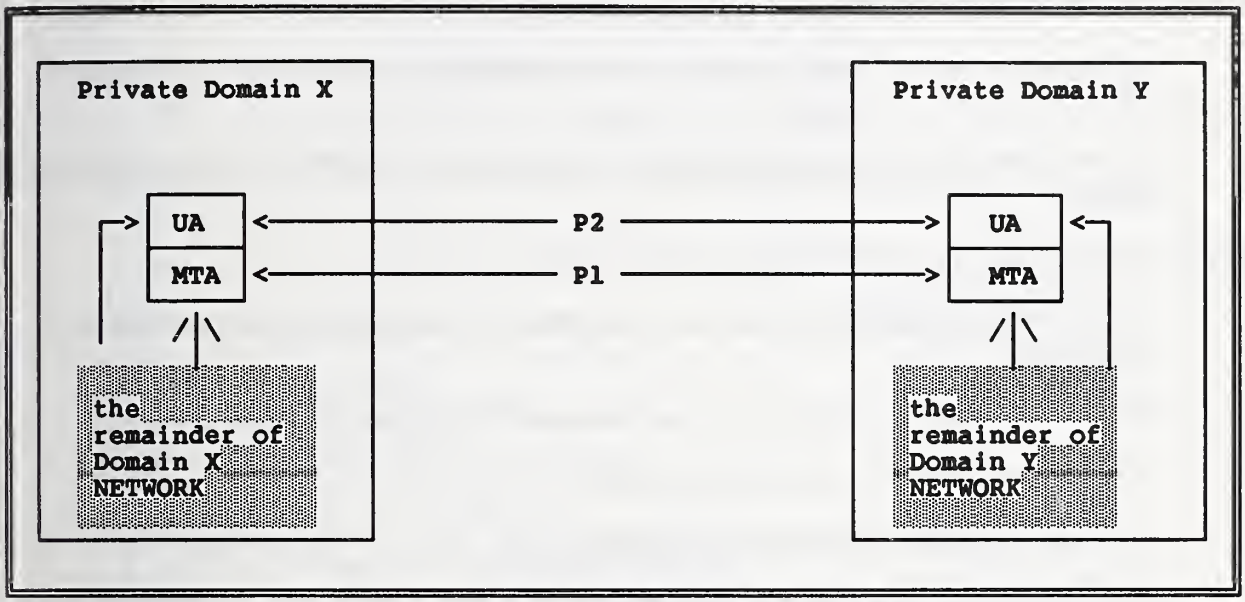


Figure 3 - Interconnection of private domains.

5.2 Service elements and optional user facilities

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

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.

5.2.1.1 Support (S)

Support means:

a) The service provider makes the service element available to the service user, and

b) The service user gives adequate support to the MHS to invoke the service element or makes information associated with the service element available.

Support for Origination means that:

a) The service provider makes the service element available to the service user for invocation, and

b) The service user gives adequate support to the end user of the MHS to invoke the service element.

Support for Reception means that:

a) 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:

b) the service element is available to the originator,

c) the service element is available to the recipient, and

d) all intermediate steps carry the information.

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.

5.2.1.3 Not Used (N/U)

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

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).

5.2.2 Summary of supported services

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 annex A.

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 annex A.

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

5.2.3 MT service elements and optional user facilities

Tables 1 through 3 show the Message Transfer (MT) service elements and optional user facilities.

Table 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

Notes
1 Not applicable to co-resident UA and MTA.

Table 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

Legend
E: Essential optional user facility.
A: Additional optional user facility.

Notes
2 A local facility subject to qualifiers in annex A.
3 Support not required for an originating MT user; support must be provided for recipient MT users.
4 Subject to qualifiers in annex A.

Table 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

Legend
A: Additional optional user facility.

5.2.4 IPM service elements and optional user facilities

Tables 4 through 6 show the IPM service elements and optional user facilities.

Table 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

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

Table 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 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)

Notes
6 A local facility subject to qualifiers in annex A.

5.3 X.400 protocol definitions

This clause reflects the agreements of the OIW regarding P1 and P2 protocol elements.

5.3.1 Protocol classification

The protocol classifications are defined in table 7.

Table 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.

5.3.2 General statements on pragmatic constraints

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.

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.

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.

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.

5.3.3 MPDU size

The following agreements govern the size of MPDUs:

a) All MTAEs must support at least one MPDU of at least 2 megabyte, and

b) The size of the largest MPDU supported by a UAE is a local matter.

5.3.4 P1 protocol elements

Table 8 contains Protocol Elements and their classes.

Table 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	

Table 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 clause 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	

Table 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	

Table 8 - P1 protocol elements (concluded)

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		

Note: [Comment on intermediate Traceinformation in the DeliveryReportContent in table 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.]

5.3.5 ORName protocol elements

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

Table 9 contains ORName protocol elements.

These agreements interpret 1984 X.400 clause 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 9 - ORName protocol elements

Element	Class	Restrictions and Comments
ORName		
StandardAttributeList	M	
DomainDefinedAttributeList	G	
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.

Table 9 - ORName protocol elements (concluded)

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, annex B.

5.3.6 P2 protocol profile (based on [X.420])

Tables 10 and 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.

5.3.6.1 P2 protocol - Heading

Table 10 specifies the support for protocol elements in P2 Headings.

Table 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	

Table 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 annex B.
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.

Table 10 - P2 Heading protocol elements (concluded)

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		

5.3.6.2 P2 protocol - BodyParts

5.3.6.2.1 BodyPart identifiers

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 4.

a) Implementations are required to generate and image IA5Text.

b) Implementations should specify the other BodyPart types supported.

c) 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.

d) For the BodyPart types currently considered, support for the protocol elements is as indicated in table 11.

5.3.6.2.2 Privately defined BodyParts

This clause describes an interim means for identifying privately defined BodyParts. This clause 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
```

Notes

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.

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.

3 Refer to clause 12.6 for recommendations regarding the implementaion of USABodyParts.

Figure 4 - X.409 definition of privately defined BodyParts.

5.3.6.3 P2 BodyPart protocol elements

Table 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 annex C.
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.

Table 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	

Table 11 - P2 BodyParts (concluded)

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.

5.4 Reliable Transfer Server (RTS)

5.4.1 Implementation strategy

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

5.4.2 RTS option selection

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.

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

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

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

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:

a) Dialogue mode: Monologue must be supported for this profile; two-way alternate is used only if both partners agree.

b) Initial turn: Kept by the initiator of the association.

The "priority-mechanism" and the "transfer-time limit" are regarded as local matters.

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*:

1) Checkpointing and No Checkpointing should be supported. Whenever possible, checkpointing should be used.

2) The minimum checkpointSize is 1 (that is, 1024 octets).

d) *Window size*:

1) Minimal value of 1 (if checkpointing is supported).

2) 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:

1) Between two adjacent minor synch points.

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

3) Between S-ACTIVITY-START and S-ACTIVITY-END without checkpoints.

g) A *monologue association* is defined as follows:

1) The RTS user responsible for establishing the association is called the initiator.

2) The initiator keeps the initial turn.

3) APDUs are transferred in the direction of the initiator to the recipient only.

4) There shall be no token passing.

5) 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:

1) The current session connection is of type "recover."

2) 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.

3) 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 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 clause 3.5:

1) 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."

2) 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 clause.

Table 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.

5.4.4 RTS protocol limitations

The RTS Protocol Limitations for this profile are listed in table 13.

Table 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	

Table 13 - RTS protocol elements (concluded)

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.

5.5 Use of session services

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

5.6 Data transfer syntax

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

6 PRMD to ADMD and ADMD to ADMD

6.1 Introduction

This clause 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 clause 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 5 illustrates the two roles played by an ADMD.

This clause is presented in the form of deviations from the agreements applicable to PRMD-to-PRMD (sec. 5). Unless explicitly noted in the remainder of this clause, all of the specifications for PRMD to PRMD apply to PRMD to ADMD and ADMD to ADMD.

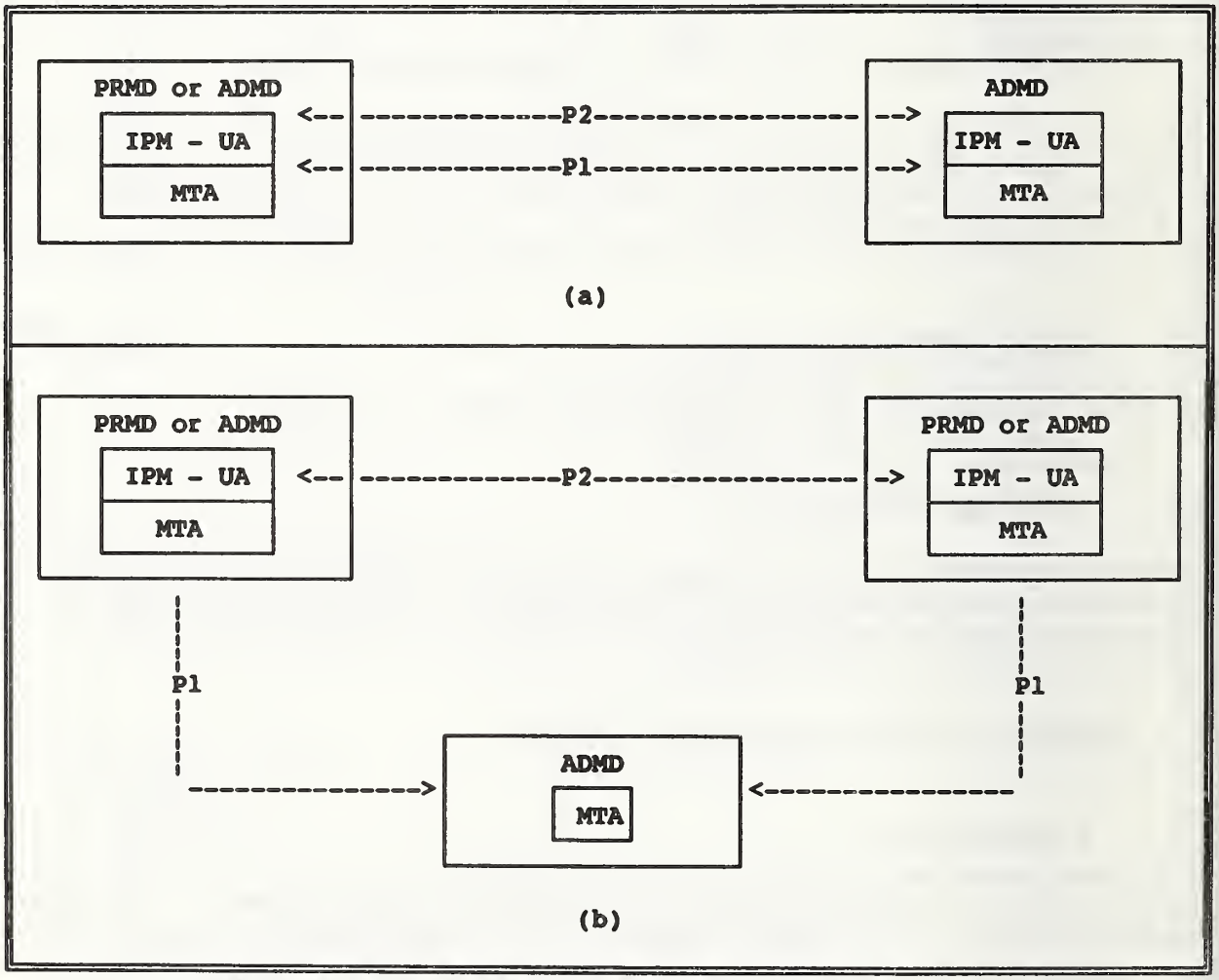


Figure 5 - An ADMD May (b) or May Not (a) Serve as a Relay.

6.2 Additional ADMD functionality

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

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.

6.2.2 P1 protocol classification changes

Table 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 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."		

Notes
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.

6.2.3 O/R Names

O/R Names shall consist of:

a) CountryName,

b) AdministrationDomainName.

as well as one or more of the following:

a) PrivateDomainName,

b) PersonaiName,

c) OrganizationName,

d) OrganizationalUnit,

e) UniqueUAidentifier,

f) X121Address.

g) DomainDefinedAttributeList. (An implementation may accept or reject an OR Name that only contains country, ADMD, and DDA list.)

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.

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:

a) to which the message is being sent (in recipient names)

b) from which the message originated (in the originator name).

6.2.5 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:

a) OrganizationName,

b) OrganizationaiUnit,

c) PersonaiName.

d) DomainDefinedAttributeList. (An implementation may accept or reject an OR Name that only contains country, ADMD, and DDA list.)

6.3 Differences with other profiles

6.3.1 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 clause 5.3.3. Other differences are shown in annex E. TTC is currently based on Version 4 of the Implementor's Guide.

6.3.2 CEPT profile

See annex E.

6.4 Connection of PRMDs to multiple ADMDs

Given that Management Domain names (both PRMD and ADMD) shall be unique within the United States, 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:

a) Naming,

b) Routing,

c) 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.

6.5 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.

6.6 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.

6.7 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:

1) originator O/R Name,

2) recipient O/R Name in the RecipientInfo,

3) Priority.

d) Only recipient Names with the responsibility flag set should be validated. The validation of O/R names is defined in 8.3.3; the validation of priority is defined in 8.3.7.1.

e) MPDU Identifier Validation

1) 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).

2) 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.

3) 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.

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

6.8 Quality of service

6.8.1 Domain availability

6.8.1.1 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.

6.8.1.2 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.

6.8.2 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 15 the delivery time targets are established.

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

16. This implies that, after these times, a Non-Delivery Notice will be generated.

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

Table 15 - Delivery time targets

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

Table 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.

6.9 Billing information

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.

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

6.10 Transparency

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:

a) Relay the message. If the MTA does relay, it must not drop any of the protocol elements.

b) Non-Deliver the message.

A receiving MTA can optionally:

a) Deliver the message

b) Non-Deliver the message.

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.

P2 extensions shall be relayed transparently by ADMDs.

6.11 RTS password management

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

a) password length

b) frequency of changes

c) exchange of passwords with communicating partners

d) loading passwords (i.e., the timing of password changes with respect to active associations).

6.12 For further study

Issues requiring further study are:

a) Intra-Domain Routing

b) Multi-Vendor Domains

7 Inter and intra PRMD connections

7.1 Introduction

This clause 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 figures 6 and 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 clause. 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 clause describes the minimum that must be provided in order to implement a relaying PRMD and a MTA within a PRMD.

This clause is presented in the form of deviations from agreements applicable to PRMD to PRMD connection (sec. 5). That is, unless specifically noted in the remainder of this clause, the agreements in clause 5 apply to both relaying PRMDs and MTAs within a PRMD.

It should be noted that the comments regarding ORNames in clause 6.5 also apply to this clause.

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 clause.

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. 6.8 and 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.

7.2.2 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 figure 8).

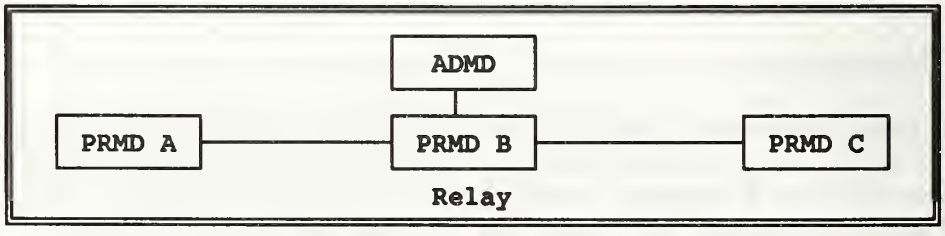


Figure 6 - Relaying PRMD.

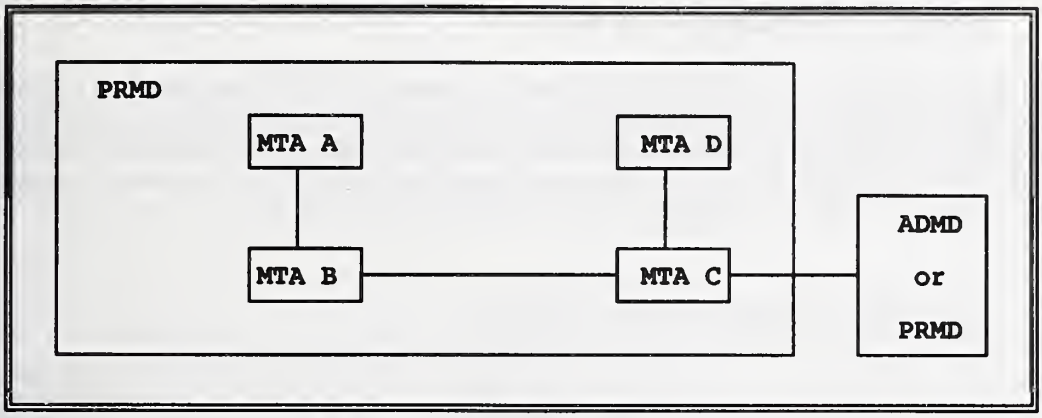


Figure 7 - Intra PRMD connections.

NOTES

1 Clause 6.6 specifies that ADMDs are not required to connect to a relaying PRMD, but they are not precluded from doing so.

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

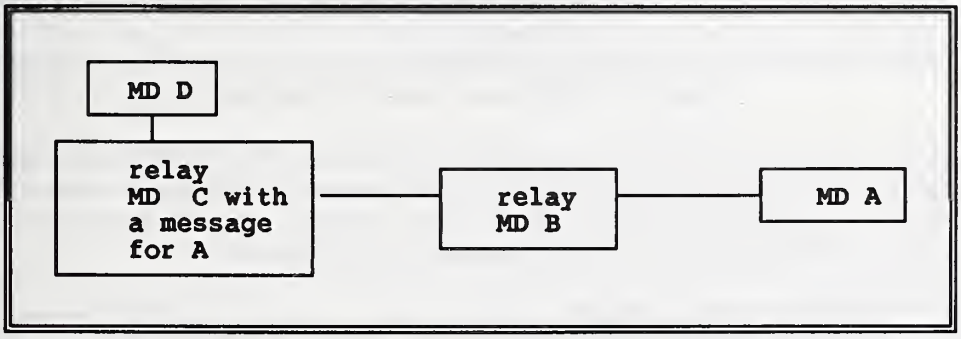


Figure 8 - MD C must know of A to route the message.

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 clause.

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 6.8 and 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 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 6.6).

7.3.2 Loop suppression within a PRMD

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 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 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.

Three actions are defined in MTASuppliedInfo: relayed, rerouted, and recipientReassignment as shown in figure 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 10 - Defined actions in MTASuppliedInfo.

7.3.3 Routing within a PRMD

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.

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 clause 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.

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.

7.3.3.1 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 17.

Table 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

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.

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.

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.

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.

7.3.3.2 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.

7.3.3.3 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.

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.

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.

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.

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

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.

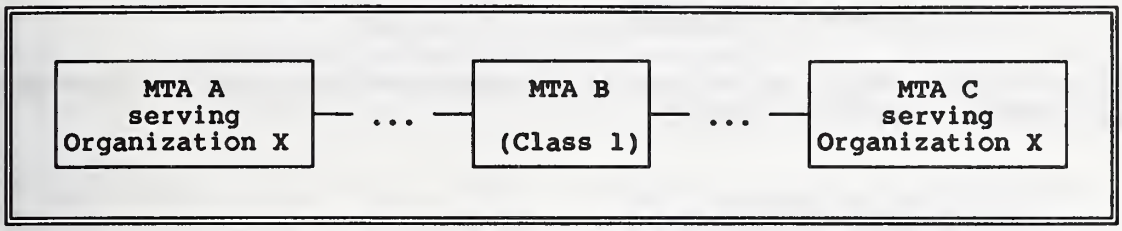


Figure 11 - Example of a configuration to be avoided.

In figure 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 11 can be corrected by replacing B with a Class 2 or Class 3 MTA.

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.

7.4 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.

7.5 X.400 protocol definitions

This clause describes additions and modifications to clause 5.3 which are required for implementation of a relaying PRMD or an MTA within a PRMD.

7.5.1 Protocol classification

The classification scheme given in clause 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.

In addition to the classifications given in clause 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.

7.5.2 P1 protocol elements

Table 18 contains protocol elements and their classes. An * signifies that the classification of the protocol element has not changed from table 8.

Table 18 - P1 protocol elements

Element	Class	Restrictions and Comments
UMPDUEnvelope		
MPDUIdentifier	M*	This field needs to be unique within a PRMD. See clause 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 clause 7.3.4 for the method of assuring uniqueness Maximum length = 32 characters.
MTAName		
MTASuppliedInfo	M	

Table 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 clause 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.

Table 18 - P1 protocol elements (concluded)

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)

Notes
1 The M classification is only applicable if an implementation is claiming conformance according to clause 10.2.

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.

8 Error handling

This clause 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.

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.

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.

8.3 Envelope

This clause 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.

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.

8.3.2 Protocol violations

If all required protocol elements are not present, a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of protocolViolation should be generated.

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.

8.3.3 O/R Names

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.

If any element in the O/R Name is unrecognized or if the CountryName, AdministrationDomainName, and one of PrivateDomainName 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).

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.

8.3.4 TraceInformation

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.

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. 8.3.2).

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.

8.3.5 InternalTraceInfo

This clause applies only to MTAs which follow the agreements of clause 7.

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. 8.3.2).

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.

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.

8.3.6.1 deferredDelivery

The delivering domain shall do one of the following:

a) deliver at once,

b) hold for deferred delivery,

c) return a nondelivery notification with a ReasonCode of unableToTransfer and a DiagnosticCode of noBilateralAgreement.

8.3.6.2 PerDomainBilateralInfo

If a delivering domain receives this element, the element can be ignored.

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.

8.3.6.4 alternateRecipientAllowed

If a delivering domain receives this element the element can be ignored.

8.3.6.5 contentReturnRequest

If a delivering domain receives this element, the element can be ignored.

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.

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.

8.3.7.2 ExplicitConversion

When an unexpected value is received for ExplicitConversion, it should be handled as in clause 8.3.6.3.

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.

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 clause 6.11.

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.

9 MHS use of Directory Services

9.1 Directory service elements

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:

a) Verify the existence of an O/R name.

b) Return the O/R address that corresponds to the O/R name presented.

c) Determine whether the O/R name presented denotes a user or a distribution list.

d) Return a list of the members of a distribution list.

e) When given a partial name, return a list of O/R name possibilities.

f) Allow users to scan directory entries.

g) Allow users to scan directory entries selectively.

h) Return the capabilities of the entity referred to by the O/R name.

i) Provide maintenance functions to keep the directory up-to-date.

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

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.

9.2 Use of names and addresses

It is recognized that these agreements enable a wide variety of naming and addressing attributes (see sec. 5.3.5 ORName Protocol Elements) wherein each PRMD may adopt particular routing schemes within its domain.

With the exception of the intra-domain connection agreements:

a) These agreements make no attempt to recommend a standard practice for electronic mail addressing.

Inter-PRMD addressing may be secured according to practices outside the scope of these agreements, such as:

a) manual directories

b) on-line directories

c) ORName address specifications

d) ORName address translation.

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 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.

10 Conformance

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 clause defines the conformance requirements and provides guidelines for the interpretation of the results from this type of testing.

This clause 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.

10.2 Definition of conformance

For this clause, the term **conformance** is defined by the following:

a) The tests indicated for this clause 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 clause.

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):

1) causes appropriate information to be displayed, or

2) 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 clause, it is not required that the inter-layer interfaces be available for testing purposes.

d) **Conformance** to the protocols requires:

1) that MPDUs correspond to instances of syntactically correct data units,

2) 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,

3) 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:

1) a domain, which may be one or more MTA's with co-located or remote UA's,

2) a single instance of an MTA and co-located UA with X.400 (P1, P2, RTS and session) software,

3) a relaying product with P1, RTS and session software,

4) a gateway product.

g) Claiming Implementation Conformance

1) An implementation which claims to be conformant as an ADMD must adhere to the agreements in clauses 5 and 6.

2) An implementation which claims to be conformant as a PRMD must adhere to the agreements in clause 5.

3) An implementation which claims to be conformant as a relaying PRMD must adhere to the agreements in clause 5 and the appropriate clauses of 7.

4) An implementation which claims to be conformant to the intra-domain connection agreements must adhere to the agreements in clause 5 and the appropriate clauses of 7.

10.3 Conformance requirements

10.3.1 Introduction

Conformance to this specification requires that all the services listed as supported in clauses 5, 6, and if appropriate, 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 annex 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.

10.3.2 Initial conformance

This clause 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 clauses.

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 clauses 5.2.1.1 and 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.

10.3.2.2 Service

For information available to the recipients via the IPMessage Heading and Body, the following should be made accessible:

a) iPMessage ID - only the PrintableString portion of the iPMessageid needs to be accessible.

b) subject,

c) primaryRecipients,

d) copyRecipients,

e) blindcopyRecipients,

f) authorizingUsers,

g) originator,

h) inReplyTo,

i) replyToUsers,

j) importance,

k) sensitivity,

l) iA5Text Bodypart.

Annex A (normative)

Interpretation of X.400 service elements

The work on service element definitions is limited to those that are defined as "supported" in clause 5 of this specification. Furthermore it is not the intent of this clause 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 clauses 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.

A.1 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 clause are as defined in X.400.

A.2 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.

A.3 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.

A.4 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.

A.5 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.

A.6 Registered encoded information types

A UMPDU with an "unspecified" value for Original Encoded Information Type shall be delivered to the UA.

A.7 Delivery notification

The UAContentID may be used by the recipient of the delivery notification for correlation purposes.

A.8 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.

A.9 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.

A.10 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.

A.11 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.

A.12 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.

A.13 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.

A.14 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.

A.15 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.

A.16 Forwarded IP message indication

The following use of the original encoded information type in the context of forwarded messages is clarified:

a) 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.

b) The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.

c) See annex B on recommended practices for the use of the delivery information as part of Forwarded IP-message.

A.17 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.

Annex B (informative)

Recommended X.400 practices

it is not necessary to follow the recommended practices when claiming conformance to these agreements.

B.1 Recommended practices in P2

B.1.1 ORDescriptor

Vendors following the OSI implementors' Workshop guidelines shall, whenever possible, generate the ORName portion of an ORDescriptor in ALL IPM heading fields.

B.1.2 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.

B.1.3 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.

B.2 Recommended practices in RTS

B.2.1 S-U-ABORT

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 min). In instances where this delay is not required or necessary, report a localSystemProblem.

B.2.2 S-U-EXCEPTION-REPORT

S-U-EXCEPTION-REPORT reason codes can be interpreted as follows:

B.2.2.1 receiving ability jeopardized (value 1)

Possible meaning: The receiving RTS knows of an impending system shutdown.

B.2.2.2 local ss-User error (value 5)

Possible meaning: The receiving RTS needs to resynchronize the session dialogue.

B.2.2.3 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.

B.2.2.4 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.

B.2.2.5 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.

B.2.3 OSI addressing information

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.

B.3 Recommended practices for ORName

Table 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 B.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.

B.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:

a) alternate Recipient Assignment

b) PROBE

c) Receipt Notification / Non-Receipt Notifications

d) 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.

PDA1 The country and postal code fields are free-form text.

PDA2 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"
```

B.5 EDI use of X.400

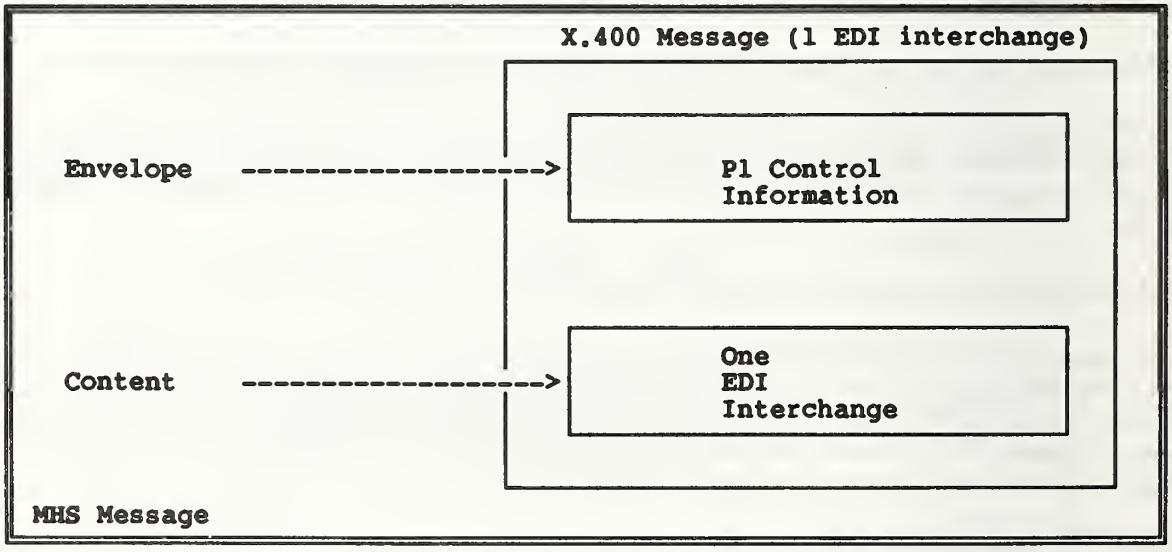
B.5.1 Introduction and scope

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.

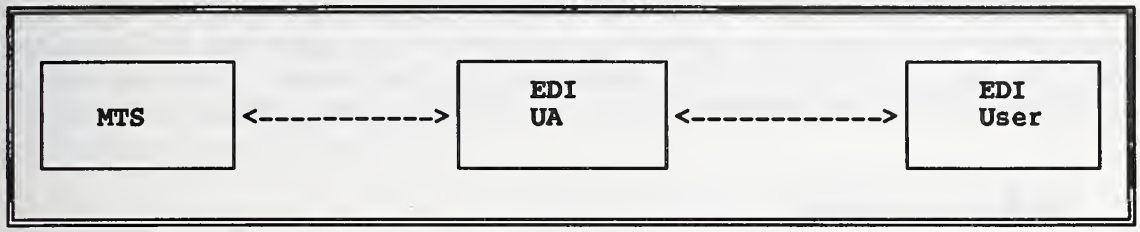
The scope of this guideline is to describe specific recommendations for adopting X.400 as the data transfer mechanism between EDI applications.

B.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.

B.5.3 Protocol elements supported for EDI

The following P1 protocol elements will be used to support EDI applications:

B.5.3.1 Content type

For EDI applications, the content type will be 0 (undefined content).

B.5.3.2 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.

B.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 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 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.

B.6 USA body parts

It is recommended that UAs can generate any USA Body Part, as defined in clause 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.

B.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.

B.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:

B.8.1 ODA 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.8.2 ODA In P1

The undefined bit (bit 0) of the EncodedInformationTypes must be set and the ODA bit (bit 10) of the EncodedInformationTypes should be set when an ODA document is present in P2. However, MTAs should be tolerant of messages containing ODA documents being received with just the undefined bit (bit 0) set, and should still deliver the message.

B.8.3 Interworking with later versions of X.400

The 1988 X.419 Recommendation acknowledges that a 1984 system may receive messages containing new distinguished [integer] values that it is not expecting, and that this may result in service irregularities. It is implied that it would be optimal for 1984 systems to accept these unexpected integer values if at all possible. (Note clause 8.3.7 of this section describes appropriate actions for unexpected values of the INTEGER fields in the P1 envelope.) No downgrading should be done for these values when passing affected messages from newer systems to older systems.

Annex C (normative)

Rendition of IA5Text and T61String characters

C.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.

C.2 Generating and imaging T61String

For further study.

Annex D (informative)

Differences in interpretation discovered through testing of the MHS for the CeBit 1987 demonstration

Several interworking problems were discovered through multi-vendor testing. These problems, and recommendations for solutions to them are discussed in this annex.

D.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.

D.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.

D.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.

D.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.
Encoding of per recipient flag and per message fFlag
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.

D.5 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.

D.6 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.

D.7 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.

D.8 Initial serial number in S-CONNECT

This should be implemented in accordance with section 3.5.1 E4 of the Implementor's Guide.

D.9 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.

D.10 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.

D.11 Old activity identifier

The Old Activity identifier in S-ACTIVITY-RESUME refers to the original activity identifier.

D.12 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 OIW agreements, since class 0 is only required if one of the partners in a connection is an ADMD.

Annex E (informative)

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 E.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

Table E.1 - Protocol element comparison of RTS (concluded)

RTS element	NIST	A/311	A/3211	PROBLEM (Y/N)
PAccept	G	G	G	N
DataTransferSyntax	M 0	M 0	M 0	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 E.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
		SO R	R		N
		.121	H		Y Conformance (E)
		ther	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		Conf(US)See Note 1
Terminal ID	X	X/G	X		Conf(US)See Note 1
PrivateDomainName	G	G	G	G	N
OrganizationName	G	G	G	G	N
UniqueUAidentifier	X	X/G	X		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 E.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	@ MT	H	H	Y Conformance (US)
		at U	?		Y Conformance (US)
ConversionProhibited	G	G	G	G	N
AlternatRecipAllowed	H	@ MT	H	X	Y Conformance (US)
		at U	?		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

Table E.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	┘	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

Table E.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-> OriginReqAlternatRecip			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

Table E.2 - Protocol element comparison of P1 (concluded)

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

Table E.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
BlindCopyRecipient	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	M	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

Table E.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	┐	N
				├M	
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

Table E.3 - Protocol element comparison of P2 (concluded)

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

NOTES

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.

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.

3 An A/3211 implementation may generate MOTIS encoded information types. See 6.11.

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.

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.

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.

7 TTC is expected to support IA5 for some international MHS communications.

Annex F (informative)

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 OSI Implementors' 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.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 8 - Message Handling Systems

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Neil Koorland, Microsoft**
SIG Editor: **Rich Ankney, Fischer International**

Foreword

The text in this part contains a set of Message Handling System (MHS) Implementation Agreements intended to serve in lieu of an International Standardized Profile (ISP) for MHS. It is the aim of the OIW X.400 SIG to pursue alignment of this part with the developing ISP. When the ISP is complete, this part will be revised to refer to the ISP, and to only highlight additional practices and North American regional requirements.

Table of Contents

List of Figures

List of Tables

Part 8 Message Handling Systems

0 Introduction

This is an implementation Agreement developed by the Implementors' 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 the CCITT X.400 (1988) series of Recommendations, the similar (but not identical) ISO MOTIS standard, and Recommendations F.435 and X.435 (1991) (see References). These Recommendations and Standards are referred to as the *base standards*. 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:

a) Preventing a proliferation of incompatible communities of MHS systems which are isolated for protocol reasons;

b) Achieving interworking with implementations conforming to the OIW Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems; and,

c) Facilitating integration of other OSI-based services (e.g., Directory) within a single real system.

This implementation Agreement is designed to encourage upgrade of existing 1984-based systems as follows:

a) To add 1988 functionality (Message Store, Remote User Agent, etc.);

b) To provide additional functionality above the minimal conformant 1988 MHS defined in the December 1989 version of the OIW implementation Agreements. These 1988 aspects are described in this Agreement as either incremental enhancements or new functional groups.

However, it is considered that the OIW Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems (part 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.

1 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 1:

a) Management Domain (MD) to MD;

b) Message Transfer Agent (MTA) to MTA within a domain;

c) MTA to remote Message Store (MS) or User Agent (UA); and,

d) 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 1.

MHS implementations may be configured as any single or multiple occurrence or combination of MTA, MS and UA, as illustrated in figure 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).

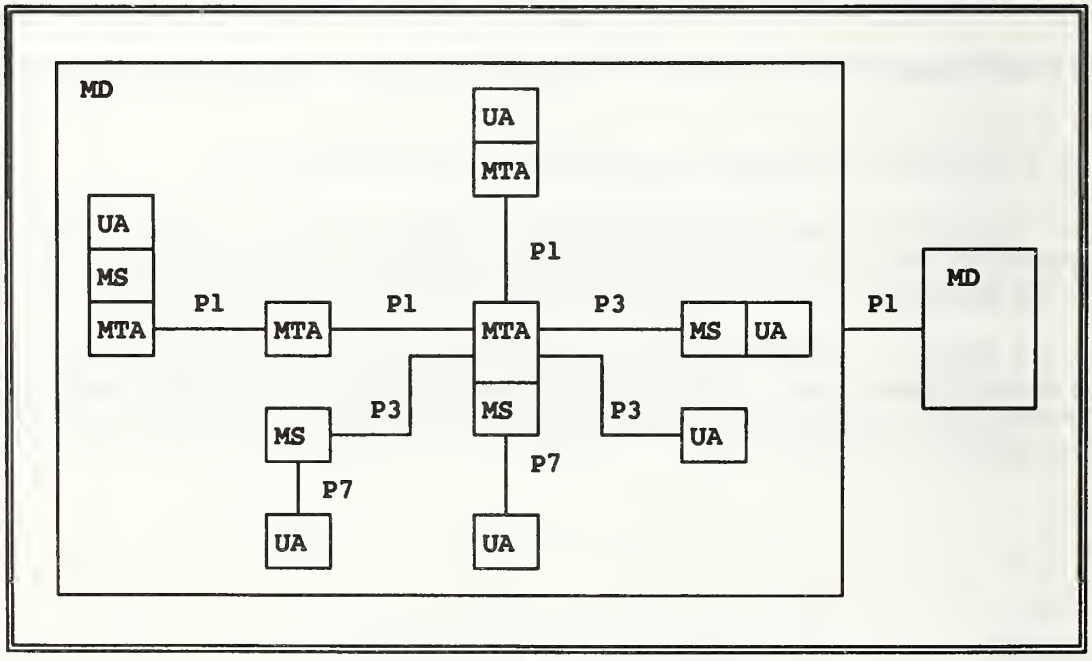


Figure 1 - Scenario definition

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. Conformance requirements for support of Functional Groups by particular configurations are specified in clause 17.

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 submission, transfer, or delivery must be processed according to the base standards.

The following functional groups are covered by this Implementors Agreement:

a) The MT Kernel in clause 5;

b) The Message Store in clause 6;

c) Remote User Agent support in clause 7;

d) Distribution Lists in clause 8.3;

e) Use of Directory in clause 8.4;

f) Address support for Teletex character sets in clause 8.5;

g) MHS Management in clause 9 (which is for further study);

h) Security in clause 10;

i) The Physical Delivery Access Unit in clause 11.1;

j) Other Access Units in clause 11.2 (which are for further study);

k) Redirection in clause 12 (which is for further study); and,

l) The IPM Service in clause 13;

m) The EDI Messaging Service in clause 14 (which is for further study).

2 References

2.1 CCITT

Application Layer - MHS

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.*

CCITT Recommendation X.121 (1988), *International Numbering Plan.*

CCITT Recommendation X.435 (1991), *Message Handling Systems, EDI Messaging System, Protocol Specifications.*

CCITT Recommendation F.435 (1991), *Message Handling Systems, EDI Messaging System, Abstract Service Definition.*

2.2 ISO

Application Layer - MHS

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.*

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, as amended by the *MHS Implementors Guide*, version 6.

The initial version of these Stable implementation Agreements included an Agreement which specified a minimal 1988-based MHS implementation and support for Message Stores and Remote User Agents, and which addresses interworking with 1984-based implementations. This version of the Agreement specifies support for several additional 1988 features. 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 necessary in the future for international harmonization, e.g., support for international character repertoires and conversion.

4 Errata

No Errata to Stable material at this time.

5 MT kernel

5.1 Introduction

This clause 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 clause does **not** include:

a) Message Store (see clause 6);

b) Remote UA (see clause 7);

c) Distribution Lists (see clause 8.3);

d) Use of Directory Services (see clause 8.4);

e) Security (see clause 10);

f) Interworking with Physical Delivery systems or Specialized Access (see clause 11); and,

g) Conversion of body parts (see clause 13.6.2).

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:

a) It will be protocol-conformant to 1988 P1;

b) It will downgrade 1988 P1 to 1984 P1 when relaying to 1984-based MTAs, as specified in Annex B of X.419 (see clause 5.5);

c) It will support both "normal" mode and "X.410-1984" ("passthrough") mode protocol stacks (i.e., as required by ISO and CCITT respectively); and,

d) A conforming implementation shall obey the criticality mechanism defined in the base standards. The following abstract operations are made critical for delivery for these Implementation Agreements: message token, content integrity check, and content confidentiality algorithm id.

5.2 Elements of service

This clause specifies the requirements for support of MT Elements of Service by an MTA conforming to the MT Kernel Functional Group of this Agreement. Table 1 specifies the support for the basic MT Kernel elements of service and table 2 specifies the support for the optional MT Kernel elements of service.

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;

Out of Scope (I): the Element of Service is outside the scope of these Implementation Agreements;

Not Applicable (-): the Element of Service is not applicable in the particular context according to the base standard; and,

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 section 19 of X.400 (1988) as listed and qualified in tables 1 and 2.

Specification of dynamic behavior in these agreements will only be included in those cases where there is

an identified functional objective which is not satisfied by the specification of dynamic behavior in the corresponding base standard(s) and where the resulting behavior does not breach base standard conformance requirements.

In these exceptional cases, there may be situations where these agreements must specify the dynamic behavior of an implementation as distinguished in annex C of ISO TR-10 000. Where this occurs, a table of dynamic conformance requirements will be presented using the classification scheme below:

Mandatory (M): The element must be implemented although use is not required for conformance to the base standard. The element shall always be used for conformance to these agreements.

Excluded (X): This element must either not be implemented, or it must be possible to prevent use of the element.

NOTE - As stated in clause 6.7 of ISO TR-10 000-1, restrictions by a profile on the dynamic conformance requirements of a base standard are exceptions, and should only apply to transmission. Restrictions should not apply to reception. In the case of Excluded options, it must be possible to ensure that such options are not initiated or transmitted. However, it is still possible that an implementation may receive an Excluded element from an implementation which does not conform to the same profile.

Table 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 collocated UA/MTA and/or MS/MTA configurations.

Table 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	M	-	-
Probe	M	M	M
Redirection Disallowed by Originator (1988)	M	M	-
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 collocated 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.

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 clause 8.3).

5 Support of this MT Element of Service does not imply the capability to perform DL expansion (see clause 8.3).

6 Messages should be held in the originating MTA to provide support for this element of service.

5.3 MTS transfer protocol (P1)

The requirements for support of MTS Transfer Protocol (P1) elements are detailed in clause A.1.

Support of MTS Transfer Protocol application contexts by an MTA is classified as in table 3.

Table 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 clause 15.

5.4 MTS - APDU size

This clause is not intended to constrain the size of PDUs that are transferred across the network, since some body part types and content types (e.g., voice, file transfer, and EDI) may require very large PDUs.

The following agreements govern the size of MTS-APDUs:

a) All MTAEs must support at least one MTS-APDU of at least two megabytes; and,

b) The size of the largest MTS-APDU content supported by a UAE is a local matter.

5.4.1 Number of recipient names

There is no specified bound on the number of recipient-names an implementation must support, other than the 32K-1 specified in the standard (Annex B/X.411).

5.5 1988/84 interworking considerations

An MTA conforming to this Agreement will downgrade 1988 P1 to 1984 P1 when relaying to 1984-based MTAs, as specified in Annex B of X.419 with the following additional requirements:

a) Supplementary information - will need to be truncated if it exceeds the pragmatic constraint identified in Version 2 of these Agreements (64 octets as opposed to 256 octets in the 1988 MHS standards);

b) ISO DIS 8883 Extensions - An implementation may perform the mapping of ISO DIS 8883 extensions to existing 1988 services when relevant, but is not obliged to. Alternatively, it may discard the extensions or generate a non-delivery report;

c) Internal Trace Information - If the 1984-based MTA does not support Internal Trace Information per clause 7.3.2 of part 7, the following description is not applicable. When a 1988-based MTA supports interworking with a 1984-based MTA that generates Internal Trace Information as per clause 7.3.3 of part 7, the 1988-based MTA must support reception of the Internal Trace Information by converting the Internal Trace Information from the form in clause 7.3.2 of part 7 to the form specified in 1988 X.411, as per the following description. When the 1988-based MTA sends to a 1984 MTA, the 1988-based MTA must apply the conversion to 1984, as described below. The Stable NBS Implementation Agreements X.400 (1984) definition for MTA's Internal Trace Information is different from the X.400 (1988) MTA definition. Consequently, a X.400 (1988) MTA operating in an MD with other MTAs of 1984 vintage, must map the Internal Trace Information to and/or from the 1984 format.

Figures 2 and 3 depict algorithms for mapping between X.400 (1988) Internal Trace element formats and the OIW IA X.400 (1984) Internal Trace element format.

To avoid potential looping within a MD composed of 1984 and 1988 vintage MTAs, MD administrators are strongly advised to name all MTAs (1984 and 1988 vintages) using only the Printable String characters. In X.400 (1988) the MTA-Name is defined to be named using IA5 String characters where in the IAs for X.400 (1984) MTAs, NBS restricted the MTA-Name to be formed using the Printable String character subset of IA5. If the 1988-based MTA Name uses IA5 characters not in the Printable String subset, that Internal Trace Element should be omitted when converting from 1988 to 1984.

```
For each Internal Trace element in the sequence:
DO
  IF MTA-Name is made up of non-Printable String characters:
    Discard this Internal Trace element;
  ELSE
    {  Discard the GlobalDomainIdentifier;
       Copy the MTAname over;
       Within the MTASuppliedInformation:
         Copy the arrival time over;
         Copy the routing action over;
         IF attempted is present
           {  IF it is a domain:
                Discard it;
              IF it is an MTA:
                Copy it to Previous MTAName;
           }
         IF the additional actions are present:
           {  IF the deferred time is present:
                Copy it over;
              IF the other-actions is present:
                Discard it;
           }
    }
END-DO
```

Figure 2 - 1988 to 1984 mapping

```
Find the [APPLICATION 30] entry in the P1 envelope;
FOR each Internal Trace element:
  DO
    Insert the GlobalDomainIdentifier of this MTA;
    Copy the MTAName over;
    Within the MTASuppliedInfo:
      Copy the arrival time;
      IF the deferred time is present:
        copy it to the additional actions field within the
          1988 Internal Trace information;
      IF the routing action is Relayed or Rerouted:
        copy it over;
      IF the routing action is Recipient-reassigned:
        map to Relayed;
      IF the previous MTAName is present:
        copy it to the MTAName in the attempted field;

  END-DO
```

Figure 3 - 1984 to 1988 mapping

NOTE - The 1988 X.419 Recommendation acknowledges that a 1984 system may receive messages containing new distinguished [integer] values that it is not expecting, and that this may result in service irregularities. It is implied that it would be optimal for 1984 systems to accept these unexpected integer values if at all possible. No downgrading should be done for these values when passing affected messages from newer systems to older systems.

6 Message store

6.1 Introduction

This clause 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 4 illustrates the logical relationship of the MS to the UA and MTS.

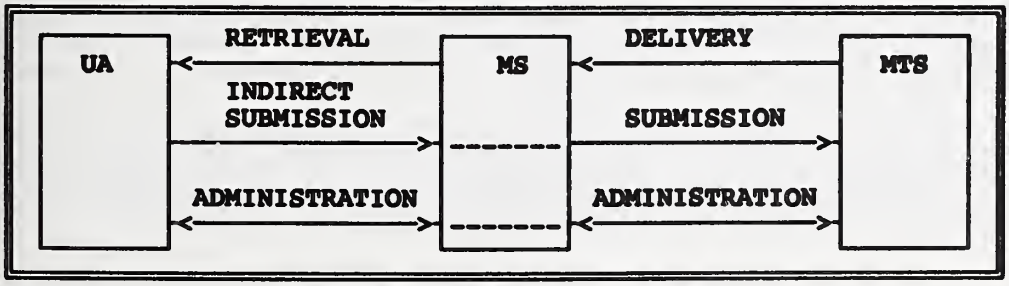


Figure 4 - Message store model

The Agreements in this clause specify the Message Store's use of the retrieval, delivery, and administration

services. Agreements on submission services are specified in clause 7, which describes support for the Remote UA.

The goal of the Agreements in this clause 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).

6.2 Scope

The scope of the Agreements in this clause is depicted in figure 5, 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.

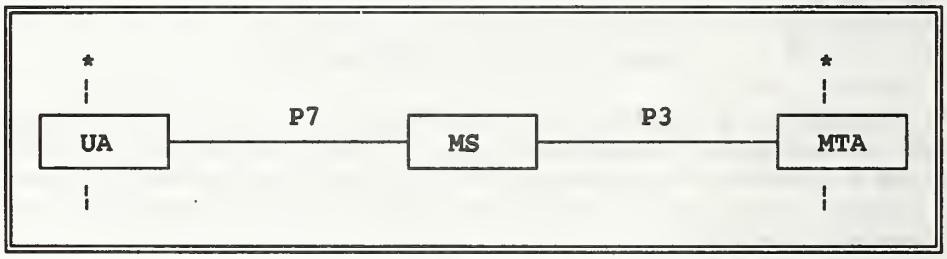


Figure 5 - Scope of message store agreements

The UA, MS and MTA configuration is not restricted; any of these components may be collocated, although they are depicted as logically separate. In the case of a collocated UA and MS, a proprietary interface may be used instead of P7. In the case of a collocated MS and MTA, a proprietary interface may be used instead of P3.

6.3 Elements of service

This clause 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 clause 5.2.

Support for Elements of Service is specified in table 4 both for the Message Store itself and for the User Agent.

Table 4 - Message store: elements of service

Element of Service	UA	MS
MS Register	O	M
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

6.4 Attribute types

Requirements for support of the attributes used in the Message Store are detailed in clauses A.8 and A.9.

There are two levels of support for General Attributes in the Message Store.

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 content-specific attributes.

The Enhanced MS supports a larger number of general attributes and is suited to MSs that also support content-specific attritbutes.

Additionally, support for security attributes is defined in clause A.9, for use in secure environments.

Refer to the content-specific clauses for support for content-specific attributes.

6.5 Pragmatic constraints for attribute types

There are no additional pragmatic constraints for attribute types beyond those of the base standards.

6.6 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 clause A.4.

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 5.

Table 5 - 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 clause 15.

6.7 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** collocated with the MTA are detailed in clause A.3.

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 sections 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 6.

Table 6 - 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 clause 15.

7 Remote user agent support

7.1 Introduction

This clause specifies Agreements for implementation of the Remote User Agent Functional Group, i.e., for support of an UA that is **not** collocated with its MTA.

The goal of the Agreements in this clause 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), and independent of any particular content type. The content-specific requirements for UAs are specified in the content-specific sections of this part of the implementor's Agreements.

7.2 Scope

The scope of the Agreements in this clause is depicted in figure 6, 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 clause 6.

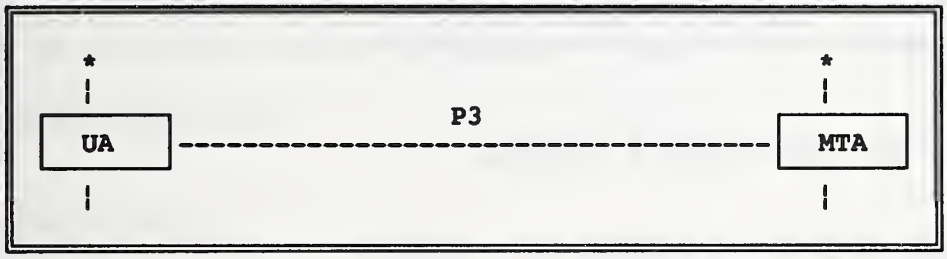


Figure 6 - Scope of remote user agent agreements

7.3 Elements of service

This clause 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 clause 5.2.

Support for Elements of Service is specified for the MT Service (table 7) and is in addition to the support requirements specified in clauses 5 and 13 if this Functional Group is supported.

Table 7 - 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

7.4 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** collocated with the MTA are detailed in clause A.3.

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 sections 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.

Table 8 - 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 clause 15.

8 Naming, addressing & routing

8.1 Use of O/R addresses for routing

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

8.2 ORAddress attribute list equivalence rules

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

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

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

b) Over-specified ORAddresses are not equivalent. Over-specification means that more attributes are present in the provided ORAddress than are present in the known ORAddresses, however, unrecognized DDA types may be ignored for these purposes.

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

d) An extension attribute encoded as Teletex String shall be compared with the corresponding standard attribute encoded as Printable String if that extension attribute is not present in both ORAddresses. Matching rules are as specified in clause 18.4 of X.402 (1988) (as modified in the

MHS Implementors' Guide) except that only teletex graphic characters from repertoire no. 102 need to be compared for Printable String equivalence (i.e., the presence of graphic characters from other repertoires can be treated as a mismatch).

NOTES

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

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

8.3 Distribution lists

8.3.1 Introduction

This clause identifies and specifies the Distribution Lists Functional Group, which covers all issues relating to the performance of distribution list (DL) expansion by an MTA. Other aspects concerned with the use of distribution lists are covered in the MT Kernel Functional Group.

8.3.2 Elements of service

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

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

Support for Elements of Service is specified for the MT Service only (table 9), and is only concerned with the performance of DL expansion by an MTA. Such support is in addition to the support requirements specified in clause 5 if this Functional Group is supported.

Table 9 - Distribution lists: MT elements of service

Element of Service	Support
DL Expansion History Indication	M
DL Expansion Prohibited	M
Use of Distribution List	M1

Notes
1 Use of DL Names is always possible because a DL name cannot be distinguished from any other OR Name on origination.

8.4 MHS use of Directory

8.4.1 Introduction

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

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

8.4.2 Functional configuration

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

8.4.3 Functionality

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

a) UA Specific Functionality:

1) Verify the existence of a Directory Name;

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

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

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

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

6) Return the members of a Distribution List;

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

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

b) MTA Specific Functionality:

1) Authentication;

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

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

4) Return the members of a Distribution List;

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

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

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

8.4.4 Naming and attributes

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

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

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

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

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

NOTE - The mhs-user object class is a generic object class which may be used in conjunction with another standard object class for the purpose of adding MHS information attributes, such as ORAddresses, to a Directory entry. The means to associate attributes of a generic object class to an entry (or to different entries) named by a standard object class(es) is by defining a new (un-)registered object class, whose superclass(es) is that of the naming object class(es), and of the generic object class. E.g., to associate mhs-user attributes in the organizationalPerson entry, a new unregistered object class can be defined as shown in figure 7.

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

Figure 7 - Example of unregistered object class definition

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

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

8.4.5 Elements of service

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

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

Support for Elements of Service is specified both for the MT Service (table 10).

Table 10 - Use of directory: MT elements of service

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

8.4.6 Directory services

These Implementation Agreements require the Directory services as defined in table 11. Indicated are the Directory services required to support the needs of the MHS UA/MTA and MHS Administrator.

Table 11 - Directory service support requirements

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

8.4.7 OIW X.400 base Directory Implementation Agreements

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

8.4.7.1 Other profiles supported

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

8.4.7.2 Standard application specific attributes and attribute sets

The standard application specific attributes and attributes sets supported by these Implementation Agreements are listed in table 12. For each attribute and attribute set, a reference is provided to the standard where it is defined.

Table 12 - Standard attributes and attribute sets

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

8.4.7.3 Standard application specific object classes

The standard application specific object classes supported by these Implementation Agreements are listed in table 13. For each object class, a reference is provided to the standard where it is defined.

Table 13 - Standard object classes

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

8.4.7.4 OIW application specific attributes and attribute sets

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

8.4.7.5 OIW application specific object classes

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

8.4.7.6 Structure rules

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

8.4.7.6.1 MHS Distribution List

Attribute commonName is used for naming.

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

8.4.7.6.2 MHS User

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

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

8.4.7.7 Use of Capabilities Information

The capabilities information in the X.500 Directory should not be considered sufficient to warrant a non-delivery decision by an originating or relaying MTA. This clause is not intended to impose any conformance requirement.

8.5 Address support for Teletex character sets

This clause identifies the Address Support for Teletex Character Sets Functional Group, which covers the generation of Teletex strings in OR Address components.

Support of this functional group implies that, if an address component is supported for origination, the corresponding Teletex component (if any) must be supported for origination.

8.6 Reply support

When originating a reply, the UA must be able to utilize the applicable addressing components of the message to which it is replying (regardless of character set support level).

9 MHS management

NOTE - For further study.

10 MHS security

10.1 Overview

The Security functional group is specified as three security classes which are incremental subsets of the security features available in the base standard. They are denoted as S0, S1, and S2. An implementation that conforms to the Security functional group map support one or more of the security classes defined in these Implementation Agreements.

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

S1: This security class requires secure functionality with the MTS-User and MTS. The MTS secure functionality is only required to achieve secure access management. As with S0, most of the security mechanisms are implemented within an MTS-User. It primarily provides integrity and authentication between MTS-Users. However, MTAs are expected to support digital signatures for peer to peer authentication, security labelling and security contexts.

S2: This security class is a superset of S1, adding security functions within MTAs and the MTS. The main security function added within this group is authentication within the MTS, and, as a consequence, due to the non-repudiable nature of the keys used for authentication, non-repudiation is also added.

In addition, each of the three security classes has a variant, denoted as S0a, S1a, and S2a, which mandates support of end-to-end confidentiality.

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

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

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

Table 14 provides an overview of the requirements made by the security classes on the MTS-User and MTA. The table entries are descriptive, and are not intended to refer to security service elements.

Table 14 - Overview of security requirements for each security class.

Class	Requirements	
	MTS-User	MTA
Kernel		Submission, delivery, and relay of EoS
S0	Content Integrity, Proof of Delivery, Message Origin Authentication (UA to UA)	Kernel
S0a	S0 plus Content Confidentiality	Kernel
S1	S0 plus Message security label, Message security context, Security Management Services	Peer entity authentication, Security context, Security Management Services, and Message Security Label
S0a	S1 plus Content confidentiality	S1
S2	S1 plus Message Origin Authentication Check, Probe Origin Authentication Check, Report Origin Authentication Check, Proof of Submission, and, Non-repudiation	S1 plus Message Origin Authentication Check, Prove Origin Authentication Check, Report Origin Authentication Check, Proof of Submission, and, Non-repudiation
S2a	S1a plus S2	S1a plus S2

The incremental functionality of the security classes can be represented diagrammatically as shown in figure 8.

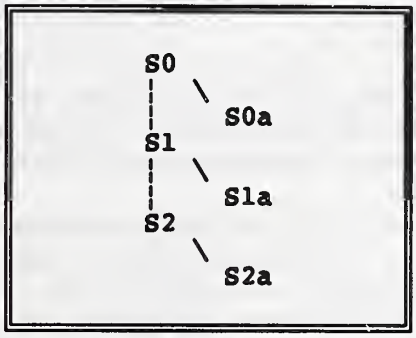


Figure 8 - Incremental functionality of the security classes

10.2 Common requirements

10.2.1 Interworking between security classes

A security class can be viewed as a tool which can be used to implement a security policy, and is not a security policy in its own right but a component of a security policy.

Interworking between implementations supporting different security classes can be achieved in terms of any common class(es) supported. As specified in the base standard, the label of the message, probe or report must be checked against the security context by any implementation claiming conformance to classes S1, S1a, S2, and S2a.

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

This profile defines security policy identifiers (annex B, figure 18) that corresponds to the security classes defined in this section. Such generic security policy identifiers only imply support of the X.400 security services as specified for these security classes in this clause. No other COMSEC or COMPUSEC functionality can be assumed by use of such policy identifiers. More specific security policies may be based on one or more of the security classes in this section but will require use of registered policy identifiers.

10.2.2 Comparison of security labels

The Security Content service ensures that the message security label matches at least one of the set of labels specified in the security content established between the communicating MHS entities.

An MTA which supports the Security Content service shall as a minimum support matching for equality on the security-policy-identifier, security-classification, and security-categories elements of the label.

NOTE - The basic support requirement is that absence of an element shall not be treated as "any value," i.e., all permissible combinations of occurrence and value for the elements of the message security label must be elaborated in the security context.

Any other matching rules (e.g., covering the privacy-mark element or based on alternative methods of comparison may be used in particular application scenarios, but such specification and usage will be subject to bilateral agreement and will depend on the security policy in force.

The message security label can be placed in the per-message extensions or in the signed or encrypted data of the per-recipient message token. It is recommended that the integrity of the security label is protected by including it in the token signed data, or (if the label is in the per-message extensions) by computing the message origin authentication check on the message. (Support of MOAC is optional in security classes S0 and S1.) Which of these labels is/are checked by the security context service is dictated by the security policy in force. The security policy should also define any requirements on allowable (per-recipient) label values in the case where the message is addressed to multiple recipients (and thus has multiple tokens).

A label may also be included in the token encrypted data with (confidential) end-to-end semantics.

10.2.3 Application context

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

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

10.3 Description of security classes

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

Figure 9 explains the column headings used in the security class tables. The classifications are defined in clause 5.2.

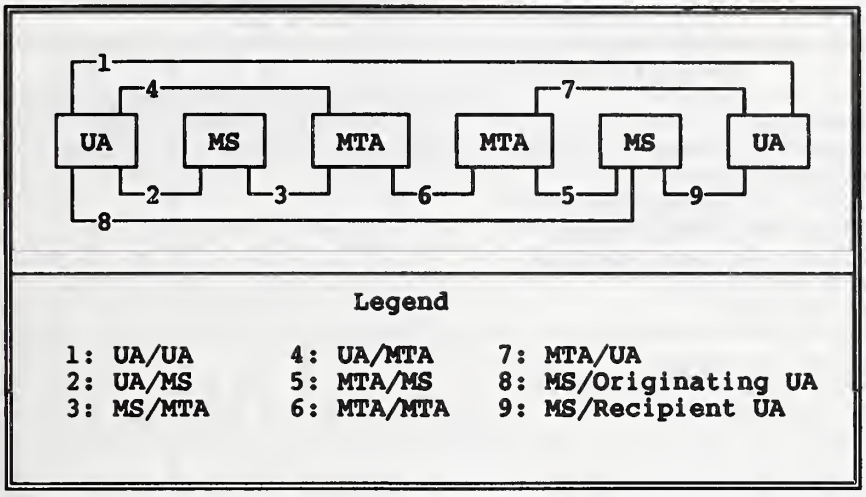


Figure 9 - Security interfaces

10.4 Security class 0 (S0)

10.4.1 Security functionality

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

a) integrity of message content;

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

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

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

There are no requirements placed on the MTA.

10.4.2 Security services for S0

Security class 0 (S0) mandates the security services listed in table 15.

Table 15 - Security class 0 (S0)

Security Service \ Security Interface	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Origin Authentication									
Message Origin Authentication[1]	M	I	-[6]	I	-	-	-	-	-
Probe Origin Authentication	-	I[6]		I	-	-	-	-	-
Report Origin Authentication	-	-	-	-	I	I	I	-	-
Proof of Submission	-	-	-	-	-	-	I	-	-
Proof of Delivery	M	-	-	-	-	-	-	M[4]	-
Secure Access Management									
Peer Entity Authentication[2,7]	-	O	O	O	O	O	O	-	O
Security Context	-	O	O	O	O	O	O	-	O
Data Confidentiality									
Connection Confidentiality[8]	-	I	I	I	I	I	I	-	I
Content Confidentiality	I	-	-	-	-	-	-	-	-
Message Flow Confidentiality	I	-	-	-	-	-	-	-	-
Data Integrity Services									
Connection Integrity[8]	-	I	I	I	I	I	I	-	I
Content Integrity	M	-	-	-	-	-	-	-	-
Message Sequence Integrity[11]	O	-	-	-	-	-	-	-	-
Non-Repudiation									
Non-Repudiation of Origin[1,5]	O	-	-	I	-	-	-	-	-
Non-Repudiation of Submission	-	-	-	-	-	-	I	-	-
Non-Repudiation of Delivery[5,10]	O	-	-	-	-	-	-	O	-
Message Security Labelling[2,3]	O	O	O	O	O	O	O	O	O
Security Management Services									
Change Credentials	-	O	-	O	O	I[9]	O	-	-
Register	-	O	-	O	-	-	-	-	-
MS-Register	-	O	-	-	-	-	-	-	-

Table 15 - Security class 0 (S0) (concluded)

Notes
1 Only provided to the message recipient.
2 Using either symmetric or asymmetric algorithms as identified by the algorithm identifier in the applicable protocol element.
3 When security labelling is used, the security policy identifier shall be included.
4 If Proof of Delivery and Content Confidentiality are both used, and delivery is to an MS, then proof of delivery can only be computed on the encrypted content. It should be noted that this will not provide non-repudiation of delivery.
5 Using either a trusted notary (symmetric) or using certificates tokens which are not repudiable (asymmetric).
6 Corrects table 7 of X.402 in the base standard.
7 Authentication between collocated objects is a local issue.
8 Refer to section 10 of X.402 and ISO/IEC 10 021-2 and IS 7498-2.
9 These services are expected to be provided by non-standard management services and are therefore outside the scope of this Implementors Agreement.
10 Non-Repudiation of Delivery can only be provided when the proof-of-delivery service is used.
11 Allocation and management of sequence numbers is outside the of this Implementors Agreement (as it is subject to bilateral agreements).

10.5 Security class 0A (S0a)

10.5.1 Security functionality

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

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

b) Content Confidentiality.

10.5.2 Security services for S0a

Security class 0A (S0a) mandates the security services of class S0 plus those listed in table 16.

Table 16 - Security class 0A (S0a)

Security Interface / Security Service	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Data Confidentiality Content Confidentiality	M	-	-	-	-	-	-	-	-

10.6 Security class 1 (S1)

10.6.1 Security functionality

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

a) Authentication of MTA, MS, and UA;

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

c) Integrity of message content;

d) Authentication of message originator;

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

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

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

h) MLS-mediation by secure access measures.

NOTES

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

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

10.6.2 Security services for S1

Security class 1 (S1) mandates the security servicesof class S0 plus those listed in table 17.

Table 17 - Security class 1 (S1)

Security Interface / Security Service	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Origin Authentication									
Message Origin Authentication[2]	M[1]	I	-	I	-	-	-	-	-
Secure Access Management									
Peer Entity Authentication[3,4]	-	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	-	M[1]
Security Context	-	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	-	M[1]
Data Integrity Services									
Content Integrity	M[1]	-	-	-	-	-	-	-	-
Message Security Labelling[3]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]	M[1]
Security Management Services									
Change Credentials	-	M	-	M	M	I[5]	M	-	-
Register	-	M	-	M	-	-	-	-	-
MS-Register	-	M	-	-	-	-	-	-	-

Notes
1 Shall always be used.
2 Only provided to the message recipient.
3 Using either symmetric or asymmetric algorithms as identified by the algorithm identifier in the applicable protocol element.
4 Authentication between collocated objects is a local issue.
5 These services are expected to be provided by non-standard management services and are therefore outside the scope of this Implementors Agreement.

10.7 Security class 1A (S1a)

10.7.1 Security functionality

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

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

b) Content Confidentiality.

10.7.2 Security services for S1a

Security class 2A (S1a) mandates the security services of class S1 plus those listed in table 18.

Table 18 - Security class 1A (S1a)

Security Interface / Security Service	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Data Confidentiality Content Confidentiality	M	-	-	-	-	-	-	-	-

10.8 Security class 2 (S2)

10.8.1 Security functionality

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

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

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

10.8.2 Security services for S2

Security class 2 (S2) mandates the security services of class S1 plus those listed in table 19.

Table 19 - Security class 2 (S2)

Security Interface / Security Service	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Origin Authentication									
Message Origin Authentication[3]	M[1]	M[1]	-	M[1]	-	-	-	-	-
Probe Origin Authentication	-	M[4]	-	M[1]	-	-	-	-	-
Report Origin Authentication	-	-	-	-	M[1]	M[1]	M[1]	-	-
Proof of Submission	-	-	-	-	-	-	-	M	-
Non-Repudiation									
Non-Repudiation of Origin	M[5]	-	-	M[2]	-	-	-	-	-
Non-Repudiation of Submission	-	-	-	-	-	-	M[2]	-	-
Non-Repudiation of Delivery	M[5]	-	-	-	-	-	-	M[2]	-

Notes
1 Shall always be used.
2 Using an asymmetric mechanism (i.e., certificates and tokens which are not repudiable for authentication within MTAs and the MTS.
3 Using the Message Origin Authentication Check as detailed in the base standard.
4 Shall always be used, and corrects table 7 in X.402.
5 Using either a trusted notary (symmetric) or using certificates tokens which are not repudiable (asymmetric).

10.9 Security class 2A (S2a)

10.9.1 Security functionality

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

a) Security functionality defined for security class S2; and,

b) Content Confidentiality.

10.9.2 Security services for S2a

Security class 2A (S2a) mandates the services of class S2 plus those listed in table 20.

Table 20 - Security class 2A (S2a)

Security Interface / Security Service	1 UA/ UA	2 UA/ MS	3 MS/ MTA	4 UA/ MTA	5 MTA/ MS	6 MTA/ MTA	7 MTA/ UA	8 MS/ UA	9 MS/ UA
Data Confidentiality Content Confidentiality	M	-	-	-	-	-	-	-	-

11 Specialized access

11.1 Physical delivery

This clause identifies and specifies the Physical Delivery Functional Group, which is intended to cover all issues relating to access to physical delivery systems by an MHS implementation.

11.1.1 Elements of service

This specifies the requirements for support of Elements of Service for conformance to the Physical Delivery Functional Group of this Agreement.

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

Support for Elements of Service is specified for:

a) the MT Service in table 21;

b) the O/R Address Attributes in table 22; and,

c) the character string support in table 23.

Editor's Note - table 23 does not appear in this part.

NOTE - All Elements of Service listed in table 21 are 1988.

Table 21 - Physical delivery: MT elements of service

Element of Service	UA Origination	PDAU Reception
Additional Physical Rendition	O	O
Basic Physical Rendition	M	M
Counter Collection	M	M
Counter Collection with Advice	O	O
Delivery via Bureaufax Service	O	O
EMS (Express Mail Service)	M	M
Ordinary Mail	M	M
Physical Delivery Notification by MHS	O	O
Physical Delivery Notification by PDS	O	O
Physical Forwarding Allowed	M	M
Physical Forwarding Prohibited	M	M
Registered Mail	O	O
Registered Mail to Addressee in Person	O	O
Request for Forwarding Address	O	O
Special Delivery	M	M
Undeliverable Mail with Return of Physical Message	M	M

Table 22 - Character string support

Character String	Origination (UA)	Reception (PDAU)
Printable Teletex	M O[1]	M O[2]

Notes
1 Mandatory if "Address Support for Teletex Character Sets" functional group is supported.
2 Mandatory if "Address Support for Teletex Character Sets" functional group is supported, with a minimum of one character repertoire.

11.2 Other access units

11.2.1 Facsimile access units

NOTE - The possible development of Agreements in this area is for further study.

11.2.2 Telex access units

It is not currently intended to develop Agreements in this area.

11.2.3 Teletex access units

It is not currently intended to develop Agreements in this area.

12 Redirection

The redirection functional group is for further study.

13 IPM service

13.1 Introduction

This clause specifies the requirements for a minimal 1988-based IPMS implementation (i.e., IPM UA) which is capable of interworking with 1984-based UAs.

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:

a) It will continue to support content type P2 (encoded as integer 2) on origination and reception;

b) It will support receipt of P2 (encoded as integer 22);

c) It may originate P2 encoded as integer 22, but the guidelines specified in section 8.18.2 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 clause A.3, or MS Submission and Retrieval based on the protocol classifications in clause A.4. However, how such information is conveyed to/from the MTS or MS in the case of a collocated UA is a local matter, and will not necessarily be subject to conformance verification.

13.2 Elements of service

This clause 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 clause 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 section 19 of X.400 (1988) as listed and qualified in tables 23 and 24.

Table 23 - 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
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 collocated UA/MTA or collocated 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 24 - IPM kernel: IPM service optional user facilities

Element of Service[1]	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[2]
Obsoleting Indication	O	M
Originator Indication	M	M
Originator Requested Alternate Recipient (1988)	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	-

Table 25 - IPM kernel: IPM service optional user facilities (concluded)

Element of Service	Orig	Recep
Restricted Delivery (1988)	-	O
Return of Content	O	-
Sensitivity Indication	O	M
Subject Indication	M	M
Use of Distribution List (1988)	O	-

Notes
1 Other UA Elements of Service are listed in Table 4.
2 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.

13.3 Interpersonal messaging protocol (P2)

The requirements for support of Interpersonal Messaging Protocol (P2) elements are detailed in clause A.2.

13.4 Body part support

This clause specifies the requirements for support of IPM body part types by a UA conforming to this Agreement.

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 33 of Annex A of this part. 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. If an implementation supports origination of forwarded messages, it must be capable of forwarding every body part that is supported on reception. The reception requirements on the UA do not necessarily include the ability to render (display) all of the characters received. If the message is forwarded, the UA must transmit exactly equivalent characters, but not necessarily from the same character set.

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 10. 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.

```
BodyPart                ::=  CHOICE {
  ia5-text                   [0] IA5TextBodyPart,
                             .
  oda-1984                   [12] IMPLICIT OCTET STRING,
  iso-6937                   [13] ISO6937BodyPart,
  bilaterally-defined        [14] Unidentified,
  externally-defined         [15] ExternallyDefinedBodyPart,
                             .
                             .
                             [310] IMPLICIT
                                  USAPrivatelyDefinedBodyParts,
                             .        }

Unidentified := OCTET STRING

The content of the ODA 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).

USAPrivatelyDefinedBodyParts 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.  OIW registers body part
numbers for privately-defined formats within the United
States.
```

Figure 10 - Privately-defined body parts

13.5 MS attributes

The IPM MS provides more flexible access to the general attributes (see clause A.8, table 43, enhanced column) as well as supporting IPM attributes (see clause A.10).

IPM UAs can make use of either the Basic MS or the IPM MS.

Clause A.10 is to be read in accordance with annex C or Recommendation X.420 (1988).

An IPM MS requires support from both the General Attributes and IPM Attributes as specified in clauses A.8 and A.10, respectively.

13.5.1 Implementation of the IPM 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:

a) The UA must generate 1984 P2 PDUs;

b) The UA must identify the content protocol as integer 2 to the MS;

c) The MS must be collocated 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:

a) The UA could conform to X.420 (1984), with 1988 UA extensions for utilizing the MS services;

b) 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 collocated are beyond the scope of these Agreements.

13.6 Body part conversion functional group

13.6.1 General

The Body Part Conversion Functional Group supports the functionality required to perform the action of message body part conversion. The Element of Service "Conversion Prohibition" is made mandatory in the MT Kernel.

13.6.2 Elements of service

The Body Part Conversion Functional Group provides support for the following Elements of Service.

Table 25 - Conversion: MT elements of service

Element of Service	Support
Conversion Prohibition in Case of Loss of Information (1988)	M
Explicit Conversion	M[1]
Implicit Conversion	O[1]
Notes 1 At least one of explicit or implicit conversion must be supported for conformance to this functional group.	

Operational Notes
Conversions to and from General Text can only be performed through implicit conversion. Among possible implicit conversions are the following:

a) Teletex to General Text;

b) IA5 Text to General Text;

c) General Text to Teletex;

d) General Text to IA5 Text.

13.6.3 Conformance

An implementation conforming to this functional group shall conform to the procedures for the Elements of Service in clause 13.6.2, and shall obey the rules defined in clauses 14.3.5 and 14.3.9 of X.411 / ISO/IEC 10021-4.

The PICS shall document which body part conversions the implementation can perform, both for implicit and explicit conversion, and whether "Conversion Prohibition in Case of Loss of Information" is supported. Conformance to this functional group does not mandate conversion between any two specific body part types.

If conversion has to take place and the Element of Service "Conversion Prohibition in Case of Loss of Information" is requested, then the MTA is not allowed to perform the conversion if loss of information may occur, according to the classification in clause 2.1 of X.408.

If the General Text body part type is supported, the implementation must support two-way conversion between the General Text IA5 subrepertoire and the IA5 Text body part.

If a UA is registered to receive multiple Encoded Information Types and its MTA receives a message for it containing any of those registered EITs, the corresponding body parts shall not be converted prior to delivery.

13.7 Security

There are no security requirements to support IPM, above and beyond those specified in clause 10.

13.8 Error handling

NOTE - For further study.

13.9 Physical delivery

Table 26 specifies the support for physical delivery elements of service as required by IPM.

Table 26 - Physical delivery: IPM elements of service

Element of Service	Origination (IPM UA)	Reception (PDAU)
Additional Physical Rendition	O[1]	O
Basic Physical Rendition	O[1]	M
Counter Collection	M	M
Counter Collection with Advice	O	O
Delivery via Bureaufax Service	O	O[2]
EMS (Express Mail Service)	M[1]	M[2]
Ordinary Mail	O[1]	M
Physical Delivery Notification by MHS	O	O
Physical Delivery Notification by PDS	O[1]	M
Physical Forwarding Allowed	O[1]	M
Physical Forwarding Prohibited	M	M
Registered Mail	O	O
Registered Mail to Addressee in Person	O	O
Request for Forwarding Address	O	O[2]
Special Delivery	M	M[2]
Undeliverable Mail with Return of Physical Message	O[1]	M

Notes
1 Provided by default (when using a physical delivery address).
2 Must support EMS and/or Special Delivery.

14 EDI messaging service

14.1 Introduction

This clause specifies the requirements for an EDI Messaging Service (EDIMS). These requirements are based on Recommendations X.435 and F.435 which define the P(edi) content type and outline various EDIMS operational scenarios.

This EDIMS Implementation Agreement separates the functions of the base standard into a Kernel and optional Functional Groups (FGs). These functional groups may be used to support the different scenarios of the EDIMS.

The following functional groups are defined:

- EDIMS Security
- EDIMS Forwarding
- EDIMS Multipart Body
- EDIMS Physical Delivery

These agreeements classify the support of these functional groups as follows:

Table 27 - EDIMS functional groups

Functional Group	Support
EDIMS Forwarding	O
EDIMS Security	O
EDIMS Multi Part Body	O
Notes	

14.2 EDIMS Elements of service

Tables 28.1 and 29 specify the requirements for support of EDIMS EoS by a UA conforming to the EDIMS functional group of this Agreement. The classification scheme for support of EoS is as defined in clause 5.2.

Table 28 - EDIMS: Basic EDI 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
EDI 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 collocated UA/MTA or collocated 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 29 - EDIMS: Optional EDI elements of service

	Kernel		Func. Group		
Element of Service	Orig	Rec	FG	Orig	Rec
Alternate Recipient Allowed	M	M			
Alternate Recipient Assignment	-	O			
Application Security Element	O	O[1]	SEC-C	M	M
Character Set	M	M			
Content Confidentiality	O	O	SEC-A,B	C[7]	C
Content Integrity[5]	O	O	SEC-A,B	C[7]	C
Conversion Prohibition	M	M			
Conversion Prohibition in Case of Information Loss (1988)	O	O			
Cross Reference Information	O	M	MPB	M	M
Deferred Delivery	M	-			
Deferred Delivery Cancellation	M	-			
Delivery Notification	M	-			
Designation of Recipient by Directory Name	O	-			
Disclosure of Other Recipients	M	M			
DL Expansion History Ind.(1988)	-	M			
DL Expansion Prohibited	M	-			
EDI Forwarding	O	-	FWD	M	-
EDI Message Type(s)	M	M			
EDI Notification Request	M	M			
EDI Standard Indication	M	M			
EDIM Responsibility Forwarding Allowed Indication	M	M			
EDIN Receiver	O	M	FWD	M	M
Expiry Date/Time Indication	O	M			
Explicit Conversion	O	-			
Grade of Delivery Selection	M	M			
Hold for Delivery	-	O[4]			
Implicit Conversion	-	O			
Incomplete Copy Indication	O	M	FWD	O[2]	M
Interchange Header	M	M			
Latest Delivery Designation	O	-			
Message Flow Confidentiality	O	-			
Message Origin Authentication[5]	O	O	SEC-A,B	C[7]	C
Message Security Labelling	O	O	SEC-A,B	C[7]	C
Message Sequence Integrity	O	O			
Multi-Destination Delivery	M	-			
Multi-Part Body	O	M	MPB	M	M
Non-repudiation of Content Originated	O	O	SEC-B	M	M
Non-repudiation of Content Received	O	O	SEC-B	M	M
Non-repudiation of Content Received Request	O	O	SEC-B	M	M
Non-repudiation of Delivery	O	O	SEC-A,B	C[7]	C
Non-repudiation of EDI Notification	O	O	SEC-B	M	M
Non-repudiation of EDI Notification Request	O	O	SEC-B	M	M

Table 29 - EDIMS: Optional EDI elements of service (concluded)

	Kernel		Func. Group		
Element of Service	Orig	Rec	FG	Orig	Rec
Non-repudiation of Origin[6]	O	O	SEC-A,B	C[7]	C
Non-repudiation of Submission	O	O			
Obsoleting Indication	O	M			
Originator Indication	M	M			
Originator Requested Alternate Recipient (1988)	O	-			
Prevention of Non Delivery Notification	O	-			
Probe	O	-			
Probe Origin Authentication	O	-			
Proof of Content Received	O	O	SEC-A,B	M	M
Proof of Content Received Request	O	O	SEC-A,B	M	M
Proof of Delivery	O	O			
Proof of EDI Notification	O	O	SEC-A,B	M	M
Proof of EDI Notification Request	O	O	SEC-A,B	M	M
Proof of Submission	O	-			
Recipient Indication	M	M			
Redirection Disallowed by Originator	O	-			
Redirection of Incoming Messages (1988)	-	O			
Related Message(s)	O	M			
Report Origin Authentication	O	O			
Requested Delivery Method	M	-			
Restricted Delivery (1988)	-	O			
Return of Content[3]	O	-			
Secure Access Management	O	O			
Services Indication	O	O			
Stored EDI Message Auto-forward	-	O			
Use of Distribution List (1988)	O	-			

Notes
1 This EOS requires a bilateral agreement.
2 Mandatory when an implementation supports the removal of body parts.
3 A defect report was submitted to CCITT/ISO by EWOS/ETSI, since the Return of Contents EoS was omitted from the list of EDIMS EoS in F.435.
4 Mandatory if P3 is supported.
5 SEC-A or SEC-B EoS may require the use of these services.
6 SEC-B EoS may require the use of this service.
7 Support of this EOS is dependent on the MHS Security Class implemented to support security class EDI-A (SEC-A) or EDI-B (SEC-B). See clause 10.

14.3 P(EDI) protocol

The requirements for EDI-UA support of the EDI protocol (Pedi) elements are defined in clause A.10.

14.4 EDIMS Multi-Part Body functional group

14.4.1 General

The EDIMS Multi-Part Body functional group defines the services and functionality required to support the generation of multiple body parts in an EDIM. Note that support on reception of Multi-Part Body is mandatory in the EDIMS Kernel.

14.4.2 Elements of service

The EDIMS Multi-Part Body functional group constitutes support of the following Elements of Service on origination:

- Cross Reference Information
- Multi-Part Body

14.5 EDI Message Store (EDI-MS)

14.5.1 MS Attributes

14.6 Conversion

14.7 EDIMS security functional group

The EDIMS Security functional group defines the services and functionality required to provide security for EDIMs and EDINs. These security features are specific to the EDIMS, and are described in X.435.

As the interface between the EDI Messaging (EDIMG) user and the EDI-UA is outside the scope of this document, implementations of the security mechanisms can be implemented as a discrete hardware/software component or within the EDI-UA.

NOTE - There are alternative methods of providing security to the EDIMG user. For example, the EDI-UA may just avail itself of the (content-type independent) security services provided or supported by the (1988) MHS and described in section 10 (e.g., content confidentiality, proof of delivery), without using the additional services of this functional group. Finally, security services may be provided within the EDI interchange itself, while

possibly using the EDI Application Security Element to convey some (bilaterally agreed) security arguments (e.g., key IDs) in the EDIM header.

The EDIMS Security functional group is specified as two security classes, denoted EDI-A and EDI-B. Note that the services provided below are provided, in some cases, by 1988 MHS security elements in the P1 (and P3) envelope. For example, depending on the security policy in force, the proof and non repudiation services below use the Content Integrity Check or Message Origin Authentication Check protocol elements.

See Section 10 of these Agreements for a description of the 1988 MHS Security functional group and classes. Annex A of these Agreements outlines support of the security protocol elements by the MTS.

The security classes EDI-A and EDI-B need the Message Origin Authentication and Content Integrity EoS. This shall be achieved either by supporting security class S0, or any other security class in clause 10, depending on the security policy in force.

NOTE - In order to counter the threat that a message could be stolen and its value credited to a third party, the use of content confidentiality is recommended. When using S0A, the base security EoS shall be used in the following way:

- the Content integrity check shall be generated from the clear content;
- the Content integrity check shall be carried in the message token;
- Content confidentiality shall be used. Encryption of the content prevents re-generation of the Content integrity check by a third party.

14.7.1 EDIMS security class EDI-A (SEC-A)

This class provides proof services; the recipient of an EDI information object can be assured that it was originated by the specified EDIMG user. Table 29 outlines support for the EoS contained in this class.

14.7.2 EDIMS security class EDI-B (SEC-B)

This class provides non repudiation services. These are "stronger" than the corresponding proof services in the sense that the recipient of an EDI information object can prove to a third party that the object was originated by the specified EDIMG user. Table 29 outlines support for the EoS contained in this class.

14.7.3 EDIMS security class EDI-C (SEC-C)

The security class EDI-C offers the following Element of Service:

- Application Security Element

This security class mandates that the above service is provided by an EDIMS end system.

14.8 EDIMS Physical Delivery functional group

14.9 EDIMS Forwarding functional group

14.9.1 General

The EDIMS Forwarding functional group defines the services and functionality required to perform forwarding of an EDI message by or on behalf of an EDIMG user.

An EDI-UA or EDI-MS claiming conformance to the EDI Forwarding functional group shall understand the semantics of the EDIMS abstract operations and service with regard to forwarding, EDI Notifications and EDIN reasons/diagnostic codes. The EDI-UA or EDI-MS shall generate appropriate EDI notifications when accepting, forwarding, or refusing responsibility for the EDI message. These notifications may be generated automatically by an EDI-MS or EDI-UA based on the presence or absence of an EDI-MS in the configuration. In addition, notifications may be generated as a result of a request by the EDIMG user. Please refer to Section 17.3.3 of X.435 for a full description of EDI Forwarding.

An EDI-UA that claims conformance to the EDIMS Forwarding functional group shall conform to clause A.12, Table 47, as regards protocol elements required by this functional group.

14.9.2 Elements of service

The EDIMS Forwarding functional group constitutes support of the following Elements of Service:

- EDI Forwarding
- EDIN Receiver

Conditional on the support of removal of body parts, the EDIMS Forwarding functional group offers the additional element of service:

- Incomplete Copy indication

14.10 Use of Directory

15 Use of underlying layers

15.1 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 clause 5.3. In X.410-1984 mode, the RTSE makes direct use of the services provided by the Session Layer, as specified in part 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 part 5 (Upper Layers) of these Agreements.

15.2 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 part 5 (Upper Layers) of these Agreements. It is recommended that RTSE be used for recovery purposes when the implementation does not use Transport Class 4 or there is a high probability of an association failure.

16 Error handling

This clause describes appropriate actions to be taken upon receipt of protocol elements which are not supported in these Implementation Agreements: 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.

16.1 PDU encoding

Various distinguished integer values will be defined in future versions of the X.400 standard that are not defined in the 1988 version. When an unknown distinguished value is encountered in an Integer or an enumerated type, it should, in general, not result in an error. The value should be treated transparently or ignored, wherever possible.

> **Editor's Note** - It is intended that this section will specify any special error handling required when unknown distinguished values are encountered.

16.2 Envelope

NOTE - For further study.

16.3 Reports

NOTE - For further study.

16.4 Pragmatic constraints

If an implementation detects a pragmatic constraint violation, then it may generate an appropriate error indication but is not required to do so.

17 Conformance

For this clause, 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 part 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 annex D in order to claim conformance to this Agreement.

Conformance requirements for support of Functional Groups by particular configuration types (see clause 1) are listed in table 30. An implementation may claim conformance to multiple configuration types (e.g., "MTA+UA" and "Class B MTA only").

Table 30 - Conformance requirements

Functional Group	Configuration[3]								
	MTA + UA[2]	MTA + MS	MTA Only[1]			MS + UA	MS Only	UA Only	
			A	B	C			P7	P3
MT Kernel	M[2]	M	M	M	M	-	-	-	-
Message Store[4]	-	M	-	-	-	M	M	M	-
Remote UA	-	-	-	M	-	-	-	-	M
Distribution List	O	O	O	O	O	*	-	*	*
Directory	O	O	O	O	O	O	O	O	O
MHS Management	*	*	*	*	*	*	*	*	*
Security	O	O	O	O	O	O	O	O	O
Redirection	*	*	*	*	*	*	*	*	*
Physical Delivery	*	*	*	*	*	*	*	*	*
Other Access Units	*	*	*	*	*	*	*	*	*
IPM Service[6]	O[5]	O	O	O	O	O[5]	O	O[5]	O[5]
EDI Service[6]	O[5]	O	O	O	O	O[5]	O	O[5]	O[5]

Notes

1 There are three conformance classes defined for the MT Kernel in clause 17.1.

2 Optional elements of a context-specific UA 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 in clause 17.2 for MS support.

5 At least one of the content-specific functional groups must be supported.

6 The content-specific functional groups may include requirements for levels of support by an MS and/or MTA (e.g., in terms of attributes supported, conversion requirements, etc.). In table 29, the support of a content-specific functional group by the MS only implies support of the MS requirements for that content type (i.e., attribute). Similarly, support in the MTA for a content-specific functional group only implies support for the MTA requirements for that content type (e.g., conversion).

17.1 MT Kernel Conformance Classes

The MT Kernel conformance classes are:

a) 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.

b) 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.

c) 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 11 illustrates several combinations of MT Kernel conformance classes. Additional combinations are possible.

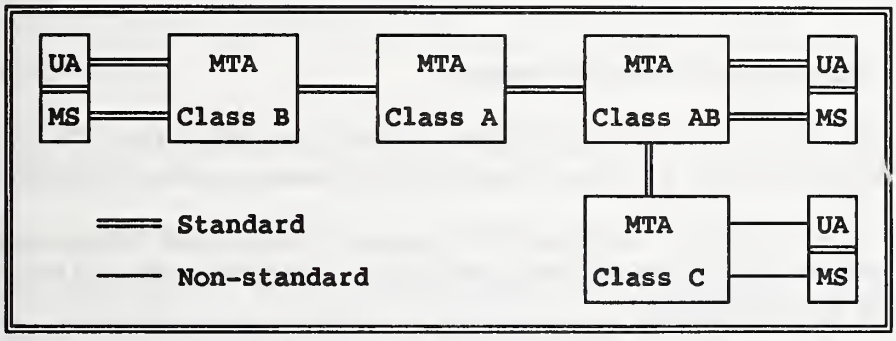


Figure 11 - MT kernel conformance classes

17.2 MS conformance levels

The MS conformance levels are:

a) A Basic MS only requires support for the General Attributes as specified in clause A.8, basic column of table 43;

b) An enhanced MS requires support for more of the General Attributes as specified in clause A.8 (enhanced column);

c) A Secure MS additionally requires support for the attributes as specified in clause A.9.

For content-specific MS requirements, see the appropriate content-specific clauses.

17.3 EDI-UA conformance

The EDI functional group requires the support of the EDIFACT and ANSI X12 EDI syntaxes.

18 Management domain agreements

The sections above describe agreements among implementors of particular X.400 components (e.g., MTAs, UAs, MSs). There are some agreements that don't apply to a single X.400 component, but instead apply to an entire domain of X.400 components. This section details any requirements for X.400 domains, independent of those for individual X.400 components. A single X.400 component cannot be conformance tested for these domain requirements, but for a domain to claim to be "operationally OIW compliant," it must abide by the rules stated below.

18.1 Management domain names

This section contains requirements on matters being considered by the U. S. CCITT Study Group D for national decisions. Such decisions are likely to supersede the relevant portions of this clause.

The implementation Agreements for 1984-based MHS implementations requires that all Management Domain Names (both Private and Administration) shall be unique within the United States. This is also a requirement for 1988-based MHS implementations.

A "Construction Syntax" is defined, which uses a registered OSI Organization Name from the ANSI US Register of Organization Names as a "root" in the construction of MHS Management Domain Names e.g., ADMD and PRMD). The constructed combinations based on this "root" will be guaranteed to be unique, and thus be safely used as MHS MD names in the United States. Other countries may wish to adopt these same rules.

MHS MD (PRMD and ADMD) names shall be constructed according to the Extended BNF grammar shown in figure 12.

```
<ADMDName> ::= <MDName>

<PRMDName> ::= <MDName>

<MDName> ::=
    <NationalOrganizationName> |
    <ConstructedName> |
    <NationalOrganizationNumber>

<ConstructedName> ::=
    <NationalOrganizationName>"+"<OrganizationallyDeterminedPart>
```

Figure 12 - Management domain name construction

Subject to all of the following rules:

Rule 1. The entire <MDName> must not exceed 16 bytes (including any constructor operators that may be included, and shall be composed entirely of PrintableString characters.

Rule 2. The <NationalOrganizationName> shall be drawn from the alphanumeric names registered in the US Register. It shall contain at least one non-numeric character, and not contain the constructor operator "+" (plus sign).

Rule 3. Each <NationalOrganizationName> obtained from the US Registry will be accompanied by a NumberForm (numeric value) which shall be bound as the <NationalOrganizationNumber> to the <NationalOrganizationName>.

Rule 4. In a <ConstructedName>, the <OrganizationallyDeterminedPart> shall be certified to be unique under the <NationalOrganizationName> (sub)authority, by the <NationalOrganizationName> registration authority.

Rule 5. A <NationalOrganizationNumber> shall be obtained from the US Register and bound to the <ConstructedName>.

Rule 6. A Private Management Domain's PrivateDomainIdentifier shall be the same as its PrivateDomainName.

NOTES

1 The PRMD names resulting from the <ConstructedName> syntax (those having a "+" in them) are atomic values from the point of view of the MTA -- in particular, it is not permissible for the MTA to route on components of the PRMD name.

2 The construction rules are such that if ABC is a Registered National Organization Name, then the owner of that name controls the MHS Domain Name space including "ABC" and "ABC+<anything>," but not "ABC<anything>."

3 A "+" is legal in an ANSI provided name.

4 If a Registered Organization Name already contains the construction operator ("+" sign), then in order to use the name as an <MDName>, its owner must also register the "root" which precedes the first "+" sign, with the US Register of Organization Names. (e.g., company B+Z+P would need to register "B" to be able to use the "constructed" name of B+Z+P.)

5 For the special case of the construction operator ("+" sign) being the first character of a Nationally Registered Name, no special action is required beyond its normal registration with the US Registry of Organization Names.

6 If the sub-authority determined by <NationalOrganizationName> so wishes, the <OrganizationallyDeterminedPart> can be constructed using rules similar to the above, resulting in a hierarchical construction separated by "+"s. In particular, the sub-authority must maintain its own registry and might (for example) define the <OrganizationallyDeterminedPart> using the syntax

```
<OrganizationallyDeterminedPart>  ::=  <DivisionName>
        |  <DivisionName> "+" <DivisionallyDeterminedPart>
```

Figure 13 - Name construction by subauthorities

where the <DivisionName> is drawn from the sub-authority's registry (and does not contain a "+"). Thus the sub-authority can delegate the use of the prefix

```
<NationalOrganizationName>+<DivisionName>
```

Figure 14 - Prefix

to someone else.

18.2 Use of ADMD names

This subsection was developed by an X.400 SIG working group in April, 1990. It contains extremely controversial positions that invoke national, commercial, and quality of service issues. The OIW may not be the correct forum to make these national decisions. Until these decisions can be reached or a national forum established, this section remains as a placeholder in the OIW X.400 SIG Working Text document only.

> **NOTE** - Version 2 of the CCITT X.400 Implementors Guide, dated 16 March 1990, allows for a single zero ("0") character as the ADMD name for the case of a PRMD that is not reachable from any ADMD. The following discussion does not apply to such PRMDs.

A PRMD may be directly connected to more than one ADMD. Since a PRMD may not alter the originators ORAddress, the Country/ADMD name pair provided in the Originator ORAddress may not match those of the first ADMD to receive the message from the PRMD. The first ADMD is required to accept such messages and may not alter the originator's ORAddress.

Any message originated by a PRMD must have an Originator's ORAddress that either uses the single space ADMD name or uses a Country/ADMD name pair for an ADMD to which the PRMD is connected. (In both cases the Country name is required.)

The X.400 Recommendations have defined a mechanism that enables PRMDs connected to multiple ADMDs to enter a single space as the ADMD name. To support this, these agreements recognize two classes of ADMDs. ADMDs in the first class, "space-supporting" ADMDs, must be able to route on PRMD name, independently from the ADMD name. Furthermore, the space-supporting ADMDs must arrange their routing configuration such that all PRMDs are reachable from all ADMDs. PRMDs using the single space ADMD name must be connected to at least one space-supporting ADMD.

ADMDs in the other class, "non-space-supporting" ADMDs, must, at a minimum, route messages for which the ADMD name is a single space to a space-supporting ADMD (in the indicated country). It is hoped that in the long term, all ADMDs will be able to route on the PRMD name when the ADMD name is a single space.

18.3 Uniqueness of MTS Identifiers within a management domain

When generating an IA5String in an MTS identifier, 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 an MD 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.

Annex A (normative)

MHS protocol specifications

Tables 31 through 48 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.**

Within the classification tables (32-48), the column "S" indicates the classification from the base standards. This is provided for reference purposes only and is intended to be in agreement with the base standards.

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 OIW 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 sub-elements 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 former classifications relate to the classifications in Part 7 of the Stable Agreements as shown in table 31.

Table 31 - 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 (32-48). The defined support for each is stated within each classification.

Implementations conforming to this Agreement must be capable of accepting the syntax of every protocol element of a protocol for which support is claimed. When an MS or MTA receives a protocol element that according to the base standard 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 PDU conveyed. Notwithstanding the above, criticality must be observed according to the base standard.

Mandatory (M) on Origination: Implementations conforming to this Agreement shall generate this element in all information objects in which, according to the base standards, it shall occur.

Mandatory (M) on Reception: Implementations conforming to this Agreement shall process this element appropriately according to its semantics.

Optional (O) on Origination: Where this element is not conveyed from one MHS entity to another, implementations conforming to this Agreement may optionally be capable of generating this protocol element, but are not required to do so.

Optional (O) on Reception: Implementations conforming to this Agreement may, but are not required to be capable of processing this protocol element.

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.

Not Applicable (-): The protocol element is not applicable in the particular context according to the base standard.

Where the dynamic behavior of protocol elements need to be specified, the following classification scheme is used:

Mandatory (m): The protocol element shall always be implemented and generated. On reception, correct action shall be taken as specified in the base standard, or as qualified or specified in these Agreements. Absence of the corresponding protocol element shall cause the appropriate abstract error to be generated.

Excluded (x): The protocol element shall not be present or it must be possible to prevent its use. Its presence shall cause the appropriate abstract error to be generated.

Dynamic conformance classifications are indicated in a single column of each of the Protocol Element tables. The classification applies to the usage only of the protocol elements which have a static classification.

There are two types of tables defining support for protocol elements: the first is a base table that contains and classifies all protocol elements, and the second is a delta table for a functional group.

Functional group tables need only list those protocol elements for which the functional group has changed the support requirements from the base table. Additional containing constructor elements may be listed in order to provide context information.

If the functional group changes the support requirements for a given element it must be classified in the delta table. Changes should only place more restrictions on the required support, for example changing an optional element to be either mandatory, excluded, or out of scope. If an element in the delta table is not classified, it is only listed for context information and the required support for it is the same as its classification in the base table.

The Dynamic column is only filled in if the profile changes the requirements for use of the element in every PDU, for example, if support for the element is required, but use of the element is optional in a given PDU.

However, if you are supporting a functional group that element will always (or never) be used.

A.1 MTS transfer protocol (P1)

Within Table 32, the columns under "Support by MT Kernel Class" refer to the MT Kernel Conformance Classes defined in clause 17.1.

Table 32 - Classification of the P1 protocol elements

MTS Transfer Protocol (P1)				Part 1 of 9
	Support by MT Kernel 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				
MessageTransfer	M	M/M	M/M	
ProbeTransfer	M	M/M	M/M	
ReportTransfer	M	M/M	M/M	See Note 7
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	
MTS-APDU				
message	M	M/M	O/M	
envelope	M	M/M	M/M	MessageTransferEnvelope
content	M	M/M	M/M	
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	

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 2 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	M/M	M/M	See Note 6
message-origin-authentication-check	O	O/O	O/O	
message-security-label	O	O/O	O/O	See Note 5
security-policy-identifier	O	M/M	M/M	
security-classification	O	M/M	M/M	
privacy-mark	O	O/O	O/O	
security-categories	O	M/M	M/M	
content-correlator	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-forwarding-prohibited	O	O/O	O/O	
physical-forwarding-address-request	O	O/O	O/O	
physical-delivery-modes	O	O/O	O/O	

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 3 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	
asymmetric-token	O	M/M	M/M	See Note 5
signature-algorithm-identifier	M	M/M	M/M	
name	M	M/M	M/M	
time	M	M/M	M/M	
sign-data	O	M/M	M/M	
content-confidentiality-algorithm-identifier	O	M/M	M/M	
content-integrity-check	O	M/M	M/M	
message-security-label	O	O/O	O/O	
proof-of-delivery-request	O	M/M	M/M	
message-sequence-number	O	O/O	O/O	
encryption-algorithm-identifier	O	M/M	M/M	
encrypted-data	O	M/M	M/M	
content-confidentiality-key	O	M/M	M/M	
content-integrity-check	O	M/M	M/M	
message-security-label	O	O/O	O/O	
content-integrity-key	O	O/O	O/O	
message-sequence-number	O	O/O	O/O	
content-integrity-check	O	M/M	M/M	See Note 6
proof-of-delivery-request	O	M/M	M/M	See Note 6
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	
per-domain-bilateral-information	O	O/O	O/O	PerDomainBilateralInfo

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 4 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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
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

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 5 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	
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	
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	

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 6 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	
value	M	M/M	M/M	
DLExpansionHistory				
DLExpansion	M	M/M	M/M	
ORAddressAndOptionalDirectoryName	M	M/M	M/M	ORName
dl-expansion-time	M	M/M	M/M	

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 7 of 9
Support by MT Kernel Class				Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	
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	
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	
ORAddressAndOptionalDirectoryName	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	
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	O/M	
given-name	O	M/M	O/M	
initials	O	M/M	O/M	See Note 4
generation-qualifier	O	M/M	O/M	
organizational-unit-names	O	M/M	O/M	

Table 32 - Classification of the P1 protocol elements (continued)

MTS Transfer Protocol (P1)				Part 8 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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	M/M	O/M	
teletex-personal-name	O	M/M	O/M	
teletex-organizational-unit-names	O	M/M	O/M	
teletex-domain-defined-attributes	O	M/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	
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

Table 32 - Classification of the P1 protocol elements (concluded)

MTS Transfer Protocol (P1)				Part 9 of 9
	Support by MT Kernel Class			Comments/References
Protocol Element	S	B/C O/R	A O/R	See Note 1
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

1 The MT Kernel implementation classes are defined in clause 17.1.

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.

3 An implementation is only required to generate EITs that correspond to the body parts it is capable of generating.

4 If the initials component of personal-name attribute is used, it should comprise all of the person's initials (including the given name) except the person's surname, as specified in X.402/IS 10021-2.

5 All S0 services may be provided without using the message token, e.g., using per-message extensions.

6 In secure messaging, use of elements within the message token is preferred to use of equivalent elements in the subject message envelope. A security policy shall define which other elements are dynamically mandated and shall define which message security labels are used for security context checking.

7 In the absence of any specific processing requirements for a particular element in the Message Transfer or Probe Transfer, the action to be taken is simply the creation of the corresponding element in the Report Transfer and is subject to constraints specified in X.411.

A.2 Interpersonal messaging protocol (P2)

Table 33 - 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	ORDescriptor
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	

Table 33 - Classification of the P2 protocol elements (continued)

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/M	
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

Table 33 - Classification of the P2 protocol elements (concluded)

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	See Note 10
parameters	M	M/M	
data	M	M/M	
GeneralTextBodyPart	O	O/M	See Note 9
ODA1984BodyPart	O	O/O	
ISO6937BodyPart	O	O/O	
BilaterallyDefinedBodyPart	O	O/O	
USAPrivatelyDefinedBodyPart	O	O/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.
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.
9 If General Text is supported, an implementation's PICS must identify which character repertoires can be generated on origination and supported on reception.
10 Any basic body part type that is supported on reception as an integer encoding must also be supported as an object identifier encoding. Support for all other externally defined body parts is optional.

Editor's Note - The draft text note regarding the meaning of "support" on reception was missing from the editing instructions.

A.3 MTS access protocol (P3)

NOTE - The support classifications for the 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 34 - Classification of the P3 protocol elements

MTS Access Protocol (P3)					Part 1 of 12
	Support by:				
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 See Note 10
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.

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 2 of 12
	Support by:				
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/-	
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	
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/-	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 3 of 12
	Support by:				
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/-	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/-	MTSIdentifier
probe-submission-time	M	-/M	-/M	M/-	
content-identifier	O	-/M	-/M	M/-	
CancelDeferredDelivery					
ARGUMENT					
message-submission-identifier	M	M/-	M/-	-/M	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	
waiting-messages	O	O/-	O/-	-/M	
waiting-content-types	O	O/-	O/-	-/M	
waiting-encoded-information-types	O	O/-	O/-	-/M	EncodedInformationTypes

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 4 of 12
	Support by:				
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	EncodedInformationTypes
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/-	EncodedInformationTypes
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	EncodedInformationTypes
deliverable-maximum-content-length	O	O/-	M/-	-/M	
default-delivery-controls	O	O/-	O/-	-/M	
restrict	O	O/-	O/-	-/M	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 5 of 12
	Support by:				
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	
permissible-encoded-information-types	O	O/-	O/-	-/M	EncodedInformationTypes
deliverable-content-types	O	O/-	M/-	-/M	
labels-and-redirections	O	O/-	O/-	-/O	
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/-	Note 8
simple	O	-/M	-/M	O/-	
strong	O	-/O	-/O	O/-	
new-credentials	M	-/M	-/M	M/-	Note 8
simple	O	-/M	-/M	O/-	
strong	O	-/O	-/O	O/-	
ChangeCredentials					MTSuser to MTS
ARGUMENT					
old-credentials	M	M/-	M/-	-/M	Note 8
simple	O	O/-	O/-	-/M	
strong	O	O/-	O/-	-/O	
new-credentials	M	M/-	M/-	-/M	Note 8
simple	O	O/-	O/-	-/M	
strong	O	O/-	O/-	-/O	
MessageSubmissionEnvelope					See Note 6
originator-name	M	M/-	M/-	-/M	ORName
original-encoded-information-types	O	M/-	M/-	-/M	EncodedInformationTypes
content-type	M	M/-	M/-	-/M	
built-in	O	O/-	M/-	-/M	
external	O	O/-	M/-	-/M	
content-identifier	O	O/-	M/-	-/M	
priority	O	M/-	M/-	-/M	All values
per-message-indicators	O	M/-	M/-	-/M	
disclosure-of-recipients	O	O/-	M/-	-/M	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 6 of 12
	Support by:				
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/-	O/-	-/M	MS Abstract Service only
PerRecipientMessageSubmissionFields	M	M/-	M/-	-/M	
recipient-name	M	M/-	M/-	-/M	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	Note 9
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	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 7 of 12
	Support by:				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
ProbeSubmissionEnvelope					See Note 6
originator-name	M	M/-	M/-	-/M	ORName
original-encoded-information-types	O	M/-	M/-	-/M	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	
content-length	O	M/-	M/-	-/M	
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	
PerRecipientProbeSubmission Fields	M	M/-	M/-	-/M	
recipient-name	M	M/-	M/-	-/M	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	See Note 9
physical-rendition-attributes	O	O/-	M/-	-/M	
MessageDeliveryEnvelope					See Note 7
message-delivery-identifier	M	-/M	-/M	M/-	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/-	
external	O	-/M	-/M	M/-	
originator-name	M	-/M	-/M	M/-	ORName

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 8 of 12
	Support by:				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
original-encoded-information-types	O	-/M	-/M	M/-	EncodedInformationTypes
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/-	ORName
this-recipient-name	M	-/M	-/M	M/-	ORName
originally-intended-recipient-name	O	-/M	-/M	M/-	ORName
converted-encoded-information-types	O	-/M	-/M	M/-	EncodedInformationTypes
message-submission-time	M	-/M	-/M	M/-	
content-identifier	O	-/M	-/M	M/-	
extensions	O	-/M	-/M	M/-	
conversion-with-loss-prohibited	O	-/M	-/M	M/-	
requested-delivery-method	O	-/M	-/M	M/-	See Note 9
physical-forwarding-prohibited	O	-/-	-/-	M/-	
physical-forwarding-address-request	O	-/-	-/-	M/-	
physical-delivery-modes	O	-/-	-/-	M/-	0-16
registered-mail-type	O	-/-	-/-	M/-	0-256
recipient-number-for-advice	O	-/-	-/-	M/-	1-32
physical-rendition-attributes	O	-/-	-/-	M/-	
physical-delivery-report-request	O	-/-	-/-	M/-	0-256
originator-return-address	O	-/-	-/-	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/-	
dl-expansion-history	O	-/M	-/M	M/-	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 9 of 12
	Support by:				
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/-	MTSIdentifier
content-identifier	O	-/M	-/M	M/-	
content-type	O	-/M	-/M	M/-	
built-in	O	-/M	-/M	M/-	
external	O	-/M	-/M	M/-	
original-encoded-information-types	O	-/M	-/M	M/-	EncodedInformationTypes
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/-	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/-	
actual-recipient-name	M	-/M	-/M	M/-	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/-	EncodedInformationTypes
originally-intended-recipient-name	O	-/M	-/M	M/-	ORName
supplementary-information	O	-/M	-/M	M/-	
extensions	O	-/M	-/M	M/-	
redirection-history	O	-/M	-/M	M/-	RedirectionHistory
physical-forwarding-address	O	-/M	-/M	M/-	
recipient-certificate	O	-/O	-/O	O/-	
proof-of-delivery	O	-/O	-/O	O/-	
ORName					MTS User to MTS
standard-attributes					
country-name	O	M/-	M/-	-/M	CountryName
administration-domain-name	O	M/-	M/-	-/M	DomainName
network-address	O	M/-	M/-	-/M	
terminal-identifier	O	M/-	M/-	-/M	

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 10 of 12
	Support by:				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
private-domain-name	O	M/-	M/-	-/M	DomainName
organization-name	O	M/-	M/-	-/M	
numeric-user-identifier	O	M/-	M/-	-/M	
personal-name	O	M/-	M/-	-/M	
surname	M	M/-	M/-	-/M	
given-name	O	M/-	M/-	-/M	
initials	O	M/-	M/-	-/M	
generation-qualifier	O	M/-	M/-	-/M	
organizational-unit-names	O	M/-	M/-	-/M	
OrganizationUnitName	M	M/-	M/-	-/M	
domain-defined-attributes	O	M/-	M/-	-/M	
DomainDefinedAttribute	M	M/-	M/-	-/M	
type	M	M/-	M/-	-/M	
value	M	M/-	M/-	-/M	
extension-attributes	O	M/-	M/-	-/M	ExtensionAttribute
common-name	O	M/-	M/-	-/M	
teletex-common-name	O	O/-	O/-	-/M	
teletex-organization-name	O	O/-	O/-	-/M	
teletex-personal-name	O	O/-	O/-	-/M	
teletex-organizational-unit-names	O	O/-	O/-	-/M	
teletex-domain-defined-attributes	O	O/-	O/-	-/M	
pds-name	O	O/-	O/-	-/M	
physical-delivery-country-name	O	O/-	O/-	-/M	
postal-code	O	O/-	O/-	-/M	
physical-delivery-office-name	O	O/-	O/-	-/M	
physical-delivery-office-number	O	O/-	O/-	-/M	
extension-OR-address-components	O	O/-	O/-	-/M	
physical-delivery-personal-name	O	O/-	O/-	-/M	
physical-delivery-organization-name	O	O/-	O/-	-/M	
extension-physical-delivery-address-components	O	O/-	O/-	-/M	
unformatted-postal-address	O	O/-	O/-	-/M	
street-address	O	O/-	O/-	-/M	
post-office-box-address	O	O/-	O/-	-/M	
poste-restante-address	O	O/-	O/-	-/M	
unique-postal-name	O	O/-	O/-	-/M	
local-postal-attributes	O	O/-	O/-	-/M	
extended-network-address	O	O/-	O/-	-/M	
terminal-type	O	O/-	O/-	-/M	
ORName					MTS to MTS User
standard-attributes					
country-name	O	-/M	-/M	M/-	CountryName

Table 34 - Classification of the P3 protocol elements (continued)

MTS Access Protocol (P3)					Part 11 of 12
	Support by:				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
administration-domain-name	O	-/M	-/M	M/-	DomainName
network-address	O	-/M	-/M	M/-	
terminal-identifier	O	-/M	-/M	M/-	
private-domain-name	O	-/M	-/M	M/-	DomainName
organization-name	O	-/M	-/M	M/-	
numeric-user-identifier	O	-/M	-/M	M/-	
personal-name	O	-/M	-/M	M/-	
surname	M	-/M	-/M	M/-	
given-name	O	-/M	-/M	M/-	
initials	O	-/M	-/M	M/-	
generation-qualifier	O	-/M	-/M	M/-	
organizational-unit-names	O	-/M	-/M	M/-	
OrganizationUnitName	M	-/M	-/M	M/-	
domain-defined-attributes	O	-/M	-/M	M/-	
DomainDefinedAttribute	M	-/M	-/M	M/-	
type	M	-/M	-/M	M/-	
value	M	-/M	-/M	M/-	
extension-attributes	O	-/M	-/M	M/-	ExtensionAttribute
common-name	O	-/M	-/M	M/-	
teletex-common-name	O	-/M	-/M	M/-	
teletex-organization-name	O	-/M	-/M	M/-	
teletex-personal-name	O	-/M	-/M	M/-	
teletex-organizational-unit-names	O	-/M	-/M	M/-	
teletex-domain-defined-attributes	O	-/M	-/M	M/-	
pds-name	O	-/O	-/M	M/-	
physical-delivery-country-name	O	-/O	-/M	M/-	
postal-code	O	-/O	-/M	M/-	
physical-delivery-office-name	O	-/O	-/M	M/-	
physical-delivery-office-number	O	-/O	-/M	M/-	
extension-OR-address-components	O	-/O	-/M	M/-	
physical-delivery-personal-name	O	-/O	-/M	M/-	
physical-delivery-organization-name	O	-/O	-/M	M/-	
extension-physical-delivery-address-components	O	-/O	-/M	M/-	
unformatted-postal-address	O	-/O	-/M	M/-	
street-address	O	-/O	-/M	M/-	
post-office-box-address	O	-/O	-/M	M/-	
poste-restante-address	O	-/O	-/M	M/-	
unique-postal-name	O	-/O	-/M	M/-	
local-postal-attributes	O	-/O	-/M	M/-	
extended-network-address	O	-/O	-/M	M/-	
terminal-type	O	-/O	-/M	M/-	

Table 34 - Classification of the P3 protocol elements (concluded)

MTS Access Protocol (P3)					Part 12 of 12
	Support by:				
Protocol Element	S	UA O/R	MS O/R	MTA O/R	Comments/References
EncodedInformationTypes					
built-in-encoded-information-types	M	M/M	M/M	M/M	See Note 3
non-basic-parameters	O	O/O	O/O	O/O	
external-encoded-information-types	O	O/M	O/M	M/O	
MTSIdentifier					
global-domain-identifier	M	M/M	M/M	M/M	GlobalDomainIdentifier
local-identifier	M	M/M	M/M	M/M	
OriginatorAndDLExpansionHistory					
originator-or-dl-name	M	M/M	M/M	M/M	
origination-or-expansion-time	M	M/M	M/M	M/M	
RedirectionHistory					
Redirection	M	M/M	M/M	M/M	
intended-recipient-name	M	M/M	M/M	M/M	
ORAddressAndOptionalDirectoryName	M	M/M	M/M	M/M	ORName
redirection-time	M	M/M	M/M	M/M	
redirection-reason	M	M/M	M/M	M/M	

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.41 (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 At least one of simple and/or strong must be specified.
9 Applies to ORNames containing Directory Names and/or ORAddresses See Recommendation X.411, section 8.2.1.1.1.14.
10 In the absence of any specific processing requirements for a particular element in the Message Submission, or Probe Submission, the action to be taken is simply the creation of the corresponding element in the ReportDelivery (subject to any constraints specified in X.411).
11 Applicable only to reception by a PDAU.

A.4 MS access protocol (P7)

Table 35 - Classification of the P7 protocol elements

MS Access Protocol (P7)				Part 1 of 6
	Support by:			
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	Opt'l in Basic MS(Note 5)
allowed-content-types	O	O/-	-/M	
allowed-EITs	O	O/-	-/M	
maximum-content-length	O	O/-	-/M	
MS-configuration-request	O	O/-	-/M	

Table 35 - Classification of the P7 protocol elements (continued)

MS Access Protocol (P7)				Part 2 of 6
	Support by:			
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/-	
available-attribute-types	O	-/M	M/-	
alert-indication	O	-/O	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	
RESULT				
SummarizeResult	M	-/M	M/-	
next	O	-/M	M/-	
count	M	-/M	M/-	
span	O	-/M	M/-	
lowest	M	-/M	M/-	
highest	M	-/M	M/-	
summaries	O	-/M	M/-	
absent	O	-/M	M/-	
present	O	-/M	M/-	
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	O/-	-/M	AttributeSelection
RESULT				
ListResult	M	-/M	M/-	
next	O	-/M	M/-	
requested	O	-/M	M/-	EntryInformation

Table 35 - Classification of the P7 protocol elements (continued)

MS Access Protocol (P7)				Part 3 of 6
	Support by:			
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	Optional in Basic MS
precise	O	M/-	-/M	
requested-attributes	O	O/-	-/M	AttributeSelection
RESULT				
FetchResult	M	-/M	M/-	
entry-information	O	-/M	M/-	EntryInformation
list	O	-/M	M/-	
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	Optional in Basic MS
sequence-numbers	O	M/-	-/M	
RESULT				
DeleteResult	M	-/M	M/-	
Register-MS				
ARGUMENT				
Register-MSArgument	M	M/-	-/M	
auto-action-registrations	O	O/-	-/O	
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	
type	M	M/-	-/M	
registration-identifier	O	M/-	-/M	
list-attribute-defaults	O	M/-	-/M	Optional in Basic MS
fetch-attribute-defaults	O	M/-	-/M	Optional in Basic MS
change-credentials	O	M/-	-/M	See Note 1
old-credentials	M	M/-	-/M	
new-credentials	M	M/-	-/M	
user-security-labels	O	O/-	-/O	
RESULT				
Register-MSResult	M	-/M	M/-	

Table 35 - Classification of the P7 protocol elements (continued)

MS Access Protocol (P7)				Part 4 of 6
	Support by:			
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 MessageSubmission -Envelope
deferred-delivery-time	O	O/-	-/M	
extensions	O	O/-	-/M	See P3 MessageSubmission -Envelope
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 MessageSubmission -Envelope
delete-after-auto-forwarding	O	O/-	-/M	
other-parameters	O	O/-	-/M	See Note 2
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

Table 35 - Classification of the P7 protocol elements (continued)

MS Access Protocol (P7)				Part 5 of 6
	Support by:			
Protocol Element	S	UA O/R	MS O/R	Comments/References
Common Data Types				
AttributeSelection				
type	M	M/-	-/M	
from	O	O/-	-/M	
count	O	O/-	-/M	
AttributeValueAssertion				
type	M	M/-	-/M	
value	M	M/-	-/M	
EntryInformation				
sequence-number	M	-/M	M/-	
attributes	O	-/M	M/-	
type	M	-/M	M/-	
values	M	-/M	M/-	
Filter				
item	O	M/-	-/M	FilterItem
and	O	M/-	-/M	See Note 3
or	O	M/-	-/M	See Note 3
not	O	M/-	-/M	See Note 4
FilterItem				
equality	O	M/-	-/M	AttributeValueAssertion (Support is Optional if ORName)
substrings	O	O/-	-/O	
type	M	M/-	-/M	
strings	M	M/-	-/M	
initial	O	O/-	-/M	
any	O	O/-	-/M	
final	O	O/-	-/M	
greater-or-equal	O	O/-	-/M	AttributeValueAssertion
less-or-equal	O	O/-	-/M	AttributeValueAssertion
present	O	O/-	-/M	
approximate-match	O	O/-	-/O	
InformationBase				
stored-messages	O	M/-	-/M	
inlog	O	O/-	-/O	
outlog	O	O/-	-/O	

Table 35 - Classification of the P7 protocol elements (concluded)

MS Access Protocol (P7)				Part 6 of 6
	Support by:			
Protocol Element	S	UA O/R	MS O/R	Comments/References
Range				See Note 6
sequence-number-range	O	O/-	-/M	
from	O	O/-	-/M	
to	O	O/-	-/M	
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	M/-	-/M	
override	O	O/-	-/M	Opt'l in Basic MS-Note 5

Notes

1 At least one of simple and/or strong must be specified.
2 The specified syntax of other-parameters is context-specific - see X.413 section 12.1.
3 For recursive use of filter, only support of the "item" and the "not" fields is required; there is only one level of recursion.
4 For recursive use of filter, only support of the "item" field is required; there is only one level of recursion.
5 If one of fetch-restrictions of MSBind and override of Selector is implemented, the other must also be implemented.
6 At least one of From or To must be implemented.

A.5 Classification of the P1 protocol elements for security classes

The protocol element classifications used in tables 36 and 37 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 33. Thus, table 36 shows the additional support required in P1 to conform to security class S1. Table 37 indicates the additional support required to support security class S2 (above and beyond that for security class S1).

NOTES

1 There are no additional classifications for security class S0.

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

Table 36 - Conformance classification of the P1 protocol elements for security class S1

MTS Transfer Protocol (P1) for Security Class S1				Part 1 of 2
	MT Kernel Static Support by MTS Class			
Protocol Element	B/C O/R	A O/R	Dyn	Comments/References
MTABind				
ARGUMENT				
<SET>				
initiator-credentials			M	
simple	O/O	O/O	X	
strong	M/M	M/M	M	
bind-token	M/M	M/M	M	
certificate	O/O	O/O		
security-context	M/M	M/M	M	
RESULT				
<SET>				
responder-credentials			M	
simple	O/O	O/O	X	
strong	M/M	M/M	M	
bind-token	M/M	M/M	M	
certificate	O/O	O/O		
MessageTransferEnvelope				
extensions				
message-security-label	M/M	M/M	M	

Table 36 - Conformance classification of the P1 protocol elements for security class S1 (concluded)

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

Notes

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

2 The message security label may appear in any or all of the indicated locations in the envelope. However the Security context service applies only to the label in the "extensions" and/or token signed-data as defined by the security policy in force. Labels in th token encrypted data have only end-to-end (UA-to-UA) significance.

Table 37 - Conformance classification of the P1 protocol elements for security class S2

MTS Transfer Protocol (P1) for Security Class S2				Part 1 of 2
	MT Kernel Static Support by MTS Class			
Protocol Element	B/C O/R	A O/R	Dyn	Comments/References
MessageTransferEnvelope				
extension				
originator-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
message-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
ProbeTransferEnvelope				
extensions				
originator-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
probe-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
ReportTransferEnvelope				
extensions				
reporting-MTA-certificate	M/M	M/M		
certificate	M/M	M/M		
certification-path	M/M	M/M		
report-origin-authentication-check	M/M	M/M	M	
algorithm-identifier	M/M	M/M		
content-identifier	M/M	M/M		
message-security-label	M/M	M/M		
per-recipient	M/M	M/M		
actual-recipient-name	M/M	M/M		
originally-intended-recipient-name	O/O	O/O		
delivery	O/O	O/O		
message-delivery-time	M/M	M/M		
type-of-MTS-user	M/M	M/M		
recipient-certificate	M/M	M/M		
proof-of-delivery	M/M	M/M		
non-delivery	O/O	O/O		
non-delivery-reason-code	M/M	M/M		
non-delivery-diagnostic-code	O/O	O/O		

Table 37 - Conformance classification of the P1 protocol elements for security class S2 (concluded)

MTS Transfer Protocol (P1) for Security Class S2				Part 2 of 2
	MT Kernel Static Support by MTS Class			
Protocol Element	B/C O/R	A O/R	Dyn	Comments/References
Certificate				
version	M/M	M/M		
serialNumber	M/M	M/M		
signature	M/M	M/M		
algorithm	M/M	M/M		
parameters	O/O	O/O		
issuer	M/M	M/M		
validity	M/M	M/M		
notBefore	M/M	M/M		
notAfter	M/M	M/M		
subject	M/M	M/M		
subjectPublicKeyInfo	M/M	M/M		
algorithm	M/M	M/M		
subjectPublicKey	M/M	M/M		

A.6 Classification of the P3 protocol elements for security classes

The protocol element classifications in tables 38, 39, and 40 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 34. Thus, table 38 shows the additional support required in P3 to conform to security class S0. Table 39 indicates the additional support required to support security class S1 (above and beyond that for security class S0). Table 40 indicates the additional support required to support security class S2 (above and beyond that for security class S1).

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

Table 38 - Conformance classification of the P3 protocol elements for security class S0

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

Table 38 - Conformance classification of the P3 protocol elements for security class S0 (concluded)

MTS Access Protocol (P3) for Security Class S0					Part 2 of 2
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
MessageDeliveryEnvelope					
other-fields					
extensions					
message-token	-/M	-/M	O/-		
asymmetric-token	-/M	-/M	O/-		
signature-algorithm-identifier	-/M	-/M	O/-		
name	-/M	-/M	O/-		
time	-/M	-/M	O/-		
signed-data	-/M	-/M	O/-		
content-confidentiality-algorithm-identifier	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		See Note 1
message-security-label	-/O	-/O	O/-		
proof-of-delivery-request	-/M	-/M	O/-		See Note 1
message-sequence-number	-/O	-/O	O/-		
encryption-algorithm-identifier	-/O	-/O	O/-		
encrypted-data	-/M	-/M	O/-		
content-confidentiality-key	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		See Note 1
message-security-label	-/O	-/O	O/-		
content-integrity-key	-/O	-/O	O/-		
message-sequence-number	-/O	-/O	O/-		
content-integrity-check	-/M	-/M	O/-		See Note 1
proof-of-delivery-request	-/M	-/M	O/-		See Note 1
ReportDeliveryEnvelope					
PerRecipientReportDelivery-Fields					
extensions					
proof-of-delivery	-/M	-/O	O/-		

Notes
1 Implementations shall generate no more that one instance of these identically-named protocol elements in a single message.

Table 39 - Conformance classification of the P3 protocol elements for security class S1

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

Table 39 - Conformance classification of the P3 protocol elements for security class S1 (continued)

MTS Access Protocol (P3) for Security Class S1					Part 2 of 3
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
ChangeCredentials					MTS to MTSuser
ARGUMENT					
old-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
new-credentials				M	
simple	-/O	-/O	O/-	X	
strong	-/M	-/M	M/-	M	
bind-token	-/M	-/M	M/-	M	
certificate	-/O	-/O	O/-		
ChangeCredentials					MTSuser to MTS
ARGUMENT					
old-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
new-credentials				M	
simple	O/-	O/-	-/O	X	
strong	M/-	M/-	-/M	M	
bind-token	M/-	M/-	-/M	M	
certificate	O/-	O/-	-/O		
MessageSubmissionEnvelope					
extensions					
message-token	M/-	M/-	-/M		
signed-data					
message-security-label	M/-	M/-	-/M		See Note 1
security-policy-identifier	M/-	M/-	-/M	M	
encrypted-data					
message-security-label	O/-	O/-	-/O		
content-integrity-check	M/-	M/-	-/M	M	
message-security-label	M/-	M/-	-/M		See Note 1
security-policy-identifier	M/-	M/-	-/M	M	
MessageDeliveryEnvelope					
extensions					
message-security-label	-/M	-/M	M/-		See Note 1
security-policy-identifier	-/M	-/M	M/-	M	
message-token	-/M	-/M	M/-		
signed-data					
message-security-label	-/O	-/O	O/-		See Note 1
encrypted-data					
message-security-label	-/O	-/O	O/-		See Note 1

Table 39 - Conformance classification of the P3 protocol elements for security class S1 (concluded)

MTS Access Protocol (P3) for Security Class S1					Part 3 of 3
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
ReportDeliveryEnvelope					
extensions					
message-security-label	-/M	-/M	M/-	M	See Note 1
bind-token					
asymmetric-token					
signature-algorithm-identifier	-/M	-/M	M/-	M	
name	-/M	-/M	M/-	M	
time	-/M	-/M	M/-	M	
signed-data	-/M	-/M	M/-	M	
encryption-algorithm-identifier	-/M	-/M	M/-		
encrypted-data	-/M	-/M	M/-		
message-security-label	-/M	-/M	M/-		
content-integrity-key	-/M	-/M	M/-		

Notes
1 The message-security-label may appear in any or all of the indicated locations in the envelope. However, the security labelling context services apply only to the label in the "extensions" field. Labels in th message token have only end-to-end (UA-to-UA) significance.

Table 40 - Conformance classification of the P3 protocol elements for security class S2

MTS Access Protocol (P3) for Security Class S2					Part 1 of 2
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
MessageSubmission					
RESULT					
extensions					
originating-MTA-certificate	-/M	-/O	M/-		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
proof-of-submission	-/M	-/O	M/-		
MessageDelivery					
RESULT					
recipient-certificate	M/-	M/-	-/O		
certificate	M/-	M/-	-/M		
certification-path	M/-	M/-	-/M		
MessageSubmissionEnvelope					
extensions					
originator-certificate	M/-	O/-	-/M		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
message-origin-authentication-check	M/-	O/-	-/M	M	
algorithm-identifier	M/-	M/-	-/M		
content	M/-	M/-	-/M		
content-identifier	M/-	M/-	-/M		
message-security-label	M/-	M/-	-/M		
proof-of-submission-request	M/-	O/-	-/M		
ProbeSubmissionEnvelope					
extensions					
originator-certificate	M/-	O/-	-/M		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
probe-origin-authentication-check	M/-	O/-	-/M	M	
algorithm-identifier	M/-	M/-	-/M		
content-identifier	M/-	M/-	-/M		
message-security-label	M/-	M/-	-/M		

Table 40 - Conformance classification of the P3 protocol elements for security class S2 (concluded)

MTS Access Protocol (P3) for Security Class S2					Part 2 of 2
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
MessageDeliveryEnvelope					
extensions					
originator-certificate	-/M	-/M	M/-		
certificate	-/M	-/M	M/-		
certification-path	-/M	-/M	M/-		
message-origin-authentication-check	-/M	-/M	M/-	M	
algorithm-identifier	-/M	-/M	M/-		
content	-/M	-/M	M/-		
content-identifier	-/M	-/M	M/-		
message-security-label	-/M	-/M	M/-		
ReportDeliveryEnvelope					
extensions					
reporting-MTA-certificate	-/M	-/O	M/-		
certificate	-/-	-/O	-/-		
certification-path	-/-	-/O	-/-		
report-origin-authentication-check	-/M	-/O	M/-	M	
PerRecipientReportDelivery-Fields					
extensions					
recipient-certificate	-/M	-/M	O/-		
certificate	-/M	-/M	M/-		
certification-path	-/M	-/M	M/-		
Certificate					
version	-/M	-/M	M/-		
serialNumber	-/M	-/M	M/-		
signature	-/M	-/M	M/-		
algorithm	-/M	-/M	M/-		
parameters	-/O	-/O	O/-		
issuer	-/M	-/M	M/-		
validity	-/M	-/M	M/-		
notBefore	-/M	-/M	M/-		
notAfter	-/M	-/M	M/-		
subject	-/M	-/M	M/-		
subjectPublicKeyInfo	-/M	-/M	M/-		
algorithm	-/M	-/M	M/-		
subjectPublicKey	-/M	-/M	M/-		

able 41 presents the classification delta to classification tables 38, 39, and 40, for the addition of mandatory ontent confidentiality in the static conformance classification.

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

Table 41 - Conformance classification of the P3 protocol elements for security classes S0a, S1a, or S2a

MTS Access Protocol (P3) for Security Classes S0a, S1a, S2a					Part 1 of 1
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
MessageSubmissionEnvelope					
extensions					
content-confidentiality-algorithm-identifier	M/-	O/-	-/O		See Note 1
message-token					
asymmetric-token					
signed-data	M/-	-/-	-/-		
content-confidentiality-algorithm-identifier	M/-	-/-	-/-		See Note 1
encrypted-data					
content-confidentiality-key	M/-	-/-	-/-		
MessageDeliveryEnvelope					
extensions					
message-token	-/M	-/M	O/-		
asymmetric-token					
signed-data	-/M	-/M	-/-		
content-confidentiality-algorithm-identifier	-/M	-/M	-/-		See Note 1
encrypted-data					
content-confidentiality-key	-/M	-/M	-/-		
content-confidentiality-algorithm-identifier	-/M	-/M	O/-		See Note 1

Notes
1 Implementors shall generate no more than one instance of these identically named protocol elements in a single message.

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

The protocol element classifications in table 42 should be viewed as a delta to the lower security class or, if there is no lower security class, to the kernel as classified in table 35. Thus, table 42 shows the additional support required in P7 to conform to security class S1.

NOTES

1 There are no additional classifications for security classes S0 and S2.

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

Table 42 - Conformance classification of the P7 protocol elements for security class S1

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

A.8 Message store general attribute support

Table 43 specifies the classification of the Message Store General Attributes.

Table 43 - Classification of the message store general attributes

Message Store General Attribute Support					Part 1 of 2
	Support by:		Bas	Enchanced	
Attribute	S	UA R	MS O	MS O	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	M	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	M	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	M	M	
proof-of-delivery-request	O	O	O	O	
redirection-history	O	O	O	M	
report-delivery-envelope	M	M	M	M	
reporting-dl-name	O	O	O	M	
reporting-mta-certificate	O	O	O	O	

Table 43 - Classification of the message store general attributes (concluded)

Message Store General Attribute Support					Part 2 of 2
Support by:			Bas MS	Enhanced MS	
Attribute	S	UA R	O	O	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.

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

The classification of the attributes in table 44 is a delta to the Enhanced MS column of the MS General Attributes in table 43. Table 44 indicates the additional attributes that must be supported in the MS for each of the security classes. There is no support required for security attributes in the basic MS.

Table 44 - MS security attribute support

Attribute	Security Class					
	S0	S0a	S1	S1a	S2	S2a
content-confidentiality-algorithm-identifier	O	M	O	M	O	M
content-integrity-check	M	M	M	M	M	M
message-security-label	O	O	M	M	M	M
message-origin-authentication-check	M	M	M	M	M	M
message-token	M	M	M	M	M	M
origination-certificate	O	O	O	O	M	M
proof-of-delivery	M	M	M	M	M	M
reporting-mta-certificate	O	O	O	O	M	M
report-origin-authentication-check	O	O	O	O	M	M
security-classification	O	O	M	M	M	M

A.10 Message store IPM attribute support

Table 45 specifies the classification of the Message Store IPM attributes. This clause is to be read in accordance with Annex C of X.420 (1988). For support of MS General Attributes, see table 43, enhanced MS column.

Table 45 - Classification of the message store IPM attributes

Message Store IPM Attribute Support				Part 1 of 2
Support by:		IPM UA	IPM MS	
Attribute	S	R	O	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	

Table 45 - Classification of the message store IPM attributes (concluded)

Message Store IPM Attribute Support				Part 2 of 2
	Support by:	IPM UA	IPM MS	
Attribute	S	R	O	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	
oda-1984-body-parts	-	O	O	
iso6937-body-parts	-	O	O	
bilaterally-defined-body-parts	-	O	O	
usa-privately-defined-body-parts	-	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	

A.11 EDI messaging service protocol (Pedi)

Table 46 - Classification of the Pedi protocol elements

EDI Messaging Service Protocol (Pedi)					Part 1 of 6
	Support by EDI UA				
Protocol Element	S	O/R	FGs	O/R	Comments/References
InformationObject					
edim	M	M/M			
edin	M	M/M			
EDIMIdentifier					
user	M	M/M			
user-relative-identifier	M	M/M			
ExtensionField					
type	M	M/M			
criticality	M	M/M			
value	M	M/M			
EDIM					
heading	M	M/M			
body	M	M/M			
Heading					
this-EDIM	M	M/M			
originator	O	M/M			
recipients	O	M/M			
edin-receiver	O	O/M	FWD	M/M	
responsibility-forwarded	O	O/M	FWD	M/M	
edi-bodypart-type	O	M/M			
incomplete-copy	O	O/M	FWD	O/M	See Note 2
expiry-time	O	O/M			
related-messages	O	O/M			
obsoleted-EDIMs	O	O/M			
edi-application-security-elements	O	O/O	SEC-C	M/M	
cross-referencing-information	O	O/M	MBP	M/M	
edi-message-type	O	M/M			
service-string-advice	O	M/M			
syntax-identifier	O	M/M			
interchange-sender	O	M/M			
date-and-time-of-preparation	O	M/M			
application-reference	O	M/M			
heading-extensions	O	O/M			See Note 3

Table 46 - Classification of the Pedi protocol elements (continued)

EDI Messaging Service Protocol (Pedi)					Part 2 of 6
	Support by EDI UA				
Protocol Element	S	O/R	FGs	O/R	Comments/References
RecipientSubfield					
recipient	M	M/M			
action-request	O	O/M			
edi-notification-requests-field	O	M/M			
responsibility-passing-allowed	O	M/M			
interchange-recipient	O	M/M			
recipient-reference	O	M/M			
interchange-control-reference	O	M/M			
processing-priority-code	O	M/M			
acknowledgement-request	O	M/M			
communications-agreement-id	O	M/M			
test-indicator	O	M/M			
authorization-information	O	M/M			
recipient-extensions	O	O/M			See Note 3
EDINotificationRequestsFields					
edi-notification-requests	O	M/M			
edi-notification-security	O	O/O	SEC-A	M/M	
			SEC-B	M/M	
edi-reception-security	O	O/O	SEC-A	M/M	
			SEC-B	M/M	
InterchangeRecipientField					
recipient-identification	M	M/M			
identification-code-qualifier	O	M/M			
routing-address	O	M/M			
RecipientReferenceField					
recipient-reference	M	M/M			
recipient-reference-qualifier	O	M/M			
EDINReceiverField					
edin-receiver-name	M	M/M			
original-edim-identifier	O	O/M	FWD	M/M	
first-recipient	O	O/M	FWD	M/M	
RelatedMessagesField					
RelatedMessageReference	M	M/M			
edi-message-reference	O	M/M			
external-message-reference	O	M/M			
EDIApplicationSecurityElements-Field					
edi-application-security-element	O	M/M			
edi-encrypted-primary-bodypart	O	M/M			
edi-application-security-extensions	O	O/M			See Note 3

Table 46 - Classification of the Pedi protocol elements (continued)

EDI Messaging Service Protocol (Pedi)					Part 3 of 6
	Support by EDI UA				
Protocol Element	S	O/R	FGs	O/R	Comments/References
CrossReferencingInformation-Subfield					
application-cross-reference	M	M/M			
message-reference	O	M/M			
body-part-reference	M	M/M			
ServiceStringAdviceField					
component-data-element-separator	M	M/M			
data-element-separator	M	M/M			
decimal-notation	M	M/M			
release-indicator	O	M/M			
reserved	O	M/M			
segment-terminator	M	M/M			
SyntaxIdentifierField					
syntax-identifier	M	M/M			
syntax-version	M	M/M			
InterchangeSenderField					
sender-identification	M	M/M			
identification-code-qualifier	O	M/M			
address-for-reverse-routing	O	M/M			
AuthorizationInformationField					
authorization-information	M	M/M			
authorization-information-qualifier	O	M/M			
Body					
primary-body-part	M	M/M			
additional-body-parts	O	O/M	MBP	M/M	
PrimaryBodyPart					
edi-body-part	O	M/M			
forwarded-EDIM	O	O/M	FWD	M/M	
EDIMBodyPart					
parameters	O	O/M	FWD	M/M	
message-data	M	M/M			
MessageParameters					
delivery-time	O	O/M	FWD	M/M	See Note 1
delivery-envelope	O	O/M	FWD	M/M	See Note 1

Table 46 - Classification of the Pedi protocol elements (continued)

EDI Messaging Service Protocol (Pedi)					Part 4 of 6
	Support by EDI UA				
Protocol Element	S	O/R	FGs	O/R	Comments/References
other-parameters	O	O/O			See Note 4
MessageData					
heading	M	M/M			
body	M	M/M			
BodyOrRemoved					
primary-or-removed	M	M/M			
additional-body-parts	O	M/M			
PrimaryOrRemoved					
removed-edi-body	O	O/M			See Note 5
primary-body-part	O	M/M			
AdditionalBodyParts					
external-body-part	O	M/M			
place-holder	O	O/M			See Note 5
EDIM-ExternallyDefinedBodyPart					
body-part-reference	O	M/M			
external-body-part	M	M/M			
EDIN					
positive-notification	O	M/M			
negative-notification	O	M/M			
forwarded-notification	O	O/M	FWD	M/M	
CommonFields					
subject-edim	M	M/M			
edin-originator	M	M/M			
first-recipient	O	M/M			
notification-time	M	M/M			
notification-security-elements	O	O/O	SEC-A	M/M	See Note 8
			SEC-B	M/M	See Note 8
			SEC-C	M/M	See Note 8
edin-initiator	M	M/M			
notifications-extensions	O	O/M			See Note 3
SecurityElementField					
original-content	O	O/O	SEC-A	M/M	See Note 6
			SEC-B	M/M	
original-content-integrity-check	O	O/O	SEC-A	M/M	See Note 6
			SEC-B	M/M	
edi-application-security-elements	O	O/O	SEC-C	M/M	
security-extensions	O	O/M			See Note 3

Table 46 - Classification of the Pedi protocol elements (continued)

EDI Messaging Service Protocol (Pedi)					Part 5 of 6
	Support by EDI UA				
Protocol Element	S	O/R	FGs	O/R	Comments/References
PositiveNotificationFields					
pn-common-fields	M	M/M			
pn-supplementary-information	O	O/M			
pn-extensions	O	O/M			See Note 3
NegativeNotificationFields					
nn-common-fields	M	M/M			
nn-reason-code	M	M/M			
nn-supplementary-information	O	M/M			
nn-extensions	O	O/M			See Note 3
NNReasonCodeField					
nn-ua-ms-reason-code	O	M/M			
nn-user-reason-code	O	M/M			
nn-pdau-reason-code	O	O/M			
NNUAMSReasonCodeField					
nn-ua-ms-basic-code	M	M/M			
nn-ua-ms-diagnostic	O	M/M			
NNUserReasonCodeField					
nn-user-basic-code	M	M/M			
nn-user-diagnostic	O	M/M			
NNPDAUReasonCodeField					
nn-pdau-basic-code	M	M/M			
nn-pdau-diagnostic	O	M/M			
ForwardNotificationFields					
fn-common-fields	M	M/M			
forwarded-to	M	M/M			
fn-reason-code	M	M/M			
fn-supplementary-information	O	O/M	FWD	M/M	
fn-extensions	O	O/M			See Note 3
FNReasonCodeField					
fn-ua-ms-reason-code	M	O/M			See Note 7
fn-user-reason-code	O	O/M			See Note 7
fn-pdau-reason-code	O	O/M			
FNUAMSReasonCodeField					
fn-ua-ms-basic-code	M	M/M			
fn-ua-ms-diagnostic	O	M/M			
fn-security-check	O	O/O	SEC-A	M/M	
			SEC-B	M/M	

Table 46 - Classification of the Pedi protocol elements (concluded)

EDI Messaging Service Protocol (Pedi)						Part 6 of 6
	Support by EDI UA					
Protocol Element	S	O/R	FGs	O/R	Comments/References	
FNUserReasonCodeField						
fn-user-basic-code	M	M/M				
fn-user-diagnostic	O	M/M				
FNPDAUReasonCodeField						
fn-pdau-basic-code	M	M/M				
fn-pdau-diagnostic	O	M/M				

Notes
1 M on origination if the implementation supports forwarding of a multi part EDIM without accepting responsibility.
2 Mandatory (on origination) when an implementation supports the removal of body parts.
3 Critical extensions must be supported in order to accept responsibility.
4 Use of supplementary information fields requires bilateral agreement.
5 Mandatory on origination if removal of body parts is supported.
6 One of these two elements must be supported on origination when using the SEC-A or SEC-B EDI security class.
7 One of these two elements must be supported on origination.
8 M on origination if EDI-notification-security or EDI-reception-security (of the EDINotificationRequestsFields) are supported on reception.

A.12 Message store EDIMS attribute support

A.13 Classification of the P3 protocol elements for physical delivery

The protocol elements used in Table 48 should be viewed as a delta to the kernel as classified in Table 34. Thus, Table 48 shows the additional supported required in P3 to conform to the Physical Delivery functional group.

Table 48 - Classification of the P3 protocol elements for physical delivery

MTS Access Protocol (P3) for Physical Delivery					Part 1 of 1
Protocol Element	Static Support by: UA O/R	MS O/R	MTA O/R	Dyn	Comments/References
MessageSubmissionEnvelope					
extensions					
originator-return-address	M/-	M/-	-/M		
PerRecipientMessageSubmission Fields					
extensions					
physical-forwarding-prohibited	M/-	M/-	-/M		
certification-path	M/-	M/-	-/M		
ORName					
extension-attributes					
physical-delivery-country-name	M/-	M/-	-/M		
postal-code	M/-	M/-	-/M		
unformatted-postal-code	M/-	M/-	-/M		

Annex B (normative)

Object identifiers

B.1 X.400 SIG object identifiers

The X.400 SIG object identifiers all allocated under the *mhsig* node in the OIW object identifier subtree, as defined in part 6 of the Stable Implementors Agreements document. This definition is duplicated in figure 15.

```
id-mhsig  OBJECT IDENTIFIER  ::=
            { iso (1)  identified-organization (3)  oiw (14)  mhsig (6) }
```

Figure 15 - Definition of the *mhsig* object identifier

The X.400 SIG has defined several categories of object identifiers. Their definition is provided in figure 16.

```
id-mhsig-content-types     OBJECT IDENTIFIER  ::=
                             { id-mhsig  content-types (0)  }

id-mhsig-body-part-types   OBJECT IDENTIFIER  ::=
                             { id-mhsig  body-part-types (1)  }
```

Figure 16 - Defintion of the X.400 SIG Object Identifier Categories.

B.2 Content types

There are presently no object identifiers for content types allocated by the X.400 SIG.

B.3 Body part types

The object identifiers for the external body part types allocated by the X.400 SIG are defined in figure 17.

```
id-privacy-enhanced-mail  OBJECT IDENTIFIER  ::=
                            { id-mhsig-body-part-types  pem (0) }
```

Figure 17 - Definition of the External body part object identifiers

B.4 Security classes

The ASN.1 expressed in figure 18 defines the security Object Identifiers specified by these Implementation Agreements. These are the same as defined in the EWOS/ETSI A/3311 profile.

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

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

-- Security Policy Object Identifiers --

security-class-0               OBJECT IDENTIFIER ::= { id-policy-id 0 }
security-class-0a              OBJECT IDENTIFIER ::= { id-policy-id 0 1 }
security-class-1               OBJECT IDENTIFIER ::= { id-policy-id 1 }
security-class-1a              OBJECT IDENTIFIER ::= { id-policy-id 1 1 }
security-class-2               OBJECT IDENTIFIER ::= { id-policy-id 2 }
security-class-2a              OBJECT IDENTIFIER ::= { id-policy-id 2 1 }

-- Security Category Object Identifiers --

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

Figure 18 - Security object identifiers

Annex C (informative)

Interpretation of elements of service

The objective of this clause is to provide clarification, where required, on the functionality of Elements of Service where the MHS standards are unclear or ambiguous. it is not the intent of this clause to define how information should be made available or presented to an MHS user, nor is it intended to define how individual vendors should design their products.

The following MHS Elements of Service require further text to be added to their definitions to represent the proposed implementation of these Elements of Service for conformance to this Agreement. Elements of Service which are not referenced in this clause are as defined in the MHS base standards.

Reply Request Indication: The reply-recipients and the reply-time may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request.

NOTE - For an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Forwarded IP-message Indication: The following use of the original encoded information type in the context of forwarded messages is clarified:

a) The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.

b) if forwarding a privately defined body part (see figure 10), the originator of the forwarding message shall set the original encoded information types in the P1 envelope to Undefined for that body part.

Annex D (informative)

Recommended practices

This clause 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 OIW 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 clause may result in different solutions to those proposed in this clause.

D.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 49 are covered by the category "other."

Table 49 - 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, table 48 and the algorithm in figure 19 should be used.

```
IF current character is in the encoding set THEN
  encode the character according to table 48
ELSE
  write the current character;
continue reading;
```

Figure 19 - ASCII to PrintableString algorithm

To decode a PrintableString representation to an ASCII representation, table 48 and the algorithm in figure 20 should be used.

```
IF current character is not "(" THEN
  write character
ELSE
  {
  look ahead appropriate characters;
  IF composite characters are in table 48 THEN
     decode per table 48
  ELSE
  write current character;
  }
continue reading;
```

Figure 20 - PrintableString to ASCII algorithm

D.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 50.

Table 50 - 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 - X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

D.3 EDI use of MHS

This section outlines a recommended method for interworking between a P(edi) UA with a UA implementing the Recommended Practice (EDI Use of X.400) in parts 7 and 8 of the OIW Stable Implementation Agreements. That Recommended Practice is commonly referred to as the "P0" approach to EDI use of the X.400 MTS.

This section does not define where the conversion between the two content types occurs. It is possible for the conversion to be performed by the P0 UA, the P(edi) UA, or a gateway. The Recommended Practice outlined in this section only attempts to document the rules that should be followed to ensure a conversion which retains the maximum amount of information.

D.3.0.1 P0 to P(edi) conversion

The converting entity may assume that the P0 content contains only one EDI interchange. This interchange will become the first and only body part of the EDIM.

The content type field of the message will have the value "undefined" before the conversion and will have the integer value "35" or the object identifier value for P(edi) which is specified in X.435 after conversion. The EDIM Heading fields can be formed using the following rules:

EDIMIdentifier: Originator ORName concatenated with the UTCTime at which the conversion from P0 to P(edi) was performed.

Originator: Originator ORName.

Recipients: Recipients from the P1 envelope. EDI Notification Requests are not specified as none are requested when using the P0 approach.

EDIBodyPartType: This element may have one of deveral values depending on the encoded information type (EIT) value of the P0 message or the ability of the converting entity to determine which EDI syntax is present in the content:

a) X.435-defined value for ANSI X12/EBCDIC if the EIT field of the P1 envelope has the value "undefined".

b) X.435-defined value for ANSI X12/ISO 646 if the EIT field of the P1 envelope has the value "IA5String".

c) Any other valid value if the entity performing the conversion can determine which EDI syntax is contained in the content and which character encoding is used for the EDI syntax.

Other heading fields will only be set if the entity performing the conversion is capable of parsing the EDI Interchange and discovering the correct values of EDI Heading fields.

As the P0 message will not contain requests for EDI Notifications, an EDI UA will never create an EDIN when it receives an EDIM converted from P0 .

D.3.0.2 P(edi) to P0 conversion

When converting a P(edi) content to a P0 content, the following rules apply:

The first body part of the EDIM will be copied to the content. **All other body parts of the EDIM will be discarded.**

The P1 envelope fields shall have the following values:

Content Type: Value for "undefined".

Originator: Originator ORName.

Recipients: Recipients from the EDIM Heading. An NN EDIN with NN Reason Code set to the value "unspecified" is created for each Recipient for whom a Notification Request was specified. The EDIN Originator is set to the Recipient ORName. It is recommended that the supplementary information field of the NN be used to provide additional information on the disposition of the EDIM.

Encoded Information Types (EITs): This element may have one of several values depending on the value of the EDI Body Part Type:

a) The EIT is set to "undefined" if the EDI Body Part Type is encoded with the EBCDIC character set.

b) The EIT is set to "IA5String" if the EDI Body Part Type is encoded using the ISO 646 (ASCII) character set.

c) A value is not present for the EIT if EDI Body Part Type does not contain one of the above mentioned values.

D.3.1 P2 recommended practice

As there are a substantial number of users in the NIST OIW community that implemented the CEC TEDIS "P2" approach to EDI use of the X.400 MTS, this section will also include text that describes interworking between a P(edi) UA and a P2 UA. This text is not maintained by the EDI Working Group of the NIST OIW X.400 SIG but is included for the convenience of our user community. Users intending to interwork between P2 and P(edi) User Agents should consult the current version of the EWOS/ETSI document "A/3331 - Functional Profile of an Electronic Data Interchange User Agent." This will ensure that the most up to date

technical information is obtained.

D.3.1.1 Conversion from IPMS to EDIMS (P2 to P(edi))

It is assumed that there is one and only one body part in the IPM Message, and that this body part contains an EDI interchange.

The IPM becomes the first, and only, body part of the EDIM.

The EDIM Heading fields are set as follows:

EDIMIdentifier: Originator ORName concatenated with the LocalIPMIdentifier portion of the IPM identifier.

Originator: Originator ORName.

Recipients: Recipient ORNames from the IPM Heading. The edi-notification-requests-field is not coded.

EDIBodyPartType: The value is a local implementation issue. If the entity performing the conversion can identify the EDI syntax of the EDI interchange then it can specify an appropriate value. Otherwise, the entity must be assuming a specific encoding and will specify the value for the syntax it is assuming.

Other heading fields may be set if the entity performing the conversion is capable of parsing the EDI interchange and discovering the correct values of the EDIM Heading fields.

Since there are not notification requests, the EDI UA will never create an EDIN when it receives a converted EDIM and therefore the action for handling EDINs in the reverse direction does not need to be considered.

D.3.1.2 Conversion from EDIMS to IPMS (P(edi) to P2)

> **NOTE** - The verification of authority to perform a particular conversion is outside the scope of this annex. It is assumed that such conversion will be done with the full knowledge of the originating and recipient parties.

The EDIBodyPart of the EDIM will be copied to the IPM body as an IA5TextBodyPart. All other body parts of the EDIM will be discarded.

The IPM Heading fields are set as follows:

IPM Identifier: EDIMIdentifier.

Originator: Originator ORName.

Recipients: Recipients from the EDIM Heading. All recipients become IPM Primary Recipients. An NN EDIN with NN Reason Code set to the value "unspecified" is created for each Recipient for whom a Notification Request was specified. The EDIN Originator is set to the Recipient ORName. The EDIN Originator is set to the Recipient ORName. IPM Notifications shall not be requested.

Subject: Not present or set to a single blank character.

If EDINs have been requested the originator will always receive an NN. Since no IPM notifications are requested, the IPM UA will never create an IPM notification when it receives an IPM converted from an EDIM and therefore handling of notifications in the reverse direction does not need to be considered and is not an option for generating EDINs.

D.4 ODA transfer

To ease interworking with 1984 implementations when transferring Office Document Architecture (ODA) documents, the following are recommended for 1988 implementations:

a) Origination UA implementing 1988 Implementation Agreements. The 1988 will generate the ODA according to CCITT Recommendation T.411 Annex E for the destination UA(s) implementing 1988 Implementation Agreements. If the destination UA supports 1984 Implementation Agreements, the approach as described in section 7.12.8 is recommended.

b) Recipient UA implementing 1988 Implementation Agreements. The recipient system will be able to handle the ODA bodypart in P2 (1984) as defined in part 7, B.8.1 for interworking with 1984 implementation, and will also be able to handle the ODA bodypart as defined in the appropriate base standards.

c) MTA downgrading rules. When transferring an P22 with ODA body part in P22 as described in T.411 to an 1984 MTA, the EITs identified by ODA Object Identifiers are mapped to bits 0 and 10 of the built-in EITs.

If the UA does not register to support P22 or ODA bodypart, a Non-Delivery-Report will be generated as required.

D.5 Use of externally defined body part

D.5.1 General

An Externally Defined body part represents an information object whose semantics and abstract syntax are denoted by an Object Identifier which the body part carries. This body part type enables the exchantge of information objects of all kinds, each unambiguously and uniquely identified.

The Externally Defined Body Part definition is reproduced in figure 22.

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

ExternallyDefinedParameters ::= EXTERNAL
ExternallyDefinedData       ::= EXTERNAL

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

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

Figure 22 - Externally Defined body part definition

On the basis of the Externally Defined body part type, all body part types are divided into two important classes as follows:

a) *basic:* Said of any body part type except Externally Defined. All basic body part types are denoted by an integer (an ASN.1 context-specific tag) and are defined in section 7.3 of X.420.

b) *extended:* Said of the Externally Defined body part type restricted to any one value of the Direct-reference component of the Data component of such a body part. Denoted by an Object Identifier.

Annex B of Recommendation X.420 defines some (but not necessarily all) extended body part types.

D.5.2 Use of equivalents of basic body part types

For each basic body part types, section B.1 of Recommendation X.420 defines an equivalent extended body part type. In order to facilitate interworking with 1984 systems, use of these extended body part types is not recommended; the basic body part types should be used instead.

Editor's Note: The requirements of this clause may change when interworking with 1984 systems is no longer critical.

D.5.3 Use of General Text body part type

Unless otherwise specified in these agreements (e.g., IA5Text, 6937Text, Teletex) the General Text body part as defined in ISO 10021-7 Annex B.2 is the preferred means of supporting unstructured text body parts. The character set registration referred to in that annex is provided by ECMA.

D.5.4 Use of File Transfer body part type

The File Transfer body part type is the recommended mechanism for the exchange of complex computer data via intra- and inter-company X.400 messages. It enables automatic type recognition for the file being sent and, possibly, automatic invocation of the appropriate application necessary to process the data.

D.5.4.1 Encoding of General Identifier

In order to optimize the machine-processing of information encoded in the Parameters and to enable registration, it is recommended that, if present, General identifiers should be encoded as Object identifiers.

D.5.4.2 Encoding of Contents Type

It is recommended that the Contents Type parameter be encoded as document type. The encoding as constraint-set-and-abstract-syntax has been provided only for backward compatibility with FTAM and its use is discouraged.

D.5.4.3 Encoding of application specific information

The type of a file can be considered from several perspectives:

a) As a specific data structure consisting of a sequence of presentation data values - the position taken by the FTAM standard;

b) As the output of a certain application - the position taken by e-mail users requiring the interchange of office documents.

The fact that registered OSI document types have to be recognized by FTAM implementations and be described according to the requirements of ISO/IEC 9834-2 "Registration procedures for OSI document types" makes use of the Contents Type parameter inappropriate for expressing point of view (b).

Considering that the environment parameter "application-reference" could describe not only the application that generated a document but, more generally, the application-level format of the document, it is recommended that the values given to the "application-reference" parameter component be Object identifiers associated with such a format.

Example: If an Object identifier has been associated with a certain word-processing file format then this Object identifier should be used as the value of "application-reference" when a file of that format is carried by a File Transfer body part, while the Content Type parameter should have as its value the Object identifier associated with the "unstrucutred-binary" document type.

D.5.4.4 EITs for the File Transfer body part

It is recommended to use only the id-eit-file-transfer Object identifier in association with the File Transfer body part.

The use of EITs describing other parameters of the File Transfer body part such as contents types, application references, etc. would force all potential recipients to register a possibly large number of EITs in order to avoid non-delivery of messages.

D.5.5 Use of other extended body part types

The following are guidelines regarding the use of Externally Defined body part types not defined in the X.400 or other standards:

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

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

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

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

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

NOTE - Use of proprietary Externally Defined body part types is recommended only if the extended body part types already defined in the standards do not provide the apporpriate functionality.

In order to communicate with 1984 systems, the use of the Bilaterally Defined body part is recommended.

D.5.6 Obtaining object identifiers

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

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

b) The application provider (organization) allocates a series of numbers to identify the application data format; these numbers are appended to the object identifier constructed in step (i) to form an object identifier that is globally unique. It is recommended that the application provider

(organization) use a hierarchical structure for identifying their data types to ease the administration of the identifiers.

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

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

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

D.6 Privacy Enchanced Mail body part

This clause describes a mechanism to convey an Internet Privacy Enhanced Mail (PEM) message across an X.400 MHS. PEM is described in internet RFCs 1113, 1114, and 1115 and their successors.

The general internet mail message format is described in RFC 822. Mapping of RFC 822 messages to and from X.400 inter Personal Messages is described in RFC 987 for 1984 X.400 and in RFC 1148 for 1988 X.400.

The PEM message is conveyed as a P2(2) body part. All of the RFC 822 header information is conveyed in the P1 envelope and P2 header per RFC 987 and RFC 1148. The PEM message (encapsulated security header and, possibly encrypted, message text as described in RFC 1113) is conveyed in a single body part. On the X.400 side, this body part may be manipulated like any other body part; e.g., it may be included in a multi-part body.

For 1988 (P22), the PEM body part is externally defined and does not require parameters. This definition is provided in figure 23.

```
privacy-enhanced-mail      EXTENDED-BODY-PART-TYPE
                             DATA  OCTET STRING
                     ::=  id-privacy-enhanced-mail

-- The object identifier is defined in annex B.
```

Figure 23 - Definition of the Privacy Enhanced Mail body part type

For interworking with 1984 (P2) systems, a USA body part (integer) will be allocated by NIST as described in figure 10.

D.7 Selection of OR name attributes

To support the transition to addresses with Teletex components, it is recommended that a printable string alternative address be established for each address containing Teletex strings.

D.8 Use of the Teletex body part

The Teletex body part should be used purely for structured teletex documents, as described in F.200 and T.60, obeying page rules, etc. It should not be used to transfer T.61 characters, in a general sense, across the MTS. If only IA5 characters are being used, the IA5Text body part should be used, especially when interworking with 1984 UAs is relevant. Otherwise, the GeneralText body part should be used to transfer unstructured character data.

D.9 Provision of security class S0A using asymmetric algorithms

This clause describes one method of providing the security services of class S0A when using asymmetric (public key) cryptographic algorithms. It is recommended that this method be used unless the security requirements or policy specifies otherwise. Asymmetric cryptographic algorithms such as RSA are used to provide digital signatures in support of the content integrity and (end-to-end) message origin authentication services, as well as proof of delivery. Since asymmetric algorithms are used, the non repudiation of origin and non repudiation of delivery services of security class S2 are also provided. Content confidentiality is provided using a combination of symmetric and asymmetric encryption. The following paragraphs discuss the protocol elements used to provide these services, as well as certificate management and other issues.

D.9.1 Protocol elements

The following protocol elements are provided by the originating UA in the submission envelope in support of the S0A security services.

Content: If the content confidentiality services is required, the message content is encrypted under the content confidentiality key.

Content Integrity Check: This per-recipient security element is a signature over the message content, and provides the content integrity, message origin authentication, and non repudiation of origin services if content confidentiality is not required. (If the message is encrypted, the content integrity check is included in the message token.)

> **NOTE** - The message origin authentication check provides a single signature, rather than a signature per recipient, thus reducing total message size in the case where multiple recipients are present. However, support for this protocol element is optional for security class S0. In addition, it is computed over the message content as sent (i.e., the encrypted content if content confidentiality is used). If the content is encrypted, this protocol element does not truly provide non repudiation of the unencrypted content. In this case, smaller message size was traded off for the additional service of non repudiation.

Proof Of Delivery Request: This per-recipient security element is used to request the recipient to generate

a proof of delivery, in the case where content confidentiality is not used. (Where content confidentiality is used, the proof of delivery request is included in the message token, as shown below.)

Originator Certificate: This security element is a set of one or more certificates which the recipient may use to obtain the originator's public key. For example, it might contain the chain of certificates from the originator, through the certification hierarchy to a top-level certification authority.

Message Token: The asymmetric message token conveys security information from an originator to a single recipient. It is a signed structure, some of whose fields may be encrypted. The message token is used only when content confidentiality is desired, and supports the content integrity, message origin authentication, content confidentiality, and non repudiation of origin services. The following fields are required, and all other fields are optional:

- *Signature Algorithm Identifier:* The algorithm identifier of the asymmetric algorithm used to sign the token.

- *Recipient Name:* The OR Address and/or Directory Name of the recipient with whom the token is associated. Since the encrypted portion of the token is encrypted under the recipient's public key, it is recommended that the directory name be included, since the recipient's certificate contains his/her directory name rather than OR Address.

- *Time:* The time of day when the token was generated.

- *Signed Data:* The following fields are signed but not encrypted:

a) *Content Confidentiality Algorithm Identifier:* The algorithm to be used to encrypt the message content.

b) *Proof of Delivery Request:* This element is used to request the recipient to compute a proof of delivery over the received message.

- *Encrypted Data:* These fields are encrypted under the recipient's public key:

c) *Content Confidentiality Key:* The symmetric key used to encrypt the message content.

d) *Content Integrity Check:* A signature on the unencrypted message content. If content confidentiality is required, this element provides the content integrity, message origin authentication, and non repudiation of origin services. This signature is encrypted in order to protect against the "low entropy" attack described in Internet RFC 1113. (In RFC 1113, the signature is encrypted under the content confidentiality key.)

NOTE - The encrypted portion of the token will then comprise two RSA encryption blocks.

The following element of service is generated by the recipient, if requested by the originator.

Proof Of Delivery: This security element provides proof and non repudiation of delivery. It is a digital signature computed over the received (possibly encrypted) message content and various delivery envelope fields, as defined in the base standard.

D.9.2 Algorithm selection

This clause makes no recommendation as to hash algorithms, asymmetric encryption algorithms, or symmetric encryption algorithms. The Implementor must select appropriate algorithms, based on factors such as performance, cost, and licensing and export restrictions. A fairly complete list of algorithms can be found in clause 7 (Security Algorithms) of Part 12 of these Agreements. In some cases, the implementor must also specify certain algorithm-dependent information. For example, when using the symmetric algorithm **DES-CBC**, the Implementor must specify the padding mechanism used, since this algorithm operates on 8-byte input blocks. Internet RFC 1115 defines such padding rules for DES and RSA in various modes, and these mechanisms are recommended unless security requirements dictate otherwise. PKCS #1 (see Bibliography, Annex F) discusses such matters in more detail.

D.9.3 Certificate management

Management of public key certificates is beyond the scope of this recommended practice. X.509 provides a generic authentication framework which uses the Directory to store certificates. In the absence of a ubiquitous Directory, local means may be used to obtain certificates. For example, the recipient of a message might choose to cache those certificates received in the **OriginatorCertificate** protocol element of the delivery envelope.

Each community of interest will define its own policy regarding certificate management and the associated trust model. An example of a centralized trust model can be found in Internet RFC 1114, while the most complete example of a decentralized trust model can be found in the paper on Digital's Distributed System Security Architecture cited in the Bibliography (Annex F).

D.9.4 Other Issues

In the case of the P2 content type, addressing information may be protected by replicating the P1/P3 recipient names in the P2 heading fields (To:, CC:, and BCC:). The X.400 security services discussed above are applied to the entire P2 IPM, including the heading and all body parts. Additional protection of heading and envelope fields may be provided using double enveloping.

When using X.400 (1988) distribution lists (DLs), one might choose to distribute the private key associated with the DL to all members of the DL. This allows an originator to create a single message token in which the content confidentiality key is encrypted under the DL's public key. (This requires support of the DL expansion history protocol element on delivery, so that the recipient may select the proper private key for decryption. Alternatively, the originating UA may expand the DL locally and generate a message token for each member [recursively]). There is no architected support for this mechanism in the base standard, nor is there architected support for performance of this function by an MTA when expanding a DL.

Annex E (informative)

Secure messaging guidelines

E.1 Introduction

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

E.2 Message handling vulnerabilities

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

a) Masquerading;

b) Message sequencing;

c) Modification of information;

d) Denial of service;

e) Repudiation;

f) Leakage of information.

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

a) Manipulation;

b) Misrouting.

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

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

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

E.3 General principles

E.3.1 Security policy

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

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

E.3.2 Security classes

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

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

These Implementation Agreements includes a set of security classes; each class stipulates a set of mandatory and optional security services. The security classes are incremental subsets of the security features in the MHS Base Specifications:

Security Class S0 only requires support of end-to-end security services between UAs (content integrity, message origin authentication, proof of delivery), and hence can be used to provide some protection even in the case of transit through an intermediary MTS which may not be trusted.

Security Class S1 additionally requires support and use of secure access management within the MTS so as to allow the enforcement of a label-based security policy and enable trusted interworking between security domains.

Security Class S2 additionally requires support and use of origin authentication checks within the MTS to verify the origin of messages, probes, and reports, thereby making it possible to provide non-repudiation within the MTS.

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

E.3.3 Dynamic behavior requirements

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

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

E.3.4 Encryption techniques

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

a) Symmetric encryption;

b) Asymmetric encryption;

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

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

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

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

The actual technique employed depends on the algorithm used. Algorithms are assumed to be bilaterally agreed or registered by a registration authority. However, the algorithm-identifier must be unique and unambiguously define the algorithm.

It is recommended that a conforming ASN.1 BIT STRING is normally used to contain the encrypted data (as generated by use for the ENCRYPTED macro), thereby ensuring insertion of padding zero bits which may be necessary for correct operation of certain algorithms. Alternatively, the implementation should take such action explicitly.

It is recommended that, in the absence of any requirement for support of other specific algorithms,

Implementations shall as a default support algorithms identified in CCITT X.509 (ISO/IEC 9594-8). It is also strongly recommended that implementations are capable of using any encryption-based technique on a "plug-in" or modular basis.

In the case of verification of SIGNATUREs (e.g., proof of delivery, MOAC, POAC, or ROAC), implementations should assume that all relevant data present in the subject message, probe, or report has been included in the signature.

E.3.5 Implementation considerations

E.3.5.1 Peer Entity authentication

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

E.3.5.2 Confidentiality

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

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

E.3.5.3 Integrity

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

Content integrity is achieved by computing a content integrity check as a function of the entire message content. When symmetric techniques are used to compute the content integrity check a secret key is required. This content integrity key may be confidentially sent to the message recipient using the message argument confidentiality security element in the message token (i.e., there may be other keys or parts of the key not sent by the originator with the message, but the key management of such external keys is outside the scope of these Implementation Agreements). It should be noted that placing the content integrity check in the encrypted data of the message token will provide additional protection against masquerade threats.

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

E.3.5.4 Message origin authentication

End-to-end (i.e., UA to UA) Message Origin Authentication is automatically provided by content integrity. Security classes S2 and S2a provide additional protection (i.e., of the integrity of the label) by requiring support of origin authentication checks within the MTS.

E.3.5.5 Non-Repudiation

If asymmetric techniques are used for content integrity it can also provide non-repudiation of origin (UA to UA) depending on the level of trust placed in the certificate. If symmetric techniques are used, content integrity can also provide non-repudiation of origin, but only by using a trusted notary to validate the content integrity and provide trusted key management facilities. A degree of non-repudication can be provided by the use of trusted accountability services.

> **NOTE** - It is assumed that an originating UA will ensure that delivery notification is requested when proof of delivery is requested.

E.3.5.6 Secure access management

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

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

E.3.5.7 Implications for the use of distribution lists

An MTA performing distribution list expansion must create all the per-recipients fields for the members of the distribution list. It may either generate a new token for each DL member (i.e., using the recipient name of that DL member) or alternatively it may copy the same token (i.e., containing the recipient name of the DL itself) into the per-recipient fields for each DL member. In the former case, the content-integrity-check should not be changed if it is to be used to provide message origin authentication. Also in such case, the DL expansion point must have at least the same security class as the originator and must have trusted functionality. The choice of which approach to use will therefore need to be determined in accordance with the security policy which may prohibit the use of distribution lists altogether.

E.3.5.8 Implications on redirection

The Security Functional Group has the effect of either requiring trust in any redirection facilities or prohibiting the use of redirection. If the Redirection facility is to be trusted, it must be subject to the security policy and obey the security labels as defined in the base standards. It is recommended that the token is not altered

on redirection (i.e., it will contain the originally-specified recipient name).

E.3.5.9 Implications for 1984 interworking

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

E.3.5.10 Implications for use of Directory

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

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

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

E.3.5.11 Implications for conversion

Implementation of the Security functional group may additionally either require that any conversion facilities are highly trusted to regenerate the appropriate security elements (notably the content integrity check) or prohibit the use of conversion within the MTS altogether. In particular, it should be noted that use of conversion facilities will invalidate any origin authentication based on the original content.

E.3.5.12 Accountability

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

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

E.3.5.13 Double enveloping

Double enveloping can be used with each secure messaging security class. For each security class it is an optional extension to the security features which can be used to counter specific vulnerabilities. When double enveloping is used, it shall be applied at the boundary of a domain, and obey the rules of an MTA at management domain boundaries. Figure 24 illustrates the technique.

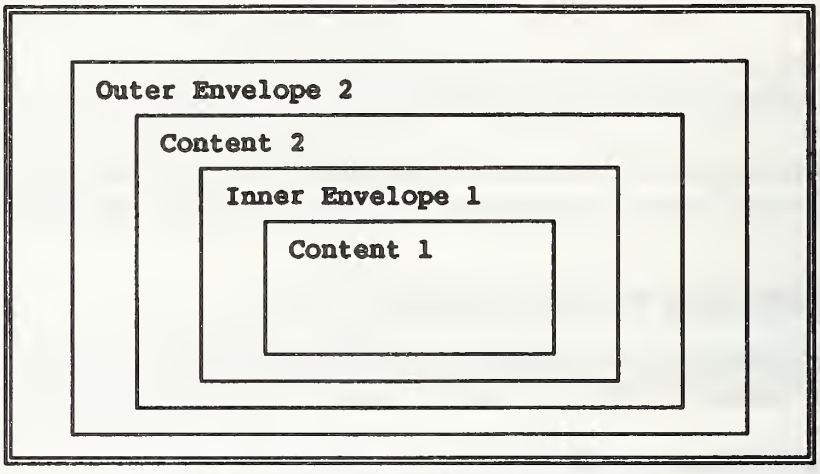


Figure 24 - Double enveloping technique

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

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

The double envelope technique can be used in 1984 and 1988 MTS environments. When used in an 1988 environment, any security class can be applied to the outer envelope. It is recommended that the inner envelope is encrypted. When the double envelope technique is used as a secure relay path via an 1984 domain, any encryption of the content 2 is subject to bilateral agreement.

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

E.4 Security class S0

E.4.1 Rationale

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

a) Content integrity;

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

c) Proof of Delivery.

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

E.4.2 Technical Implications

The technical implications for security class S0 are:

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

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

E.5 Security class S1

E.5.1 Rationale

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

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

E.5.2 Technical Implications

The technical implications of security class S1 are:

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

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

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

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

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

f) it is necessary that reliable access is always supported;

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

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

E.6 Security class S2

E.6.1 Rationale

Security Class S2 is a superset of Security Class S1. It enhances the facilities of the MTAs in order to check the origination of messages, probes, and reports within the MTS and to provide enhanced integrity checks on the security label while in the MTS. The extra security services provided by this security class can help to provide trusted routing within an MTS.

Additionally, it is possible to provide non-repudiation within an MTS.

E.6.2 Technical implications

The extra security services specified by Security Class S2 use asymmetric techniques exclusively.

The technical implications are as in Security Class S1, plus:

a) it is necessary to provide mechanisms in an MTA and UA that can process the signed macro of certificates;

b) The constraint that the option of supporting Content Confidentiality cannot be allowed when the message origin authentication check (MOAC) is used to provide non-repudiation services. Under this condition Content Confidentiality is not supported. If the MOAC is not used for this purpose, Content Confidentiality can be supported as an optional security service;

c) it is necessary to provide mechanisms in a MTA which can generate and process the signature macro of a message, probe, and report authentication check (MOAC, POAC and ROAC);

d) it is necessary to provide mechanisms in a UA and MTA that can interface with an X.500 directory supporting the Authentication Framework as defined in X.509/ISO 9594-8 or can distrubute public keys by other trusted means which is compliant with X.509;

e) it is necessary to provide a trusted means of generating certificates;

f) it is necessary to provide mechanisms in the MTA which can process a proof of submission request and generate the proof of submission signature;

g) it is necessary to provide a mechanism in an MTA which will generate ROAC signatures with reports;

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

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

E.7 Confidential security class variants (S0a, S1a, and S2a)

E.7.1 Rationale

These security class variants are supersets of S0, S1, and S2, adding the requirement for support of end-to-end content confidentiality. The rationale for these variants is to avoid the implementation cost and processing overhead involved in encrypting the entire message content unless there is a definite requirement. It is also possible to protect the encryption techniques and mechanisms (i.e., algorithms, key lengths, key versions, etc.) by Secure Access Management.

E.7.2 Technical Implications

The technical implications of the confidential security class variants are the same as those for the corresponding primary security class, plus:

a) it is necessary to provide mechanisms in a UA which can use the encrypted macros to encrypt and decrypt the message content.

Annex F (informative)

Bibliography

F.1 ANSI

Procedures for Registering Organization Names in the United States of America, ISSB 843, December 5, 1989.

Procedures for Registering Names in the United States of America, ISSB 840, December 5, 1989. The U. S. Register is included.

F.2 Internet

Message Encipherment and Authentication Procedures, RFC 1113.

Certificate-based Key Management, RFC 1114.

Algorithms, Modes, and Identifiers, RFC 1115.

Annex G (informative)

Defense message handling profiles

G.1 Introduction

Several additional requirements for Message Handling Systems (MHS) in the U.S. Department of Defense (DoD) are currently being investigated by the Data Communications Protocol Standards (DCPS) Technical Management Panel (DTMP). This annex describes the DoD Standardized Profile(s) (DSP) that are required for Defense Message System (DMS) use.

Two multipart DoD profiles are currently defined, namely:

- DSP AMH1n(D) - Information Technology - Defense Standardized Profiles AMH1n(D) - Message Handling Systems - Common DoD Messaging

- DSP AMH2n(D) - Information Technology - Defense Standardized Profiles AMH1n(D) - Message Handling Systems - Military Messaging

These profiles will be published as part of the MIL-STD-2045 series. The AMH1n(D) profile consists of a DoD delta to the AMH1n ISP. AMH2n(D) is a standalone profile of a new military messaging content type (P772) based on the IPM content type. These extensions support military-unique functionality required by the DMS.

For further information on these profiles, contact:

DTMP WG/2 Chairman
c/o Defense Information Systems Agency (DISA)
Joint Interoperability Engineering Office (JIEO)
Code TBBD
Fort Monmouth, NJ 07703-5000
Phone: 908-532-7726

Annex H (informative)

Differences between OIW Agreements and EWOS/ETSI Draft Profile A/3312

H.1 P7

The "and," "or," and "not" elements of Filter are optional in A/3312.

The "equal," "greater-or-equal," "less-or-equal," and "present" elements of FilterItem are optional in A/3312; however, at least one must be implemented.

The List and Summarize operations are optional for the UA Kernel in A/3312.

The "precise" element in the Fetch operation is optional for the UA Kernel in A/3312.

Index

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 9 - FTAM Phase 2

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Joe Mohen, Proginet**
SIG Editor: **Larry Friedman, Digital Equipment Corporation**

Foreword

This part of the Stable Implementation Agreements was prepared by the File Transfer, Access and Management Special Interest Group (FTAM SIG) of the Open Systems Environment Implementors' Workshop (OIW). See Procedures Manual for Workshop charter.

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject. There is no significant technical change from this text as previously given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

List of Tables

List of Figures

Part 9 - File Transfer, Access and Management Phase 2

NOTE - In document type names, constraint set names, and abstract syntax definitions, the "NBS" designation will be preserved.

0 Introduction

This part 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 part) is based on the international Standard (IS).

1 Scope

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 part 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.

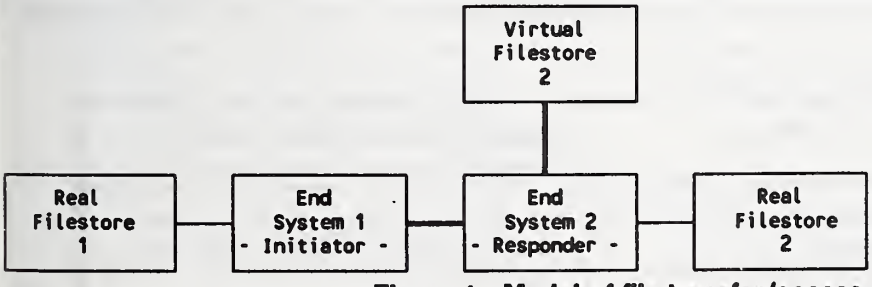


Figure 1 - Model of file transfer/access.

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 clauses 5 through 16, minimum functionality (Conformant Implementations) in clause 17, and functionality for several Implementation Profiles which are tailored to different classes of user requirements in clauses 18 and 19.

2 Normative References

ISO 8571-1: 1988(E), Information Processing Systems - Open Systems interconnection - File Transfer, Access and Management Part 1: General introduction

8571-2: 1988(E), Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 2: Virtual Filestore Definition ISO

ISO 8571-3: 1988(E), Information Processing Systems - Open Systems interconnection - File Transfer, Access and Management Part 3: The File Service Definition

ISO 8571-4: 1988(E), Information Processing Systems - Open Systems interconnection - File Transfer, Access and Management Part 4: The File Protocol Specification

3 Status

Version 3 of the FTAM Phase 2 implementation Agreements was completed December 15, 1989, and published in version 3 of the Stable implementation Agreements (NIST Special Publication 500-177, December 1989). Some editorial clarifications and technical changes (due to international harmonization of Profiles) were added to Version 3 Agreements in the course of 1990. All these changes are now fully incorporated in this version 4 of the FTAM Phase 2 Stable Agreements.

No further enhancements will be made to this version 4 of FTAM Phase 2 (see the clause 4 ERRATA).

This version 4 of the FTAM Phase 2 Agreements fully replaces the version 3 as published in NIST SP 500-177. Therefore, the old version 3 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 AETitles	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	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

4 Errata

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	CLAUSE	
CP 3/91-1	EDITORIAL	NIST-SP 500-183	All	Update to ISO style. General formatting and error corrections. Alignment with the wording of the ISP. Consistent naming conventions.
CP 9/91-1 CP 9/91-2 CP 9/91-3 CP 9/91-4 CP 9/91-5 CP 9/91-6	EDITORIAL	NIST-SP 500-183	Table 9 Table 10 Table 11 Clause 17 Annex A A.5.11.1 A.6.11.1 A.7.7 A.7.9.1 A.7.9.2	Datatype1 ASN.1 comment to use the object descriptor Include Corrigenda Editors note to point at ISP for registered objects. Change header to Structural Simplification. Added definitions for Datatype3

5 Assumptions

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.

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.

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.

Consideration of any standardized service interface is not covered by these Agreements.

These Agreements define no restrictions for the values used for the <communication quality of service> parameter in <F-INITIALIZE>.

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 part) is based on the International Standard (IS). THERE IS NO BACKWARD COMPATIBILITY WITH 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 OSI Implementors' Agreements or with respect to FTAM profiles of other bodies (see clause 12). 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 FTAM Phases, facilitating backward compatibility for future FTAM products, assuming different new versions of the respective IS's also enable backward compatibility.

The FTAM specific ASE requirements for ACSE in the Upper Layers part 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.

Use of <shared ASE information> parameter and <charging> parameter is not defined within the scope of the Agreements.

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.

These Agreement 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.

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 also annex C.

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

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 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 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.

9.1 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.

9.2 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.

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 annex 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 1

Table 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 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 2.

Table 2 - Parameters for NBS-6, NBS-7, NBS-8

Parameter	Prim Type	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 number 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 3.

Table 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 annex C.3. It is derived from existing standards IEC 559 and IEEE 754.

10.1 Character set

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

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.

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 4.

Table 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)

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. It is recommended that OSI implementations use <CR><LF> pairs as line terminators. Failure to follow this recommendation may result in an

implementation which does not interoperate with other implementations conforming to these agreements.

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>.

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.

10.2 Document type Negotiation rules

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.

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.

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.

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:

structural relationship between DUs,

structure of DUs, called DU syntax, and

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.

11 F-CANCEL action

When an F-CANCEL is sent or received, the following occurs:

no more data is sent,

checkpoint numbers are removed, and

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).

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 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.

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.

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.

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."

16 Security

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.

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.

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.

17 Requirement for conformant implementations

This clause 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 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. Clause 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.

Implementations conforming to these Agreements require adherence to the General Agreements in clauses 5 through 16 of these Agreements.

Implementations conforming to these agreements shall implement the defect report solutions contained in:

ISO 8571-1/Cor.1:1990
ISO 8571-2/Cor.1:1990
ISO 8571-3/Cor.1:1990
ISO 8571-4/Cor.1:1990

ISO 8571-3/Cor.2:1990
ISO 8571-4/Cor.2:1990

> **Editor's Note** - The corrigenda ISO 8571-3/Cor.2 and ISO 8571-4/Cor.2 is to be published. Until it is available, the solutions can be found in the documents ISO/IEC JTC1/SC21 N 5234 and ISO/IEC JTC1/SC21 N 5235.

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 17.3 to 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 17.3 to 17.8).

These role combinations and this interoperability are shown in table 5 below.

Table 5 - Interoperable configurations.

		Initiator		Responder	
		sender	receiver	sender	receiver
Initiator	sender				x
	receiver			x	
Responder	sender		x		
	receiver	x			

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 subclauses 17.3 to 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.

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 clause 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 clause 10

the preferred transfer syntax "Basic Encoding of a single ASN.1 type"

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 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 17.6 and 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.

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 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 17.6 and 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:

If the failure was due to either a protocol or Filestore failure, and then:

1) precisely which action failed,

2) 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

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>.

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.

17.6.1 Initiator Senders

Every implementation of an FTAM sender which is also an FTAM Initiator shall support:

the write functional unit and protocol, and

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).

17.6.2 Responder Senders

Every implementation of an FTAM sender which is also an FTAM Responder shall support:

the read functional unit and protocol, and

for the document type FTAM-3 the following bulk data transfer specification parameters:

1) FADU identity first

2) Access context UA

and support the applicable procedures, defined in ISO 8571-4 clause 14 (Bulk data transfer responding entity actions).

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.

17.7.1 Initiator Receivers

Every implementation of an FTAM receiver which is also an FTAM Initiator shall support:

the read functional unit and protocol, and

for the document type FTAM-3 the following bulk data transfer specification parameters:

1) FADU identity first

2) Access context UA

and support the applicable procedures, defined in ISO 8571-4 clause 13 (Bulk data transfer initiating entity actions).

17.7.2 Responder Receivers

Every implementation of an FTAM receiver which is also an FTAM Responder shall support:

the write functional unit and protocol, and

for the document type FTAM-3 the following bulk data transfer specification parameters:

1) FADU operation replace

2) FADU identity first

and support the applicable procedures, defined in ISO 8571-4 clause 14 (Bulk data transfer responding entity actions).

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 6. A conforming implementation may support a wider range of values for any parameter.

Table 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 17.4 and 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-create-with-old-attributes' is optionally supported. All values are optional for Initiators.

Table 6 - Required minimal parameter support. (concluded)

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 subclause 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.

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 clause of the Upper Layers part.

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 $2^{31}-1$.

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 part 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.

18 Implementation Profiles

This clause 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.

File Transfer

File Access

File Management

Unconstrained

File Transfer and Management

NOTE - Any given implementation may support more than one service class.

Support of an Implementation Profile requires adherence to:

corresponding definition in 8571-3 clause 8 and any related procedures in 8571-4 clause 8-17,

requirements given in clauses 5-18 of these Agreements, and

requirements for parameter and attribute support as defined in 17.8.

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 clause 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.

18.2 (deleted)

18.3 Document type requirements for the defined Implementation Profiles

Implementations conformant to Implementation Profiles defined in table 7 will support the following document types with the caveats and procedures given.

FTAM-1

FTAM-2

FTAM-3

NBS-6

NBS-7

NBS-8

NBS-9

NOTES

1 Support of this document type entails the naming of FADUs by their position in preorder traversal.

2 Caveat: Other methods of naming FADUs depend on the system, application, and specific file, and as such

are not described here.

3 Those document types are defined in annex A and clause 10 of these Agreements, and in ISO 8571-2.

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.

NOTE - 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 annex A of these Agreements and, where defined, the parameterization scheme given in clause 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 clause 16.

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 clause 13 of these Agreements.

The <initiator identity> parameter is supported. Use must be consistent with clause 16 of these Agreements.

Implementations are not precluded from using other parameters for security and/or accounting. Responders must state the syntax and the semantics applying to <account> and <charging> parameters. The Responder's minimum implementation is to accept but ignore the <account>.

18.5 Parameter ranges for the defined Implementation Profiles

Parameter ranges for Implementations Profiles are as stated for primitive data types in clause 10 of these Agreements.

18.6 File attribute support for Implementations

Implementations of the Implementation Profiles will support file attributes or attribute groups in the following ways:

a) mandatory - This feature is mandatory in the ISO 8571-2 standard and shall therefore be implemented by all implementations claiming conformance to these Agreements;

b) 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;

c) 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;

d) not supported - This feature is outside the scope of these Agreements.

Kernel Group	mandatory
Filename	mandatory
Permitted Actions	mandatory
Contents Type	mandatory
Storage Group	optionally supported
Storage Account	optionally supported
Date and Time of Creation	optionally supported
Date and Time of Last Modification	optionally supported
Date and Time of Last Read Access	optionally supported
Date and Time of Last Attribute Modification	optionally supported
Identity of Creator	optionally supported
Identity of Last Modifier	optionally supported
Identity of Last Reader	optionally supported
Identity of Last Attribute Modifier	optionally supported
File Availability	supported
Filesize	supported
Future Filesize	optionally supported
Security Group	optionally supported
Access Control	supported
Legal Qualifications	optionally supported
Private Group	not supported

Table 7 - Implementation profile support requirements

Functional Units	Service Classes (see note 8)				
	T	M	A	TM	Uncon-strained
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 Management	see note 6	M1	see note 6		
Enhanced File Management		M1			
Grouping	T1, T2, T3	M1		see	see
File Access			A1, A2	note 4	note 5
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		
NBS-6	[T2], T3	[M1]	[A1], A2		
NBS-7	[T2], T3	[M1]	[A1], A2		
NBS-8	T3	[M1]	A2		
NBS-9	[T1], [T2], [T3]	[M1]			

NOTES

to 18.3 and table 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 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 clause 17. This is a general basic requirement and is not also reflected in table 7.

19 PROVISION OF SPECIFIC FUNCTION

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.

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"

FTAM-3 "ISO FTAM unstructured binary"

NBS-6 "NBS-6 FTAM sequential file" (optional)

NBS-7 "NBS-7 FTAM random access file" (optional)

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 or a single FADU which is identified by position, and/or

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.

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:

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 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to

read a complete file or a single FADU which is identified by key or by position, and/or

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.

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.

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.

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.

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 8 - Implementation profiles (OIW) and profiles (SPAG/CEN-CLC)

Implementation Profile	SPAG / CEN-CENELEC
T1	A/111
T2	A/112
T3	A/113
A1	A/122
A2	A/123
M1	A/13

Annex A (normative)

FTAM Document Types

Editor's Note - The objects defined in A.5 through A.8, B.2, C.3 and C.4 will be removed from this document after ISO/IEC ISP 10607-2 and ISO/IEC ISP 10607-2/Amd.1 are published. During the period between publishing the ISP and removal of the definitions from this document, the definitions in the ISP will take precedence over this document. When the object definitions are removed, clauses A.1 through A.4, B.1, C.1 and C.2 will be changed to point to the ISP.

The objects defined in A.5 through A.8, B.2, C.3 and C.4 will be removed from this document after ISO/IEC ISP 10607-2 and ISO/IEC ISP 10607-2/Amd.1 are published. During the period between publishing the ISP and removal of the definitions from this document, the definitions in the ISP will take precedence over this document. When the object definitions are removed, clauses A.1 through A.4, B.1, C.1 and C.2 will be changed to point to the ISP.

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 part 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 part 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 part 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 part 9, A.8.

A.5 NBS-6 Sequential file document type

A.5.1 Entry Number: NBS-6

A.5.2 Information objects

Table 9 - Information objects in NBS-6

document type name	{iso identified-organization oiw(14) ftamsig(5) 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) 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 --  NBS-AS1

Datatype2 ::= Node-Descriptor-Data-Element
```

A.5.3 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.

A.5.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

A.5.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

A.5.6 Abbreviations

FTAM File Transfer, Access and Management

A.5.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely zero or one data unit 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) 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 annex C.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.

For datatype 1, 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.

A.5.8 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.

A.5.9 Definition of transfer

A.5.9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 9, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 9, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

A.5.9.2 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 established to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context established 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).

A.5.9.3 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.

A.5.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 9 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

A.5.11 ASE specific specifications for FTAM

A.5.11.1 Structural Simplification

This structural 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.

A.5.11.2 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.

A.5.11.3 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

A.6.1 Entry number: NBS-7

A.6.2 Information objects

Table 10 - Information objects in NBS-7

document type name	{iso identified-organization oiw(14) ftamsig(5) 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) 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) constraint-set(4) nbs ordered-flat(1)} "NBS ordered flat constraint set"

file contents:

```
Datatype1 ::= PrimType --  NBS-AS1

Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                         Enter-Subtree-Data-Element }
                         Exit-Subtree-Data-Element }
```

A.6.3 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.

A.6.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

A.6.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

A.6.6 Abbreviations

FTAM File Transfer, Access and Management

A.6.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely zero or one data unit 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 10). These definitions appear in annex B.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 annex C.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.

For datatype 1, 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.

A.6.8 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.

A.6.9 Definition of transfer

A.6.9.1 Datatype definitions

The file consists of data values which are of either:

a) Datatype1 defined in table 10, where the PrimType in the datatype is given by the NBS-AS1 definition;

b) Datatype2 defined in table 10, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

A.6.9.2 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 established to support the abstract syntax name "asname1";

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context established 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).

A.6.9.3 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.

A.6.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 10 for all presentation data values transferred. implementation may optionally support other named transfer syntaxes.

A.6.11 ASE specific specifications for FTAM

A.6.11.1 Structural simplification

This structural 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>.

A.6.11.2 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.

A.6.11.3 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

A.7.1 Entry Number: NBS-8

A.7.2 Information Objects

Table 11 - Information objects in NBS-8

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) indexed-file(8)} "NBS-8 FTAM indexed file"
abstract syntax names: a) name for asname1	{iso identified-organization oiw(14) ftamsig(5) abstract syntax(2) nbs-as1(1)} "NBS abstract syntax AS1"
b) name for asname2	{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 -- NBS-AS1 Datatype2 ::= CHOICE { Node-Descriptor-Data-Element, Enter-Subtree-Data-Element, Exit-Subtree-Data-Element } Datatype3 ::= Primtype -- as defined by the NBS abstract syntax AS1	

A.7.3 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.

A.7.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

A.7.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

A.7.6 Abbreviations

FTAM File Transfer, Access and Management

A.7.7 Document semantics

The document consists of zero, one or more file access data units. Each FADU contains precisely one data unit 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 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 annex C.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.

For Datatype1 and Datatype3, 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 annex C.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 12.

Table 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 < KeyPosition> parameter.

KeyPosition = 0 implies the key is not part of the data

position > 0 specifies the actual data element in the data unit.

A.7.8 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.

A.7.9 Definition of transfer

A.7.9.1 Datatype definitions

The file consists of data values which are of

a) Datatype1 defined in table 11, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 11, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3 defined in table 11, which specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where user-coded is defined as EXTERNAL.

A.7.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

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 established to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2"; or

c) a value of "Datatype3" carrying a key. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1."

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).

A.7.9.3 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.

A.7.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 11 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

A.7.11 ASE specific specifications for FTAM

A.7.11.1 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.

A.7.11.2 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.

A.7.11.3 The INSERT operation

When the <INSERT> operation is applied the transferred material shall be the series of FADUs 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.

A.7.11.4 The EXTEND operation

This operation is excluded for use with this document type.

A.8 NBS-9 File directory file

A.8.1 Entry Number: NBS-9

A.8.2 Information objects

Table 13 - Information objects in NBS-9

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) file-directory(9)} "NBS-9 FTAM file directory file"
abstract syntax names:	{iso identified-organization oiw(14) ftamsig(5) abstract-syntax(2) nbs-as2(2)} "NBS file directory entry abstract syntax"
transfer syntax names:	

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) }
```

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 annex C,
                --C.4  of this document
```

A.8.3 Scope and field of application

This document defines the contents of a file for transfer (not for storage) using FTAM.

A.8.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management.

A.8.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1

A.8.6 Abbreviations

FTAM File Transfer, Access and Management.

A.8.7 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.

A.8.8 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).

A.8.9 Definition of transfer

A.8.9.1 Datatype definition

The file consists of zero or more values of Datatype1 defined in table 13.

A.8.9.2 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 13.

A.8.9.3 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.

A.8.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 13 for all presentation data values transferred. Implementations shall optionally support other named transfer syntaxes.

A.8.11 ASE specific specifications for FTAM

Relaxation is allowed to any bitstring combination of the document type parameter.

Annex B (normative)

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 part 9, B.2.

B.2 NBS ordered flat constraint set definition

B.2.1 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.

B.2.2 Basic constraints

Table 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)	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'.
Write whole file (replace)	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'.

B.2.3 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.

B.2.4 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).

B.2.5 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 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

Annex C (normative)

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 part 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 part 9, C.4.

C.3 Abstract Syntax NBS-AS1 definition

Abstract syntax name: {iso identified-organization oiw(14) ftamsig(5) 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) 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.

Annex D (informative)

FTAM-1 Document Type Tutorial

D.1 Introduction

This annex is informative. It does not specify any additional requirements.

The purpose of this tutorial is to describe methods to convey lines of text in a FTAM-1 document type.

ISO 8571-2 defines a number of document types for files. One of these document types is FTAM-1. ISO defines the FTAM-1 document type for usage with files that contain unstructured text. A file that has a document type of FTAM-1 consists of one FADU that consists of zero or more character strings. In order to reduce ambiguities it is useful to assume that one string corresponds to one Data Element.

FTAM-1 document type parameters are defined in ISO 8571-2 clause B.1. These parameters are used to define:

the allowed character sets that may be contained in the strings (universal-class-number);

the maximum allowed length of a string (maximum-string-length);

the significance of the boundaries of string (string-significance)

D.2 Document type Parameters

D.2.1 Universal-Class-Number

The universal-class-number parameter determines the character sets that are allowed to be used in a FTAM-1 file. The values of the universal-class-number parameter are ASN.1 types whose definition can be found in ISO 8824. For example, GraphicString, IA5String, and GeneralString are some ASN.1 universal types. The important thing for this discussion is that some string classes allow only graphic characters to be used while other string classes allow both graphic and control characters to be used. (Control characters include "format effector" characters such as carriage return <CR> and line feed <LF>).

D.2.2 Maximum-String-Length

The maximum-string-length parameter determines the maximum number of characters allowed in a string of the FTAM-1 file. It does not determine the maximum number of octets allowed in the string.

GeneralStrings illustrate how the number of octets in a string can differ from the number of characters in a string. GeneralStrings can contain escape sequences that are used for purposes such as invoking different

character sets. An escape sequence is considered to be a bit string, not a character string. Therefore, the combined length of any escape sequences contained in a GeneralString contributes to the number of octets in the GeneralString but does not contribute to the number of characters in the GeneralString.

The length value of the ASN.1 encoding of a character string always reflects the number of octets in the character string. This value will always be greater than or equal to the number of characters in the string. The ASN.1 string must be processed to determine the actual number of characters in the string.

OIW FTAM Phase 2 agreements state that a conformant FTAM implementation must support a maximum-string-length parameter of at least 134 for a FTAM-1 file (see part 9 clause 10). There is no minimum requirement for maximum-string-length in the FTAM phase 3 agreements. The minimum requirement implies that a minimally conformant OIW FTAM responding implementation will not accept a FTAM-1 file whose actual maximum-string-length parameter has a value greater than 134. The relaxation rules for FTAM-1 files allow a FTAM-1 file to be opened for read using a maximum-string-length parameter that is greater than or equal to the value of the maximum-string-length file attribute actually associated with the file, a smaller value is not allowed (see ISO 8571-2 B.1 clause 11.1.1.2). This implies that a minimally conformant OIW FTAM initiating implementation can not read a FTAM-1 file whose actual string length parameter has a value greater than 134.

To increase interoperability, a sending FTAM system should be able to divide a file with string-significance of not-significant into strings of no more than 134 characters. A receiving FTAM system should be able to use the strings to form the file which was sent. If a file has a maximum-string-length associated with it that is greater than 134 interworking will not be possible with a minimally conformant system.

D.2.3 String-Significance

The string-significance parameter determines the significance of the character strings (semantics of string boundaries). Fixed string-significance means that each string contains exactly the number of characters defined by the maximum-string-length parameter. Variable string-significance means that the length of each string is less than or equal to the maximum-string-length parameter. When string-significance is fixed, then maximum-string-length must be present. For string-significance of fixed or variable the boundaries of the character strings are preserved and contribute to the document's semantic. A value of not-significant means that the length of each string is less than or equal to the maximum-string-length parameter and that the boundaries of the character strings are not necessarily preserved when the file is stored and do not contribute to the document's semantics. In this case, string-significance may not be maintained, thus the sender entity explicitly declares that string boundaries have no meaning.

Note the OIW FTAM Phase 2 agreements require the support of only the not-significant value for string-significance. Fixed and variable string-significance are outside the scope of the Phase 2 agreements, but are required in the Phase 3 agreements.

It is in the area of not-significant strings where most interoperability problems have occurred.

> **NOTE** - the difference between variable significance and not-significant significance. If a file has a significance of fixed or variable, it is the responsibility of any storer of the file to "remember" where the boundaries of each character string are located within the file. The storer of a file with a significance of not-significant has no such responsibility. For example, when working with a not-significant file, the sending application may find that 512 byte chunks of data is convenient and useful. The 512 byte size may have no relation to the file layout, but is

easy to read from disk.

D.3 New Line Function

When a sequence of characters are being displayed on a character imaging device, e.g., printer or video display terminal the term "new line function" is used to mean the repositioning of the current character display position one row down and back to column one. A new line function may be implemented in a variety of ways. A UNIX system implements the new line function with a <LF> character (sometimes called <NL>). A MS-DOS system implements the new line function with a <CR><LF> character sequence. A typical word processor will implement a new line function as a "wrap around" function that depends upon a defined page width. A record oriented file system may interpret an end of record condition as implying a new line function.

ISO suggests (see ISO 646 clause 4.1.2.2) that a new line function be accomplished with a <CR><LF> combination. If there is a prior arrangement,e.g., a bilateral agreement, between a sender and a receiver, and only in this case, may a vertical format effector, i.e., a <LF> be used to accomplish a new line function. The OIW FTAM agreements contain no such prior arrangement (see OIW Part 9 clause 10.1.2).

It is strongly suggested that files being sent to a remote FTAM implementation represent the local new line function as a <CR><LF> pair and files received from a remote FTAM implementation have <CR><LF> pairs converted to the local new line function. See D.5 for the reasons for this suggestion.

It is important to realize that a new line function represents a display positioning function and it does not represent anything more that. A new line function is not intended to act as either a string terminator or a string separator.

D.4 Character Strings Versus Lines

A line of characters is generally considered to be a sequence of graphic characters followed by a new line function (or possibly by an end of line condition).

A character string is simply that, a string of characters from one or more character sets. Characters within a string come from allowed character sets. It is the "universal-class-number" parameter defined in ISO 8571-2 B.1 that determines which character sets may be used to compose a string. For example, a GraphicString consists of characters from any graphic character set but may not contain characters from a control character set (it can not contain format effectors); a GeneralString consists of characters from any graphic character set and characters from any control character set (it can contain format effectors).

Text files will be transferred using the Document type FTAM-1. The supported character sets and their recommended line delimiters are:

IA5String	(line boundaries via format effectors, preferably <CR><LF>)
GeneralString	(i.e. ISO 646 International Reference Version and ISO 8859-1. Line boundaries via format effectors. preferably <CR><LF>)
VisibleString	(IA5 String without control characters, line boundaries via Data Element boundaries)

GraphicString (i.e. ISO 646 International Reference Version without control characters and ISO 8859-1, line boundaries via Data Element boundaries)

NOTE - A string is really a language (programming or otherwise) concept. File systems generally have no concept of a string, although a file system, especially a record oriented file system may have some concept of a line.

The standard gives no relation between character string and a line of characters. A character string may contain a portion of a line of characters or it may contain multiple lines of characters. A character string can contain zero, one, or many <CR><LF> pairs. For those character sets which include format effectors, a character string may or may not end with a <CR><LF> pair. In fact, an entire file of character strings may not contain a single <CR><LF> pair, even when those characters are allowed to be used in the character strings.

The following figure is an example of how lines of text could be conveyed using IA5String or GeneralString with string-significance of not-significant.

String-1	String-2		String-3	String-4	String-5
Line-1 <CR><LF>	Line-2 <CR><LF>	Line-3 <CR><LF>	Line-4 <CR><LF>		Line-5 <CR><LF>

The following figure is an example of how lines of text could be conveyed using VisibleString or GraphicString with string-significance of fixed or variable.

String-1	String-2	String-3	String-4	String-5
Line-1	Line-2	Line-3	Line-4	Line-5

D.5 Mapping FTAM-1 Files to Real Files

The lack of equivalence between a line of characters and a character string can cause implementation problems. It is common for a record oriented file system to store a line of characters as a record. How does such a system decide how large a record to allocate for a line of characters? A line of characters may be contained in a part of one string, one or more strings, or it may actually consist of an entire file. How does such a system identify the end of a line (record)? It must scan the string for a <CR><LF> pair (or end of transmission) and probably remove the <CR><LF> before storing a record. What happens if the line is bigger than the size of the record allocated? The system would likely break the string and store it in the available record size. In this case, the FTAM-1 document type should not be used to transfer this file when the sender is not sensitive to the receiver's limitations.

Another problem can occur when a system whose new line function is implemented by a <CR><LF> pair sends a file to a system whose new line function is implemented by <LF>. For example, a MS-DOS system

could send a file that contains <CR><LF> pairs and also contains single <LF> characters to a UNIX system. The UNIX system would likely translate both <CR><LF> and <LF> to UNIX new line functions, i.e. a <LF>. In this case, the FTAM-1 document type should not be used to transfer this type of file.

D.6 Printing or Displaying a File without Format Effectors

There is no relation between a character string and a line of characters (see ISO 8571-2 B.1 clause 7) except when character strings that come from character sets that do not contain format effector characters (for example, VisibleStrings and GraphicStrings) are transferred to a device such as a printer. In this case the end of a string implies the invocation of the device's new line function. This means that, in this case, a string is equivalent to a line.

The rendition of such a file made of character strings belonging to a set that does not contain format effector characters (for example, VisibleString, and GraphicString) to be transferred first to disk and then to a character imaging device might not be equivalent to the rendition of the same file transferred directly to a character imaging device.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 10 - FTAM Phase 3

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Joe Mohen, Proginet**
SIG Editor: **Larry Friedman, Digital Equipment Corporation**

Foreword

This part of the Stable Implementation Agreements was prepared by the File Transfer, Access and Management Special Interest Group (FTAM SIG) of the Open Systems Environment Implementors' Workshop (OIW). See Procedures Manual for Workshop charter.

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject. There is no significant technical change from this text as previously given. References to Part 9 are made in this part.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

Annex C (normative)

List of Tables

Part 10 - 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."

0 Introduction

This clause 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 through Version 4, Edition 1 (NIST Special Publication 500-183, this document part 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 clauses specify the functionality of OIW FTAM Phase 3:

a) Clauses 1 and 8 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;

b) Annex 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 8571-5;

c) Annex 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 clause in the NIST OIW Stable Agreements where the respective object is being defined;

d) Annexes C, D, and E provide definitions for additional document types, constraint sets and abstract syntaxes;

1 Scope

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;

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).

2 Normative References

Amendments and corrigenda to the base standards referenced: See annex G for a complete list of these documents.

ISO 8571-1: 1988(E), Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 1: General Introduction

ISO 8571-2: 1988(E), Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 2: Virtual Filestore Definition

ISO 8571-3: 1988(E), Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 3: The File Service Definition

ISO 8571-4: 1988(E), Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 4: File Protocol Specification

3 Status

These FTAM Phase 3 Agreements were completed December 15, 1989. No further enhancements will be made to this version (see also next clause 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 1 - Phase 2/Phase 3 Interworking

Additional requirements in FTAM phase 3	Backward compatibility to FTAM phase 2
FTAM-1: GraphicString,VisibleString FTAM-2: VisibleString 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 1 - 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 parameter for Initiator	
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 functional units	

Table 1 - Phase 2/Phase 3 Interworking (concluded)

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 requirements for maximum-string-length parameters for document types	if a Phase 3 system stays below this minimum requirement

4 Errata

Table 2 - List of Errata

No. of errata	Type	Referenced document	Clause	Description
CP 3/91-1	Editorial	NIST-SP 500-183	All	Update to ISO style. General formatting and error corrections. Alignment with the wording of the ISP. Consistent naming conventions.
CP 6/91-1	Editorial	NIST-SP 500-183	8.6.1 A.13.9.1.2 A.13.9.1.3 A.13.9.1.4	Previous errata changed the Profile Requirements List (PRL) support of Concurrency Control from "m" to "o". This change was not reflected. Alignment with the ISP.

No. of errata	Type	Referenced document	Clause	Description
CP 9/91-1	Editorial	NIST SP 500-183	Table 4	Include "FTAM" in object descriptor for consistency with other OIW FTAM objects.
CP 9/91-2			Table 5	Add definition for Datatype3
CP 9/91-2			Table 8	Delete last line of Write Whole File [previous change incomplete].
CP 9/91-3			Clause 2	Add reference to corrigenda.
CP 9/91-4			A.12.16.1 A.12.16.5 A.12.17.1 A.12.17.5	Support level from "o" to "m". Add note that must support at least one action. Add note about supporting at least one optional FU.
CP 9/91-5			A.13.6.1 A.13.6.2	Change to spelling of ASN.1 text types.
CP 9/91-6			C.2.7 C.2.9.1 C.2.9.2	Changes to add Datatype3 to text descriptions
CP 9/91-7			C.1.11.1 C.2.11.1 C.3.11.1	"Structural Simplification" to "Simplification"
CP 9/91-8			E.1 E.2 E.3	Changed "will" to "can"
CP 9/91-9			Annex B	Added Editors note of intention to remove object definitions when the ISP is published.
CP 9/91-10			Added Annex G	New annex to list corrigenda

5 Conformance

In addition to the specific requirements specified in the following subclauses, conformance to this Phase 3 specification requires

conformance to ISO 8571: 1988

conformance to Phase 2 FTAM, unless specified otherwise in this part 10.

The access Profiles A1.3 and A2.3 do not include the requirement for transferring files using the File Transfer service class.

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.

7 Filestore Agreements

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 Annex C) are defined for optional support.

Table 2 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 3 - Implementation Profiles and Document Types - 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	GraphicString (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, T3.3, A1.3, A2.3	FTAM-3	-	'not-significant'
[T2.3], [T3.3], [A1.3], [A2.3]	FTAM-4	-	'not-significant'

Table 3 - Implementation Profiles and Document Types - NBS-6 Through NBS-11 (continued)

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 3 - Implementation Profiles and Document Types - NBS-12 (concluded)

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 GO character set and the ISO 646, IRV CO set.

4 The minimum level of support for Graphic String is the ISO 646, IRV GO character set and the 8859-1 GO and G1 sets.

5 The minimum level of support for General String is the ISO 646, IRV GO character set and the 8859-1 GO and G1 sets, and ISO 646, IRV CO set.

6 If the Character-Set parameter is absent, the following defaults apply:

Universal-class-number		Default registration numbers CO	GO	G1
IA5String	[22]	1	2	-
GraphicString	[25]	-	2	-
VisibleString	[26]	-	2	-
GeneralString	[27]	1	2	-

Registration number	Content	Escape Sequence
1	CO 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

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.

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.

8 Protocol Agreements

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 Annex 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.

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.

8.3 Implementation Information Parameter

In addition to the Agreements as specified for FTAM Phase 2, part 9 clause 12 , the following value is defined

NBS-Phase3.

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.

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.

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.

8.5.2 Parameters for Error Recovery

The following information is given:

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 functional unit is implemented. The Basic FERPM should wait at least the amount of time as given by the <suggested delay> term before attempting to recover.

8.6 Concurrency Control

8.6.1 Concurrency Control to whole file

If <concurrency control> parameters are supported, 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.

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.

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.

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.

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.

Annex A (normative)

Profile Requirements List

Editor's Note - The page numbering of the PICs tables may not be aligned with the text of this document. The reason for this problem is that the PICs tables are coded using a different wordprocessor. The tables are being converted, but until this is completed the page numbering, and format of the tables may be aligned with the text of this document.

In the event of a discrepancy becomming apparent in the body of these agreements and the tables in this annex, this annex is to take precedence.

Editor's Note - Delete lines A.13.9.1.2, A.13.9.1.3, A.13.9.1.4, when the PICS tables are converted to WordPerfect Version 5.1 format.

Editor's Note - Change table A.5 to reference Annex G. See ISO/IEC ISP 10607-4:1990 A.5. When Annex A is converted to Wordperfect V5.1.

Editor's Note - A.12.16.1, A.12.16.5, A.12.17.1, and A.12.17.5 replace the "o" with "m" in the A1.3 column. Add a note to tables A.12.16 and A.12.17 "For the profile A1.3, the support of at least one of insert, replace, or extend is required." Also add a note to tables A.12.16 and A.12.17 " For profiles T1.3 and T2.3, the support of at least one of read, insert, replace or extend is required." When Annex A is converted to WordPerfect V5.1.

Editor's Note - A.13.6.1, and A.13.6.2 change parameter names to "Universal time," "Generalized time," "IA5String," "Boolean," "Bit," "Integer." When Annex A is converted to WordPerfect V5.1.

Annex A

(normative)

Profile Requirements List for NIST OIW FTAM Phase 3

A.0 Introduction

This annex 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 annex 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 annex 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 annex is a summary of all definitions of FTAM Phase 3 as they appear in the Stable Implementation Agreements for OSI Protocols, Version 4 Edition 1, December 1990, parts 9 and 10.

A.0.1 Conformance requirement of Base Standards

The D-column of clauses 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 clauses A.1 to A.13. The following convention is applied for this purpose :

- a 'PROFILES' column is valid for all Profiles T1.3, T2.3, A1.3 and M1.3
- 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

- a single value in these columns applies to the Initiator as well as to the Responder role of an implementation
- 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.

Section 1

A.1 (void)

A.2 (void)

Section 2: General ISO 8571 Detail

A.3 ISO 8571 Protocol versions

1	FTAM protocol version number(s)	**version-1**

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 part 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 part 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 clause 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	File availability	f	m	
9	Future filesize	f	m	see part 9 17.9

NOTE - An initiator shall not partially support attributes

	STORAGE GROUP (RESPONDER)	D	PROFILES R full	R partial	RANGE OF VALUES
10	Storage account	p	o	o	
11	Date and time of creation	p	o	o	
12	Date and time of last modification	p	o	o	
13	Date and time of last read access	p	o	o	
14	Date and time of last attribute modification	p	o	o	
15	Identity of creator	p	o	o	
16	Identity of last modifier	p	o	o	
17	Identity of last reader	p	o	o	
18	Identity of last attribute modifier	p	o	o	
19	File availability	p	m	x	
20	Filesize	p	m	x	see part 9 17.9
21	Future filesize	p	o	o	see part 9 17.9

	SECURITY GROUP (INITIATOR)	D	PROFILES I full	RANGE OF VALUES
22	Access control	f	m	see A.12.2
23	Legal qualifications	f	m	

NOTE - An initiator shall not partially support attributes

	SECURITY GROUP (RESPONDER)	D	PROFILES R full	R partial	RANGE OF VALUES
24	Access control	p	m	x	see A.12.2, part 9 9.2
25	Legal qualifications	p	o	o	

A.10.2.3 Filename detail

See part 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 ACTION	CONSTRAINT SET unstructured D	T1.3
1	Locate	———	
2	Read	o	o
3	Insert	———	
4	Replace	o	o
5	Extend	o	o
6	Erase	o	i

	CONSTRAINT SET											
	RESPONDER	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

	CONSTRAINT SET											
	RESPONDER	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	CONSTRAINT SET unstructured	
	ACCESS CONTEXT	D	T1.3
1	US	—	—
2	UA	o	m
3	FS	—	—
4	FL	—	—
5	FA	—	—
6	HN	—	—
7	HA	—	—

	RESPONDER	CONSTRAINT SET unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACCESS CONTEXT	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3
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 / ACCESS CONTEXT	CONSTRAINT SET: unstructured D	unstructured A1.3	sequential flat D	sequential flat A1.3	ordered flat D	ordered flat A1.3	ordered flat with unique names D	ordered flat with unique names A1.3	NBS ordered flat D	NBS ordered flat A1.3	NBS random access D	NBS random access 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 A.12.15.

Section 5 : File Protocol Detail

A.11 File protocol

See part 9 clauses 5.1 - 5.3 and 17

Subclauses 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 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								
		Ker-nel	Read	Write	Ac-cess	LFM	EFM	Grou-ping	Reco-very	Re-start
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 (e.g., 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, part 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 part 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, part 9 18.4
13	Initiator identity	o	-	m	-	see 8.8, part 9 16.1 and 18.4
14	Account	o	-	o	-	see 8.8, part 9 18.4
15	Filestore password	o	-	m	-	see A.12.11, 8.8, part 9 16.1
16	Diagnostic	-	o	-	m	see A.12.6, 8.5.2, part 9 13
17	Checkpoint window	m	m	m	m	see note 2, 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 accordance with 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 part 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, part 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, 8.5.2, part 9 13

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, part 9 17.9
5	Requested access	m	-	m	-	see A.12.16
6	Access passwords	o	-	m	-	see 8.8, part 9 16.2
7	Concurrency control	o	-	o	-	see A.12.13, 8.6.1
8	Shared ASE information	o	o	I	I	see part 9 5.8
9	Account	o	-	o	-	see 8.8, part 9 18.4
10	Diagnostic	-	o	-	m	see A.12.6, part 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 part 9 5.8
5	Diagnostic	-	o	-	m	see A.12.6, part 9 13

A.11.8 File creation

		D		PROFILES		
		I	R	I	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 8571
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, part 9 10.2.2,17.9
6	Create password	o	-	m	-	see A.12.12, 8.7, 8.8, part 9 16.2
7	Requested access	m	-	m	-	see A.12.16
8	Access passwords	o	-	m	-	see 8.8, part 9 16.2
9	Concurrency control	o	-	o	-	see A.12.13, 8.6.1
10	Shared ASE information	o	o	I	I	see part 9 5.8
11	Account	o	-	o	-	see 8.8, part 9 18.4
12	Diagnostic	-	o	-	m	see A.12.6, part 9 13

A.11.9 File deletion

		D		PROFILES		
		I	R	I	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 8571
3	Shared ASE information	o	o	I	I	
4	Charging	-	o	-	o	see A.12.10
5	Diagnostic	-	o	-	m	see A.12.6, part 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, part 9 17.9
5	Diagnostic	-	o	-	m	see A.12.6, part 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, part 9 17.9
4	Diagnostic	-	o	i	-	m	see A.12.6, part 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	o o	i	see A.12.13, 8.6.1
7	Shared ASE information	o o	i i	i	see part 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, part 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 note

NOTE - The values depend upon the functional units implemented in ISO 8823.

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 part 9 5.8
4	Diagnostic	o	m	i	see A.12.6, part 9 13

A.11.14 Beginning of grouping

		D		T1.3, T2.3		A1.3		
		I	R	I	R	I	R	
1	**F-BEGIN-GROUP PDU**	c	c	m	m	o	o	see A.11, A.12.5
	FIELD NAME							**RANGE OF VALUES OR REFERENCE**
2	Threshold	m	-	m	-	m	-	

A.11.15 End of grouping

		D	T1.3, T2.3	A1.3	
1	**F-END-GROUP PDU**	c	m	o	see A.11, A.12.5
		The F-END-GROUP PDU carries no fields.			

A.11.16 Regime recovery

See 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 clause 9
6	Requested access	m	-	m	-	i	see A.12.16
7	Access passwords	o	-	m	-	i	see 8.8, part 9 16.2
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, 8.5.2, part 9 13
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 part 9 17.9
4	FADU lock	o	-	i	o	-	i	see A.12.14
5	Diagnostic	-	o	i	-	m	i	see A.12.6, part 9 13

A.11.18 Erase file access data unit

		D I	D R	T1.3, T2.3	A1.3 I	A1.3 R	M1.3	
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 857
3	FADU identity	m	-	i	m	-	i	see part 9 17.9
4	Diagnostic	-	o	i	-	m	i	see A.12.6, part 9 13

A.11.19 Read bulk data

		D I	D R	T1.3, T2.3 I	T1.3, T2.3 R	A1.3 I	A1.3 R	M1.3	
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 part 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 I	D R	T1.3, T2.3 I	T1.3, T2.3 R	A1.3 I	A1.3 R	M1.3	
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 part 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, part 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 part 9 5.8
4	Diagnostic	-	o	-	m	I	see A.12.6, part 9 13

A.11.23 Cancel data transfer

See part 9 clause 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 part 9 5.8
4	Diagnostic	o	m	I	see A.12.6, part 9 13

A.11.23.1 F-CANCEL mapping

See part 9 clauses 11 and 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 8.3, part 9 5.6 and 12

A.12.2 Access control detail

See 7.3, part 9 9.2

	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, 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	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 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	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 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 part 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 part 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 part 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 5.1, 8.1, part 9 table 7

		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 8.1, 8.2, part 9 table 7

	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	o	i
17	Grouping	o	o
18	FADU Locking	o	o
19	Recovery	o	o
20	Restart	o	o

see 8.6.2

See 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 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 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 part 9 clause 13).

A.12.7 Contents type detail

A.12.7.1 Contents type list parameter

See part 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 part 9 10.2.3

		D	PROFILES	REFERENCE
1	document type specifications	o	m	see part 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 8.5.2

A.12.9 Details of shared ASE information

(Void)

A.12.10 Details of charging

See part 9 5.8 and 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 part 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 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
Read	o	o	o	o	o	o	o	o
Insert	o	o	o	o	o	o	o	o
Replace	o	o	o	o	o	o	o	o
Extend	o	o	o	o	o	o	o	o
Erase	o	I	o	I	o	I	o	I
Read attrib	o	o	o	o	o	o	o	o
Change attrib	o	I	o	I	o	I	o	I
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
Read	o	o	o	o	o	o	o	o
Insert	o	o	o	o	o	o	o	o
Replace	o	o	o	o	o	o	o	o
Extend	o	o	o	o	o	o	o	o
Erase	o	o	o	o	o	o	o	o
Read attrib	o	o	o	o	o	o	o	o
Change attrib	o	I	o	I	o	I	o	I
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 8.6.1, part 9 clause 14

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 A.10.5

A.12.16 Requested Access

See part 9 clause 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	i	i	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	i	i	o	i

A.12.18 Recovery mode

See 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 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 part 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 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 7.2, part 9 clause 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	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 constr 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 part 9 clause 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 constr 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 7.1)

A.13.1.1 Universal class number parameter (See part 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 part 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 part 9 10.1.1 and 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 part 9 10.1-3)**

A.13.2 FTAM-2 (see 7.1)

A.13.2.1 Universal class number parameter (see part 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 part 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 part 9 10.1.1 and 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 part 9 10.1.1-3)**

A.13.3 FTAM-3

A.13.3.1 String length parameter and string significance parameter combinations (see 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 FTAM-4 (see 7.1)

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

A.13.5 NBS-6

See part 9 tables 2, 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 part 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

A.13.6 NBS-7

See part 9 tables 2, 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 part 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

A.13.7 NBS-8

See part 9 tables 2, 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 part 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

A.13.8 NBS-11

See part 9 tables 2, 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 part 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 7.1)

A.13.9.1 Universal class number parameter (see part 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	see 7.1 table 3(c)
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	see 7.1 table 3(c)

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 part 9 10.1.1 and 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 part 9 10.1.1-3)**

- END OF FTAM PHASE 3 PROFILES REQUIREMENTS LIST -

Annex B (normative)

Register of FTAM Objects

B.1 Introduction

The objects defined in B.2.1 and B.2.2 will be removed from this document after ISO/IEC ISP 10607-2 and ISO/IEC ISP 10607-2/Amd.1 are published. During the period between publishing the ISP and the removal of the definitions from this document, the definitions in the ISP will take precedence over this document.

When the object definitions are removed, clauses B.2.1 and B.2.2 will be changed to point to the ISP.

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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex B clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex C clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex C clause C.2
AP-Title		{nist-adhoc ftam-nil-ap-title(7) }	Dec 15, '89	Stable Agreements Vers. 4, Ed. 1, December '90 NIST SP 500-183 part 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause A.6
NBS-8	NBS-8 FTAM indexed file	{nist-oiw-ftam document-type(5) indexed-file(8) }	March 16, '90	Stable Agreements Vers. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex A clause A.8
	NBS ordered flat constraint set	{nist-oiw-ftam constraint-set(4) nbs-ordered-flat(1) }	March 16, '90	Stable Agreements Vers. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex B clause B.2
NBS-AS1	NBS abstract syntax AS1	{nist-oiw-ftam abstract-syntax(2) nbs-as1(1) }	March 16, '90	Stable Agreements Vers. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex C clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 9, annex C clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex C clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex C clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex C clause C.3
	NBS Random Access	{nist-oiw-ftam constraint-set(4) nbs-random-access(2) }	Dec 15, '89	Stable Agreements Vers. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex D clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex E clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex E clause 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. 4, Ed. 1, December '90 NIST SP 500-183 part 10, annex E clause E.3

Annex C (normative)

Document Types

C.1 NBS-10 Random Binary Access File

C.1.1 Entry Number: NBS-10

C.1.2 Information objects

Table 4 - Information objects in NBS-10

document type name	{iso identified-organization oiw(14) ftamsig(5) document-type(5) random-binary(10)} "NBS-10 FTAM 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 ::= OCTET STRING 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 Random Access Node-Name Abstract Syntax	

C.1.3 Scope and field of application

This document type defines the contents of a file for storage, for transfer and access by FTAM.

C.1.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

C.1.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

C.1.6 Abbreviations

FTAM File Transfer, Access and Management

C.1.7 Document semantics

The document consists of zero, one, or more File Access Data Units. Each FADU contains precisely one data unit which consists of precisely one data element. The data element is made up of one octet. The order of each of these elements is significant. The semantics of the data elements 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 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.

C.1.8 Abstract syntactic structure

The abstract syntactic structure of the document is a series of octets.

C.1.9 Definition of transfer

C.1.9.1 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."

C.1.9.2 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 4.

> **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.

C.1.9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of Data Units within the file.

C.1.10 Transfer syntax

An implementation supporting these document types shall support the transfer syntax generation rules named in table 4 for all presentation data values transferred.

Implementations may optionally support other transfer syntaxes.

C.1.11 ASE Specific Specifications

C.1.11.1 Simplification

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.

C.1.11.2 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.

C.1.11.3 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.

C.1.11.4 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 File With Unique Keys

C.2.1 Entry Number: NBS-11

C.2.2 Information objects

Table 5 - Information objects in NBS-11

document 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) 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  NBS-AS1

Datatype2 ::= CHOICE {
        Node-Descriptor-Data-Element,
        Enter-Subtree-Data-Element,
        Exit-Subtree-Data-Element }
Datatype3 ::= Prim Type -- as defined by the NBS abstract syntax AS1
```

C.2.3 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.

C.2.4 References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

C.2.5 Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

C.2.6 Abbreviations

FTAM File Transfer, Access and Management

C.2.7 Document semantics

The document consists of zero, one, or more File Access Data Units. Each FADU consists of precisely one data unit 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 5). 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 part 9 Annex C, NBS abstract syntax AS1. 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.

For Datatype1 and Datatype3, 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 part 9 Annex C, C.3 of NIST SP 500-183.

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 6.

Table 6 - 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 ASN.1 GeneralString	(1-16)	lexical order
ASN.1 OCTET STRING	(1-16)	lexical order
ASN.1 GeneralizedTime	(1-16)	increasing value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing time value
		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.

C.2.8 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.

C.2.9 Definition of transfer

C.2.9.1 Datatype definitions

The file consists of data values which are of

a) Datatype1 defined in table 5, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in table 5, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU; or

c) Datatype3, defined in Table 5, which specifies the user-coded form of the Node-Name in the FTAM FADU abstract syntax, where user-coded is defined as EXTERNAL.

C.2.9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is

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 established to support the abstract syntax name "asname1" or

b) a value of "Datatype2." All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2."; or

c) a value of "Datatype3" carrying a Key. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1".

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).

C.2.9.3 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.

C.2.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 5 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

C.2.11 ASE Specific Specifications

C.2.11.1 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>.

C.2.11.2 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.

C.2.11.3 The INSERT operation

When the <INSERT> operation is applied, the transferred material shall be the series of FADUs 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.

C.2.11.4 The EXTEND operation

This operation is excluded for use with this document type.

C.2.11.5 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

C.3.1 Entry Number: NBS-12

C.3.2 Information objects

Table 7 - Information objects in NBS-12

document type names	{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 OCTET STRING 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	

C.3.3 Scope and field of application

The document type defines the contents of a file for storage, and for transfer and access by FTAM.

NOTE - Storage refers to apparent storage within the Virtual Filestore.

C.3.4 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.

C.3.5 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" in ISO 8571-2.

C.3.6 Abbreviations

FTAM File Transfer, Access and Management

C.3.7 Document semantics

This document consists of zero, one, or more File Access Data Units. Each FADU consists of precisely one data unit which consists of precisely 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.

C.3.8 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-Text."

C.3.9 Definition of transfer

C.3.9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in table 7, 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 7, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO 8571-FADU.

C.3.9.2 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 7; 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 7.

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.

C.3.9.3 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.

C.3.10 Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in table 7 for all presentation data values transferred.

C.3.11 ASE Specific Specifications

C.3.11.1 Simplification and relaxation

C.3.11.1.1 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.

C.3.11.1.2 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.

C.3.11.1.3 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 relaxation, 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.

C.3.11.1.4 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.

C.3.11.2 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.

C.3.11.3 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.

Annex D (normative)

Constraint Sets

D.1 NBS random access constraint set

Table 8 - Basic constraints in the NBS Random Access Constraint Set

Constraint set descriptor	"NBS random access constraint set"
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 a FADU Identity of "end" and a file access action of "insert," or with a FADU Identity of "begin" and an action of "replace," or with an FADU identity of "node number" and an action of "replace."

Table 9 - 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

D.1.1 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 Node-Name.

D.1.2 Basic constraints

The basic constraints in the NBS Random Access constraint set are given in table 8.

D.1.3 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.

D.1.4 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.

D.1.5 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 9.

Annex E (normative)

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 can 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 4.
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"

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 can be used:

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

Annex F (normative)

Delta Protocol Implementation Conformance Statement (PICS) Pro forma

(Refer to the Working Implementation Agreements.)

Annex G (normative)

Amendments and Corrigenda

Implementations conforming to these agreements shall implement the defect report solutions contained in the following:

FTAM:

a) ISO 8571-1/Cor.1:1990;

b) ISO 8571-2/Cor.1:1990;

c) ISO 8571-3/Cor.1:1990;

d) ISO 8571-4/Cor.1:1990;

e) ISO 8571-3/Cor.2;

f) ISO 8571-4/Cor.2.

Editor's Note - The corrigenda ISO 8571-3/Cor.2, and ISO 8571-4/Cor.2 is to be published. Until it is available, the solutions can be found in the documents ISO/IEC JTC/SC21 N5234 and ISO/IEC JTC1/SC21 N 5235.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 11 - Directory Services Protocols

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Kenneth J. Rossen, SHL Systemhouse**
SIG Editor: **Michael Ransom, NIST**

Foreword

This part of the Stable Implementation Agreements was prepared by the Directory Services Special Interest Group (DSSIG)of the Open systems Environment Implementors' Workshop (OIW). See Procedures Manual for Workshop charter.

Text In thls part has been approved by the Plenary of the above mentioned Workshop. This part replaces the previously existing chapter on Directory Services Protocol.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 11 - Directory Services Protocols

0 Introduction

Editor's Note - the text in this Implementation Agreement will be significantly reorganized in 1993 due to the alignment and submission by Regional Workshops of International Standardized Profiles ISO/IEC pdISP 10615 and 10616. The text in tese pdiSPs, in some cases containing technical changes, will replace substantial segments of the text in this Agreement. In addition, text addressing the forthcoming 1993 edition of the Directory Documents, currently interspersed among sections of this Agreement, will be moved to a new Agreement appearing in Part 28 of this document and expanded. Please refer to the aligned part of the Working Agreements Document for the most recent results of these realignments.

This is an implementation Agreement developed by the implementor's Workshop sponsored by the National institute of Standards and Technology to promote the usefui 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. Whiie 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 clause 2 of this part(hereafter referenced as Directory Documents). Where technical differences between the ISO and CCITT texts of these documents exist (e.g., Transport Requirements) the iSO texts are given precedence.

The Directory User Agents (DUAs) and Directory System Agents (DSAs) provide access to The Directory on behalf of humans and applications such as Message Handiing and Fiie Transfer, Access, and Management. See clause 1 for more information on the model used in the Directory.

This document covers the Directory Access Protocol (DAP), the Directory System Protocol(DSP), and the Directory Information Shadowing Protocol (DiSP) defined in the Directory Documents. A good working knowledge of the Directory Documents is assumed by this chapter. Ali terminology and abbreviations used but not defined in this text may be found in those documents.

1 Scope

Centralized and distributed directories can both be accommodated in this Agreement by the appropriate choice of protocols and pragmatic constraints from those specified. Figure 1 iiiustrates a centraiized directory and figure 2 iiiustrates 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.Bilaterai agreements between a DUA and DSA or DSA and DSA may be impiemented 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 iogical structure of the Directory information Base (DiB) is described in the Directory Documents. The manner in which a iocal portion of the DiB is organized and accessed by its DSA is not in the scope of this agreement.

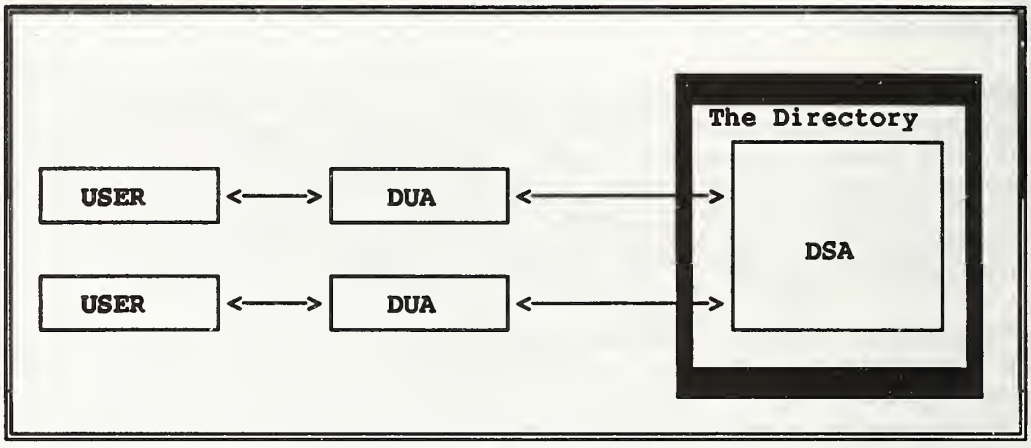


Figure 1 - Centralized directory model

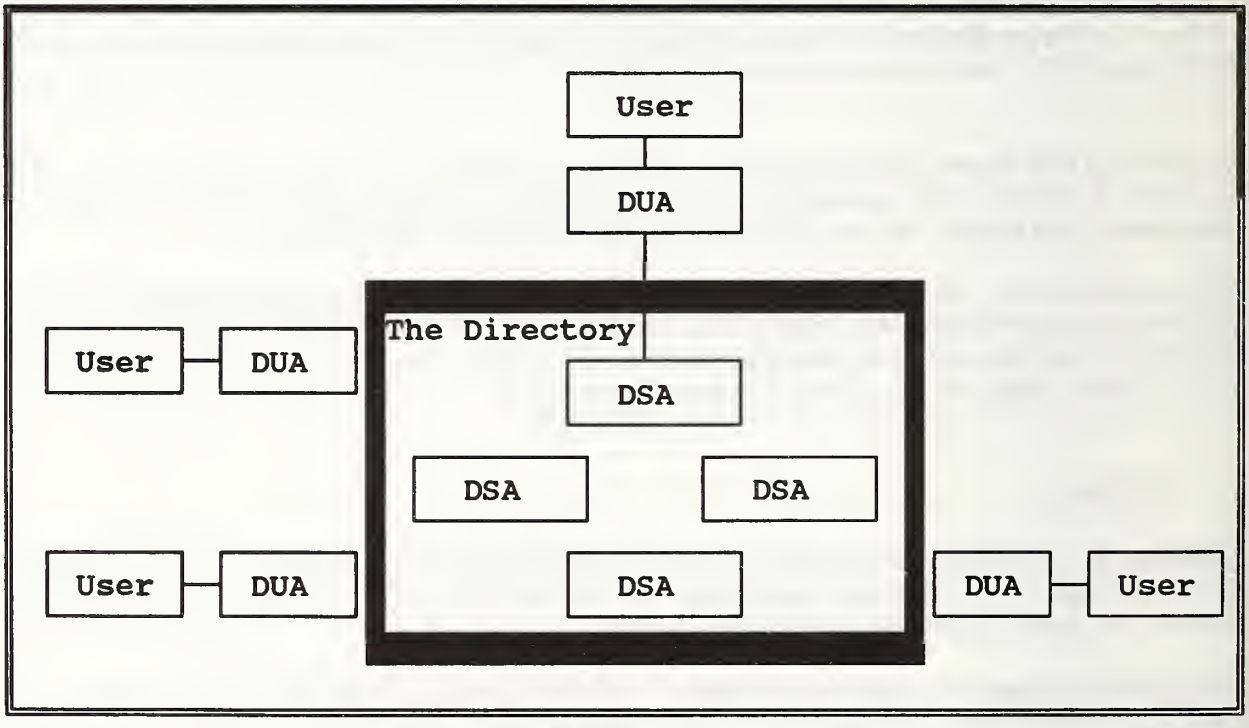


Figure 2 - Distributed directory model

2 References

2.1 Normative References

2.1.1 Base Edition of the Directory Standard

ISO/IEC 9594-1:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 1: Overview of Concepts, Models, and Services.

ISO/IEC 9594-2:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 2: Models.

ISO/IEC 9594-3:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 3: Abstract Service Definition.

ISO/IEC 9594-4:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 4: Procedures for Distributed Operation.

ISO/IEC 9594-5:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 5: Protocol Specifications.

ISO/IEC 9594-6:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 6: Selected Attribute Types.

ISO/IEC 9594-7:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 7: Selected Object Classes.

ISO/IEC 9594-8:1990(E),Information Technology - Open Systems Interconnection - The Directory - Part 8: Authentication Framework.

CCITT Recommendation X.500:1988,The Directory - Overview of concepts, Models and Services.

CCITT Recommendation X.501:1988,The Directory - Models.

CCITT Recommendation X.509:1988,The Directory - Authentication Framework.

CCITT Recommendation X.511:1988,The Directory - Abstract Service Definition.

CCITT Recommendation X.518:1988,The Directory - Procedures for Distributed Operations.

CCITT Recommendation X.519:1988,The Directory - Protocol Specifications.

CCITT Recommendation X.520:1988,The Directory - Selected Attribute Types.

CCITT Recommendation X.521:1988,The Directory - Selected Object Classes.

2.1.2 Extended Edition of the Directory Standard

The following references represent a forthcoming edition of the OSI Directory standard. Alignment to that edition within these agreements is only where explicitly indicated within particular subclauses.

ISO/IEC 9594-1 / DAM-1.2 for Replication, Schema, and Access Control.

ISO/IEC 9594-2 / DAM-1.3 for Access Control.

ISO/IEC 9594-2 / DAM-2.2 for Schema.

ISO/IEC 9594-2 / DAM-3.2 for Replication.

ISO/IEC 9594-3 / DAM-1.3 for Access Control.

ISO/IEC 9594-3 / DAM-2.2 for Replication, Schema, and Enhanced Search.

ISO/IEC 9594-4 / DAM-1.3 for Access Control.

ISO/IEC 9594-4 / DAM-2.2 for Replication, Schema, and Enhanced Search.

ISO/IEC 9594-5 / DAM-1.2 for Replication.

ISO/IEC 9594-6 / DAM-1.2 for Schema.

ISO/IEC 9594-7 / DAM-1.2 for Schema.

ISO/IEC 9594-8 / DAM-1.2 for Access Control.

ISO/IEC 9594-9 / DIS for Replication.

2.2 Informative References

Directory Implementors' Guide, Version ~~5, July 1991~~ 7, November 1992.

3 Status

This version was completed in December 1992.

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.

5 Directory ASEs and Application Contexts

This clause 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:

a) Read ASE

b) Chained ASE

f) Chained Modify ASE

g) Operational Binding Management ASE

c) Search ASE
d) Chained Search ASE
e) Modify ASE
h) Shadow Supplier ASE
i) Shadow Consumer ASE

ROSE and ACSE also form part of the Directory Application Contexts.

The following Application Contexts (ACs) are described in the Directory Document:

a) Directory Access Application AC
b) Directory System AC
c) Directory Operational Binding Management AC
d) Shadow Supplier Initiated AC
e) Shadow Consumer Initiated AC
f) Reliable Shadow Supplier Initated AC
g) Reliable Shadow Consumer Initated AC

6 Schema

There are seven (7) major topics that relate to schema.

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 StateOrProvinceName AVA

as an RDN.

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 thls 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 facllitles permitted by unregistered object classes (viz. Directory Documents, Part 2, clause 9.4.1 (a) and Note).

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.

6.4 Support of Attribute Syntaxes

Suggested methods for the Interpretation of selected Attribute Syntaxes are defined in annex A.

6.5 Naming Contexts

The root of a naming context shall not be an alias entry.

6.6 Common Profiles

This subclause identifies profiles that are commonly useful for various applications while an application-specific profile(s) is identified by the application.

6.6.1 OIW Directory Common Application Directory Profile

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.

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:

a) Top;

b) DSA;

c) Alias.

The following object classes are expected to be generally useful in the creation of the upper portion of the DIT:

a) Country;

b) Locality;

c) Application Process;

d) Organization;

e) OrganizationalUnit.

The following object classes are expected to be generally useful in the creation of DIT leaf entries:

a) Alias;

b) ApplicationProcess;

c) ApplicationEntity;

d) DSA;

e) Device;

f) Group of Names;

g) OrganizationalPerson;

h) OrganizationalRole;

i) ResidentialPerson.

6.6.2 OIW Directory Strong Authentication Directory Profile

6.6.2.1 Other Profiles Supported

This profile is used in conjunction with the OIW Directory Common Application Directory Profile.

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:

a) Strong Authentication User;

b) Certification Authority.

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.

7 Pragmatic Constraints

This clause 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 7.1 and 7.2). The second includes those aspects of pragmatic constraints which are specific to particular attribute types (see 7.3).

7.1 General Constraints

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:

a) T.61;

b) PrintableString;

c) NumericString.

7.1.2 DSP APDU Size

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 2**18 - 1 (i.e., 262,143) octets in size;in this case they may be discarded and an "unwillingToPerform" error reporting service shall be used.

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.

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 3 may be considered as an example by DSAs that implement this Service Control. In figure 3, note that:

a) the DSA maintains three logical queues corresponding to the three priority levels;

b) the DSA Scheduler is separate and distinct from any scheduling function provided by the underlying operating system or control program services;

c) 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;

d) the DSA Scheduler will not preempt a request once it has been passed to the underlying operating system service.

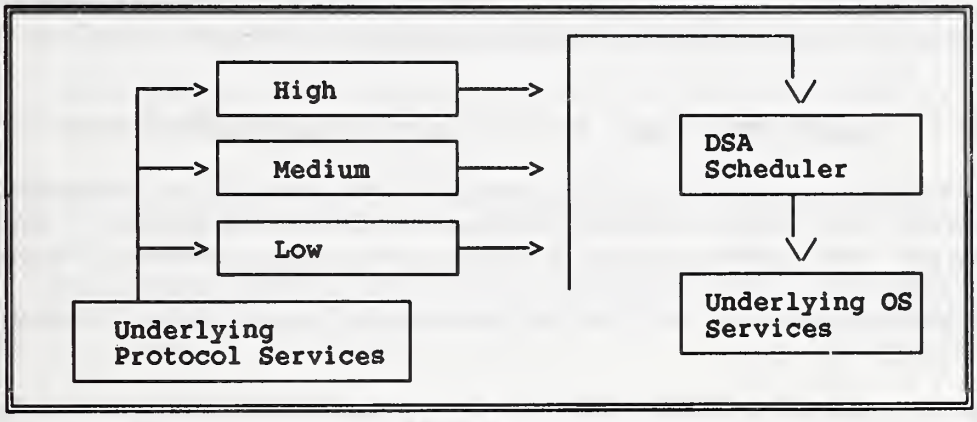


Figure 3 - Logical DSA application environment

7.2 Constraints on Operations

There are no overall constraints upon service arguments or results except those implied in 7.1.2 of this document.

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 Search Argument may return the Service Problem "unwillingToPerform."

7.2.2 Errors

There are no constraints upon any Error service except the APDU size limit as defined in 7.1.2.

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:

a) 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);

b) 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.

7.3 Constraints Relevant to Specific Attribute Types

Table 1 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 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 1 the length constraint for strings is expressed as the number of allowable characters.

In addition to the constraints given in table 1, the following constraints apply to alphabets and integer values:

a) 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;

b) 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.

8 Conformance

The following subclauses 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.

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 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:

a) Some are intended to support human users, some application users;

b) 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;

c) 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;

d) Some DUAs will provide automatic referral services with automatic establishment and release of associations; others will place the burden on the user;

e) Some DUAs will provide a variety of authentication means; others will support no authentication;

f) 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 subclause, 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.

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:

a) *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;

b) *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.

8.3 DSA Conformance Classes

A DSA implementation shall satisfy the conformance requirements as defined in the Directory Documents, Part 5, subclause 9.2,and shall support the "Versions" argument of "Bind."

Per the conformance clause 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 6.3 of this Implementor's Agreement.

Additionally, an implementation conformant to these agreements shall state which of the following conformance classes it implements:

8.3.1 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 Service Error with the "unwillingToPerform" Service Problem.

8.3.2 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.

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:

a) No authentication at all; **(None)**;

b) **Simple Uncorroborated**: identification without verification;

c) **Simple Uncorroborated** authentication with verification: verified identification without a password;

d) **Simple Corroborated** authentication: verified identification with a password; intended to make masquerading difficult;

e) **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 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.

8.5 Directory Service Conformance

The following subclauses 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 subclauses. 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.

8.5.1 Service Conformance

To specify the support for operations and errors, two classifications are used as follows.

8.5.1.1 r: required

The operation shall be implemented and the respective error shall be handled for conformance to these agreements.

For DUAs, *required* means:

a) or ARGUMENT parameters, create the DAP protocol elements to convey the service request to the DSA;

[1] It is the case that some DSAs containing public information may not require authentication.

b) for RESULT and ERROR parameters, accept the DAP protocol elements.

For DSAs, *required* means:

a) for ARGUMENT parameters, accept the protocol elements when received and create the protocol elements when acting as a requesting DSA;

b) 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.

8.5.1.2 n: not required

It is left to implementations as to whether the operation or error is implemented or not.

8.5.2 Protocol Conformance

To specify the support for protocol elements, four classifications are used as follows.

8.5.2.1 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."

8.5.2.2 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.

8.5.2.3 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.

8.5.2.4 O: optional

When generating protocol elements:

a) 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;

b) 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 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.

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 2. For the services in table 2 which are supported, these agreements further require DUAs to support the protocol elements as defined in the DUA column in table 3 (parts 1 - 7).

These agreements require implementations of the DSA to support the Directory Services as defined in the DSA column in table 2. These agreements further require DSAs to support the protocol elements as defined in the DSA column in table 3. Table 3 is listed in seven parts. Note that the requirements for a centralized DSA and a cooperating DSA are different.

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 4. Further, these agreements require DSAs to support the protocol elements as specified in table 5. Table 5 is listed in nine parts.

DSAs are required to support the requestor role for all the services as defined in table 4 if conforming to the chained mode of interaction.

8.8 Digital Signature Protocol Conformance Profile

Table 6 and table 7 provide information on the digital signature protocol conformance profile.

Note that elements in CommonArguments and CommonResults SecurityParameters that are not specified in table 6 and table 7 are covered in the Directory Service Protocol Support (table 5) and Directory Access Protocol Support (table 3).

8.9 Strong Authentication Protocol Conformance Profile

Table 8 and table 9 provide information on the strong authentication protocol conformance profile.

8.10 Subtree Specification Classes

NOTE - This subclause contains agreements on the forthcoming edition of the OSI Directory standard, and is based on the DAM/DIS Directory documents referenced in 2.1 of these agreements.

This profile defines three classes of refinement that may occur in subtree specifications. These classes may be used in describing units of replication for use by DISP or in describing DACDs for use by Basic Access Control:

- Class 0 (Complete Subtree): A subtree definition in which only the base component is specified;
- Class 1 (Chop Subtree): A subtree definition in which only the base and chop components are specified;
- Class 2 (Refined Subtree): A subtree definition in which the base, chop, and specification-filter components are specified.

8.11 Replication Conformance

NOTE - This subclause contains agreements on the forthcoming edition of the OSI Directory standard, and is based on the DAM/DIS Directory Documents referenced in 2.1 of these agreements.

A DSA implementing DISP shall conform to the basic conformance requirements for a DSA as defined in the Directory Documents, part 5, clause 9.2. However, it is not required for such a DSA to be either centralized or distributed as defined by 8.3 of this implementation agreement.

8.11.1 Shadowing Roles

All DSAs implementing DISP shall be capable of acting both as a shadow supplier and as a shadow consumer as defined in the Directory Documents, part 9, clause 3, and as such shall meet conformance requirements stated in part 5, 9.3 and 9.4.

8.11.2 Minimum Shadowing Requirements

Additionally, conformance to this profile requires a minimum as listed below:

a) support for both the directoryShadowConsumerAC application context and the directoryShadowSupplierAC application context;

b) support for an updateMode whose mode choice includes a specification of schedulingParameters;

c) support for schedulingParameters specifications which specify a periodic strategy.

8.11.3 Support for Unit of Replication

This profile defines three classes regarding the level of refinement to be supported by a DSA in the definition of a unit of replication. The provider of a conforming implementation shall state which of the following Unit of Replication Conformance Classes the implementation supports:

a) Class 0 (Basic UnitOfReplication): A DSA conforming to this class shall be capable of shadowing a Unit of Replication with the following characteristics:

1) the area includes a class 0 subtree as defined in 8.10 of these agreements;

2) the area includes a specified knowledgeType (e.g., master, copy, or both).

b) Class 1 (Intermediate UnitOfReplication): A DSA conforming to this class shall fully support the Basic UnitOfReplication and, in addition, shall be capable of shadowing a unit of replication with the following characteristics:

1) the area includes a class 1 subtree as defined in 8.10 of these agreements;

2) the knowledge includes the extendedKnowledge element with value TRUE.

c) Class 2 - (Maximal UnitOfReplication): a DSA conforming to this class shall fully support the Intermediate UnitofReplication and, in addition, shall be capable of shadowing a unit of replication whose specification uses AttributeSelection (including selection on class). Furthermore, a DSA conforming to this class shall be capable of supporting overlapping replicated areas as described in the Directory Documents, part 9, 9.2.5.

NOTES

1 No replication conformance class requires (nor precludes) support for a class 2 subtree specification.

2 Filtering using a specification-filter in the definition of a subtree allows filtering on class when specifying which entries are to be part of the subtree.

3 AttributeSelection is used in shadowing to determine which attributes of the entries in a subtree will be shadowed. ClassAttributeSelection allows choosing specific attributes or all attributes in an class. A list of classes for shadowing can be devised using a sequence of class and classAttributes.

8.12 Recommended Practices for Shadowing

NOTE - This subclause contains agreements on the forthcoming edition of the OSI Directory standard, and is based on the DAM/DIS Directory Documents referenced in 2.1 of these agreements.

8.12.1 APDU Size

In shadowing, updates for an entire Unit of Replication are carred in one APDU. Since the size of such an APDU is application-specific, no pragmatic constraint has been specified in the Directory Documents or implementation Agreements.

Some examples of APDU size implementors can expect would be useful. For instance, an entry size of 2000 octets and a Unit of Replication consisting of 2000 entries would result in a APDU of 4 Megabytes. It is recommended that DSA implementations be capable of supporting an APDU of at least this size. This example does not reflect entries which include large attributes, such as photographic images.

8.12.2 Duplicate Shadow Agreements

Administrators should not allow duplicate shadow agreements between DSAs. Duplicate shadow agreements are those which include the same consumer, supplier, and Unit of Replication.

8.12.3 Consistency Between Supplier and Consumer Information

After an updateShadow operation, the standard does not guarantee consistency between the resulting shadowed information in the consumer DSA and the information in the replicated area in the supplier DSA, since changes may be made during assembly of the APDU containing the shadowed information.

If consistency between the supplier and consumer information is required, the contents of the replicated area in the supplier DSA must not be modified while the APDU is being assembled.

However, the shadowed information must be internally consistent. For example, while the shadowed information is being assembled, changing a distinguished name within the replicated area could lead to internal inconsistency.

8.12.4 Management of Shadowing Agreements Without DOP

For DSAs not supporting the directoryOperationalBindingManagementAC as defined in the Directory Documents, part 5, management of shadowing agreements is by out-of-band means. The results of procedures followed by such DSAs must be the same as if the DSAs had managed the same agreements using the procedure for operational binding management outlined in 8.2 of the Directory Documents, part 9.

For example, when shadowing DSAs arrange to modify the parameters of an existing shadowing agreement, they must revise the AgreementID so that its version component is incremented.

9 Distributed Operations

9.1 Static Requirements

9.1.1 Reference Types

This Functional Standard requires conforming implementations to be able to hold and use reference types as summarised below (and clarified in 9.1.2):

REFERENCE TYPES	HOLDING AND USING CAPABILITY	NOTES
Superior	see note	Non-first-level DSAs shall hold precisely one single superior reference. A First-Level DSA does not hold any superior reference
Subordinate	Mandatory	
Non-specific Subordinate	Optional	
Cross-reference	Mandatory	

9.1.2 Superior References and Root Contexts

9.1.2.1 First-Level DSAs

A DSA conformant to this Functional Standard acting as a first level DSA shall be able to hold and use the root context and, in addition, shall hold as master (i.e., have administrative authority for) at least one naming context immediately subordinate to the root of the DIT. A DSA conforming to this Functional Standard is not, however, required to have the capability of being a first level DSA.

NOTE - The root context never contains any non-specific subordinate references and first level DSAs should not hold such references in respect of the root context to avod circular references.

9.1.2.2 Return-Cross-References

The support of the "return-cross-references" facility, either as requester or as supplier, as defined in the Directory Documents, clause 10.4., is optional.

9.1.3 Support of Application Contexts

All DSAs compliant with this Functional Standard shall support the DirectoryAccessAC or DirectorySystemAC or both.

If a DSA supports DirectorySystemAC, then it must be able to accept a chained request and must be able to generate a referral. The generation of chained requests is optional. See Table 4.

Editor's Note - Table 4, referenced in the above paragraph, is located in the current stable agreements.

9.1.4 DSA-level Security

As a consequence of security policy, a DSA may:

a) refuse associations from any or particular DSAs;

b) refuse invokes on existing associations in which case a SecurityError or ServiceError is returned.

9.1.5 Aliases

DSAs shall be able to carry out name resolution and search continuation for an alias whose dereference points to an entry held outside the DSA (as well as those held inside the DSA).

9.1.6 Authentication for DSA Bind

In the case of simple authentication, if any of the DSAs listed in the trace information is untrusted, the originating user identified by the originator field in the chaining argument should be treated as unauthenticated.

Editor's Note - Use of traceInformation in making security decisions will be a subject of continued discussion and contributions.

9.1.7 Authentication of User Whose Entry Is Held by Another DSA

If a DSA is to be able to carry out simple authentication of a user whose entry is potentially held by some other DSA, the DSA must be able to invoke DSA "compare" and "read" operations to complete authentication by reference to other DSAs. All such DSAs shall support the DirectorySystemAC.

9.2 Dynamic Requirements

9.2.1 Detection of Search Loop

Refer to 7.2.3 of these Agreements.

9.2.2 Generation of 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, ioops which might arise from inconsistent knowledge or from the presence of ailas 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 (Directory Documents, 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.

There is no arbitrary limit on the size of Traceinformation other than that imposed by the maximum APDU size limit.

9.2.3 Integrity of Operation Arguments

Any abstract service operation arguments that are signed must be passed unchanged to the presentation layer. This does not constrain the hoice of transfer syntax used by the presentation layer.

9.2.4 Referrals and Chaining

it is recommended that a DSA which has chained a request act upon any referrals which it receives, rather than returning them to the requestor if the "prefer-chaining" service control is present, unless prevented from doing so byadministrative limitations or service policies.

However, if a DSA which is carrying out a List or a Search operation receives a set of unexplored Continuation References, it shall never pursue these if the result was signed (but was not collated by the DSA with other results), since this will result in duplication. If the result was unsigned, it may act on them (removing them from the consolidated result), or it may pass them back to the invoker of the operation. The DSA can act on the references and remove them if correlated.

If a DSA is unable to establish an association with a remote DSA for the purpose of chaining an operation, then it should return a DSA referral or continuation reference as appropriate.

10 Underlying Services

This section specifies requirements over and above those given in the Directory Documents.

10.1 ROSE

It should be noted that support of "abandon" implies support of operation class 2.

10.2 Session

All directory implementations are required to support Session Version 2.

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 Access Control

Guidelines relating to access control for the base edition of the Directory standard can be found in Annex F of the Directory Documents, Part 2. Specifications for access control in the extended edition of the Directory standard are found in DAM-1.3 to ISO/IEC 9594-2, DAM-1.3 to ISO/IEC 9594-3, and DAM-1.3 to ISO/IEC 9594-4.

12 Test Considerations

This clause 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.

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:

a) Conformance Statement;

b) Distinguished names (e.g., name resolution, equivalence of various forms);

c) Entries and Attributes (e.g., accessibility by operations, compliance with rules);

d) Handling of distributed operations (e.g., naming contexts and knowledge);

e) Schemas:

1) Structure rules (e.g., storage and maintenance of structure and of naming rules);

2) Object classes and sub-classes (e.g., storage and extension of rules for object attributes);

3) Attribute types (e.g., storage and maintenance of syntax classes and rules for multi or single valued attributes);

4) Attribute syntax (e.g., maintenance and support for attribute value testing and matching, to specification for a defined set of attribute types);

f) Operations:

1) all operations;

2) correct function;

3) correct result;

4) correct responses;

g) Aliases (e.g.,correct resolution, error responses);

h) Authentication and Access Control (e.g., limitation of modify access);

i) ROSE (e.g., correct handling of invokes, results, rejects, and invoke ids);

j) ACSE (e.g., association establishment / refusal for invalid application contexts,etc.).

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.

13 Errors

This clause provides clarification of the semantics of various operation errors and implementation guidelines on their usage.

13.1 Permanent vs. Temporary Service Errors

This subclause 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 2**18 - 1 octets.

13.2 Guidelines for Error Handling

NOTE - The error handling tables include symptoms and situations for the DISP as defined in the forthcoming edition of the OSI Directory standard.

13.2.1 Introduction

This subclause provides a recommended mapping of error situations which may be encountered to ROSE Rejects or to the errors provided in the DAP, DSP, and DISP 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:

a) Symptom (i.e.,the manner in which the error was detected);

b) 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).

13.2.2 Symptoms

Table 10 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.

13.2.3 Situations

Table 11 identifies recognized situations within which particular symptoms may give rise to distinct error actions.

13.2.4 Error Actions

Table 13 summarizes specific error actions for each possible combination of symptom and situation. Symptoms are described in 13.2.2 and situations are described in 13.2.3.

Each entry in table 13 corresponds to the symptom in the left-most column and the situation given in the column header. Each entry may specify:

a) a specific error action. The error action is described using the notation shown in table 12;

b) a specific error action and a relevant note. The note will be indicated by a number enclosed in parentheses. The notes can be found at the end of table 13;

c) only a relevant note;

d) a blank (which indicates the corresponding combination of symptom and situation is not meaningful in the context of these Agreements).

The entries in table 13 which specify a specific error action will do so using the notation shown in table 12.

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:

a) Errors indicating attempted breaches of security;

b) Errors indicating local software or hardware malfunction;

c) Errors indicating malfunction or other unacceptable behavior on the part of the invoker of an operation;

d) Errors indicating loss of chaining service by another DSA;

e) Error conditions that would be difficult to diagnose with the level of detail supplied over the protocol;

f) Aborts and other exceptional communications events.

The form and accessibility of any such logs is for further study.

14 Specific Authentication Schemes

This clause identifies authentiction algorithms for use in Directory authentication. Informative text and ASN.1 definitions describing these algorithms appears in part 12 (Security). Use of algorithms other than those cited in this clause or described in the Directory Documents is by bilateral agreement.

14.1 Specific Strong Authentication Schemes

This subclause cites one alternative to the RSA digital signature scheme, the "ElGamal" digital signature scheme. Future contributions may result in other alternatives being added to this subclause.

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 RSA and ElGamal are governed by U.S.A. patent law.

14.1.1 ElGamal

The ElGamal digital signature scheme was originally described by Taher ElGamal in [ELGA85]. Part 12 (Security) of these agreements contains details on the use of ElGamal, including an informative description of the scheme using the notation described in part 8 of the Directory Documents and known constraints on algorithm parameters.

14.1.2 One-Way Hash Functions

This subclause cites alternative one-way hash functions for use in Strong and Protected Simple Authentication. The Security SIG continues to investigate the security of additional one-way hash functions, and the Directory Services SIG will consider the applicability of these hash functions to Directory authentication.

A recent development in this area is the citation by the Security SIG of RSA MD4. In another recent development, the two-pass application of the SNEFRU algorithm was announced by Ralph Merkle to have been broken. Future study of MD4 and other contributions may result in other additions to this subclause.

At the present time, implementors may choose to provide one-way hash functionality based on MD2 or some other scheme appropriate for use in the OSI Directory environment.

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.

14.1.2.2 MD2 Algorithm

MD2 is a one-way hash function and is described in [RFC1115].

14.1.2.3 Use of One-Way Hash Functions in Forming Signatures

MD2 may be used to form digital signatures in conjunction with RSA or ElGamal.

14.1.3 ASN.1 for Strong Authentication Algorithms

This subclause 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}


md2WithElGamal ALGORITHM
    PARAMETER NULL
    ::= {signatureAlgorithm 2}

END -- of Algorithm Object Identifier Definitions
```

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 D.3). Note that the use of MD2 maybe 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 Simple Credentials. Table 14 shows the relationship between SimpleCredentialfields and the elements of protected simple authentication as shown in figure 2 of the Directory Documents, Part 8.

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.

Table 1 - Pragmatic constraints for selected attributes

Attribute Type	Content	Constraints	Primary Source	Notes
Aliased Object Name	Distinguished Name			Note 3
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
Destination Indicator	Printable String	ub-destination-indicator 128	CCITT X.520	
Facsimile Telephone Number	Facsimile Telephone Number	ub-telephone-numb er 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	OIW	About 1 screen full
Locality Name	T.61 or Printable String	ub-locality-name 128	CCITT X.520	
Member	Distinguished Name			Note 3
Object Class	Object Identifier	256 octets	OIW	
Organization Name	T.61 or Printable String	ub-organization-na me 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
Physical Delivery OfficeName	T.61 or Printable String	ub-physical-office-n ame 128	CCITT X.520	

Table 1 - Pragmatic constraints for selected attributes (continued)

Attribute Type	Content	Constraints	Primary Source	Notes
Post Office Box	T.61 or Printable String	ub-post-office-box 40	CCITT X.520	
Postal Address	Postal Address	ub-postal-line6 ub-postal-string30	CCITT X.520	UPU
Postal Code	T.61 or Printable String	ub-postal-code 40	CCITT X.520	
Presentation Address	Presentation Address	224 octets	NIST	Note 2(page ?), ISO 7498.3 & X.200
Registered Address	Postal Address	ub-postal-line6 ub-postal-string30	CCITT X.520	
Role Occupant	Distinguished Name			Note 3
Search_Guide	Guide	256	OIW	
See Also	Distinguished Name			Note 3 (page ?)
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	
Supported Application Context	Object Identifier	256	OIW	
Surname	T.61 or Printable String	ub-surname 64	CCITT X.520	
Telephone Number	Printable String	ub-telephone-numb er 32	CCITT X.520	E.123

Table 1 - Pragmatic constraints for selected attributes (concluded)

Attribute Type	Content	Constraints	Primary Source	Notes
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-number14 ub-country-code4 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

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 2 - Directory access service support

Operations and Errors	Support Classification		Comments
	DUA	DSA	
– 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)	

Table 2 - Directory access service support (concluded)

Operations and Errors	Support Classification		Comments
	DUA	DSA	
SecurityError	(note 4)	r	
ServiceError	(note 4)	r	
UpdateError	(note 4)	4	

NOTES

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, Part 3, clause 9.3.6

3 Centralized DSAs would not generate referrals.

4 See EntryInformationSelection information under Common Data Types (table 3, Part 6)

Table 3 - DAP protocol support

Protocol Element	Support Classification		Comments
	DUA	DSA	
- 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

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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.
- OPERATIONS, ARGUMENTS AND RESULTS -			
- READ OPERATIONS -			
Read			
ReadArgument	M	S	
object	M	S	
selection	O	S	See note 2
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	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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)	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
ListResult	S	G	See note 1 for additional information related to the DSA support classification.
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	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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	
ModifyRDN			
ModifyRDNArgument	M	S	
object	M	S	
newRDN	M	S	
deleteOldRDN	O	S	
CommonArguments	O	G	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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, subclause 12.4.2.2)
type	S	M	
value	S	G	
NameError			
problem	S	M	
matched	S	M	
Referral			
candidate	S	G	
SecurityError			
problem	S	M	
ServiceError			
problem	S	M	
UpdateError			
problem	S	M	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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, subclause 12.4.2.2)
type	S	M	
value	S	G	
NameError			
problem	S	M	
matched	S	M	
Referral			
candidate	S	G	
SecurityError			
problem	S	M	
ServiceError			
problem	S	M	
UpdateError			
problem	S	M	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
- COMMON ARGUMENTS / RESULTS -			
CommonArguments			
ServiceControls	O	S	
SecurityParameters	O	S	See subclause 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 subclause 8.8.
certification-path	O	G	
name	O	G	
time	O	G	
random	O	G	
target	O	G	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
performer	O	G (O)	
aliasDereferenced	O	G	
- 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	

Table 3 - DAP protocol support (continued)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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	
SecurityParameters	O	O	See subclause 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	

Table 3 - DAP protocol support (concluded)

Protocol Element	Support Classification		Comments
	DUA	DSA	
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

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 3, part 6)

Table 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

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 5 - DSP protocol support

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: inappropriate Authentication, invalidCredentials.
DSAUnbind			The DSAUnbind has no arguments.

Table 5 - DSP protocol support (continued)

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	
- OPERATIONS, ARGUMENTS AND RESULTS -			
- CHAINED SEARCH OPERATIONS -			
ChainedList			
ChainingArguments	M	S	

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
ListArgument	M	S	
object	M	D	
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	

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
unexplored	S	G	
unavailableCriticalExt	S	G	
- 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	

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
newRDN	M	S	
deleteOldRDN	G	S	
CommonArguments	G	S	
ChainingResults	S	M	
ModifyRDNResult	S	M	
- ERRORS and PARAMETERS -			
Abandoned			
AbandonFailed			
problem	S	M	
operation	S	M	
AttributeError			Min.1 error (see Directory Documents, part 3, subclause 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	
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 subclause 8.8.

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
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 subclause 8.8.
requestor	S	G	
aliasDereferenced	S	G	
- 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	

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
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	
- 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, subclause 10.4.1
referenceType	G	S	
DomainInfo	O	O	
timeLimit	G	S	
SecurityParameters	O	S	See note 1 regarding the support classification for Request. Also see subclause 8.8
ChainingResults			
Info	O	O	
crossReferences	S	G	

Table 5 - DSP protocol support (continued)

Protocol Element	Support Classification		Comments
	Request	Response	
SecurityParameters	S	O	See note 1 regarding the support classification for Response. Also see subclause 8.8
CrossReference			
contextPrefix	S	M	See Directory Documents, Part 4, subclause 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	
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	

Table 5 - DSP protocol support (concluded)

Protocol Element	Support Classification		Comments
	Request	Response	
tSelector	S	G	
nAddress	S	M	

NOTES

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 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 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 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 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 10 - Error symptoms

Symptom	Description
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 carrying 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).

Table 10 - Error symptoms (continued)

Symptom	Description
E_ARG_SYNTAX	An operation argument either has incorrect ASN.1 encoding or correct ASN.1 encoding, but does not comply to the syntax as defined in the Directory Documents. **NOTES** 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). **NOTES** 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.
E_ATT_SYNTAX	An attribute value either has incorrect ASN.1 encoding or it has correct ASN.1 encoding but does not comply with 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.

Table 10 - Error symptoms (continued)

Symptom	Description
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_CANT_CONSTRUCT	The update to be transmitted exceeds a local size limit.
E_CANT_INCORPORATE	The update received exceeds a local APDU size limit.
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 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_INACTIVE_AGREEMENT	The specified is not currently active.
E_INVALID_AGREEMENT	A valid agreement does not exist with the DSA.
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_MISSED_PREVIOUS	The value received in lastUpdate or is not consistent with the time the recipient DSA understands was the time of the last update.
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_MORE_CURR_UPD_RCD	A consumer DSA processing supplier-initiated updates determines that the update the supplier is attempting to send is older than one the consumer has already received.
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_NO_AGMT_W_THIS_DSA	The receiving DSA has no agreements in place with the sending DSA.

Table 10 - Error symptoms (continued)

Symptom	Description
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.
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_PREVIOUSLY_COORD	A supplier DSA, while processing consumer-initated updates, has received a coordinateShadowUpdate referring to a shadow agreement for which a previous coordinateShadowUpdate has already been received and is still outstanding.
E_PREVIOUSLY_SOLICITED	A supplier DSA, while processing consumer-initated updates, has received a requestShadowUpdate referring to a shadow agreement for which a previous requestShadowUpdate has already been received and is still outstanding.
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_TIMESTAMP_MISMATCH	An unrecoverable timestamp mismatch has been detected.

Table 10 - Error symptoms (continued)

Symptom	Description
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_UNCOORDINATED	A consumer DSA, while processing supplier-initated updates, has received an updateShadow request for which there is no outstanding coordinateShadowUpdate.
E_TOO_MANY_UPDATES	Supplier DSA determines that there are too many updates for incremental refresh and that a full update is required.
E_UNDEFINED_ATT	An unregistered attribute has been encountered.
E_UNRELIABLE_DATA	A DSA has detected internal data inconsistencies.
E_UNSOLICITED	A consumer DSA, while processing consumer-initated updates, has received an updateShadow for which there is no outstanding requestShadowUpdate.
E_UNSUPPORTED_OC	The object class of the entry is not supported as a valid object class for entries within this DSA.
E_UNSUPPORTED_STRAT	The refresh strategy selected is not supported by this DSA.
E_UNUSABLE_DATA	A consumer DSA has decided that the received data is completely unusable due to error.
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 - Error situations

Situation	Description
ABANDON	An Abandon operation is being carried out.
ADD-ENTRY	The entry is being generated.
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.
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.
COMPARE	A Compare operation is being carried out on the entry.
COORDINATE-SHADOW-UPDATE	The shadow consumer has received a coordinateShadowUpdate from the supplier DSA and is evaluating its contents.
LIST	A List operation is being carried out on the entry.
MODIFY-ENTRY	The entry is being modified.
MODIFY-RDN	The RDN is being modified.
NAME-RESOLUTION	Name resolution is being carried out.
READ	The entry is being read.
REMOVE-ENTRY	The entry is being removed.
REQUEST-SHADOW-UPDATE	The supplier DSA is processing a RequestShadowUpdate received from a consumer.
REQUEST-SHADOW-UPDATE-RESULT	The consumer DSA has received a reply to a request for update.
SEARCH-ENTRY	A Search operation is being carried out; the required entry information is being evaluated or acted upon.
SEARCH-FILTER	A Search operation is being carried out; the filter is being evaluated or acted upon.
TRACE-EVALUATION	The trace element is being evaluated for loops.
UPDATE-SHADOW	The consumer DSA has received an UpdateShadow from the supplier and is trying to incorporate the updated information.

Table 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
SH(<qualifier>)	Shadow Error is generated. The qualifier may take on values codified as follows: IAID - Invalid Agreement ID IA - Inactive Agreement IIR - Invalid information received IS - Invalid Sequencing US - Unsupported strategy MP - Missed previous FUR - Full update required UWP - Unwilling to perform UT - Unsuitable timing UAR - Update already received
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

Table 12 - Notation used to describe error actions. (concluded)

Error Action Notation	Meaning
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 DSAEAE - 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 13 - Error actions

Symptom (See Table 10)	Situation (See Table 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) (15, 16)	N(IAS) (15, 16)	A(CV)	A(CV)
E_ATT_OR_VALUE_EXISTS					A(AVE)	A(AVE)
E_ATT_SYNTAX	SC(IC)	(7)	N(IAS) (15, 16)	N(IAS) (15, 16)	A(IAS)	A(IAS)
E_ATT_VALUE	SC(IC)	(7)	N(IAS) (15, 16)	N(IAS) (15, 16)	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_CANT_CONSTRUCT						
E_CANT_INCORPORATE						
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)		

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)					
	Bind-Local	Bind-Remote-Resolution	Name-Resolution	Add-Entry-Name-Resolution	Add-Entry	Modify-Entry
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) (15, 16)	N(IAS) (15, 16)	A(IAS)	A(IAS)
E_INACTIVE_AGREEMENT						
E_INVALID_AGREEMENT						
E_LOOP		S(UA)	S(LD)			
E_MATCH	SC(IC)	SC(IC)	A(IM)	A(IM)		A(IM)
E_MISSED_PREVIOUS						
E_MISSING_AVA					U(NAR)	U(NAR)
E_MISSING_OBJECT_CLASS					U(OCV)	U(OMP)
E_MORE_CURR_UPD_RCD						
E_MULTI_DSA				U(AMD)		
E_NAMING_VIOLATION				U(NV)		
E_NO_AGMT_W_THIS_DSA						
E_NO_ENTRIES_IN_ST						
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_OUTSIDE_UOR						

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)					
	Bind-Local	Bind-Remote-Resolution	Name-Resolution	Add-Entry-Name-Resolution	Add-Entry	Modify-Entry
E_PREVIOUSLY_COORD						
E_REFERENCE		S(UA)	S(IR) (17)			
E_SCOPE			S(OOS)			
E_PREVIOUSLY_SOLICITED						
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_TIMESTAMP_MISMATCH						
E_TOO_MANY_UPDATES						
E_UNABLE_TO_COMPLETE						
E_UNABLE_TO_PROCEED		(2)	(2)			
E_UNCOORDINATED						
E_UNDEFINED_ATT	SC(IC)		(3)	U(NV)	A(UAT)	A(UAT)
E_UNRELIABLE_DATA						
E_UNSOLICITED						
E_UNSUPPORTED_OC					U(OCV)	
E_UNSUPPORTED_STRAT						
E_UNUSABLE_DATA						
E_VERSION	S(UA)					
E_ZERO_VALUES					A(CV)	A(CV)

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 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_CANT_CONSTRUCT					
E_CANT_INCORPORATE					
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					

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)				
	Modify-RDN	Remove-Entry	Read	Compare	Trace-Evaluation
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_INACTIVE_AGREEMENT					
E_INVALID_AGREEMENT					
E_LOOP					
E_MATCH	A(IM)			A(IM)	(7)
E_MISSED_PREVIOUS					
E_MISSING_AVA					
E_MISSING_OBJECT_CLASS					
E_MORE_CURR_UPD_RCD					
E_MULTI_DSA	U(AMD)	U(AMD)			
E_NAMING_VIOLATION	U(NV)				
E_NO_AGMT_W_THIS_DSA					
E_NO_ENTRIES_IN_ST					
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_OUTSIDE_UOR					

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)				
	Modify-RDN	Remove-Entry	Read	Compare	Trace-Evaluation
E_PREVIOUSLY_COORD					
E_REFERENCE					
E_SCOPE					
E_PREVIOUSLY_SOLICITED					
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_TIMESTAMP_MISMATCH					
E_TOO_MANY_UPDATES					
E_UNABLE_TO_COMPLETE					
E_UNABLE_TO_PROCEED					
E_UNCOORDINATED					
E_UNDEFINED_ATT	A(UAT)		A(NSA)(4)	A(NSA)	(7)
E_UNRELIABLE_DATA					
E_UNSOLICITED					
E_UNSUPPORTED_OC					
E_UNSUPPORTED_STRAT					
E_UNUSABLE_DATA					
E_VERSION					
E_ZERO_VALUES					(11)

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 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_CANT_CONSTRUCT				
E_CANT_INCORPORATE				
E_CHAIN				
E_CREDENTIALS				
E_DBE	S(DE)	S(DE)	S(DE)	
E_DIT_STRUCTURE				
E_DN		A(IAS)		
E_DSA		(5)		

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)			
	List (Filter)	Search (Filter)	Search Entry	Abandon
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_INACTIVE_AGREEMENT				
E_INVALID_AGREEMENT				
E_LOOP		(5)		
E_MATCH		A(IM)		
E_MISSED_PREVIOUS				
E_MISSING_AVA				
E_MISSING_OBJECT_CLASS				
E_MORE_CURR_UPD_RCD				
E_MULTI_DSA				
E_NAMING_VIOLATION				
E_NO_AGMT_W_THIS_DSA				
E_NO_ENTRIES_IN_ST				
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_OUTSIDE_UOR				

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)			
	List (Filter)	Search (Filter)	Search Entry	Abandon
E_PREVIOUSLY_COORD				
E_REFERENCE				
E_SCOPE				
E_PREVIOUSLY_SOLICITED				
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_TIMESTAMP_MISMATCH				
E_TOO_MANY_UPDATES				
E_UNABLE_TO_COMPLETE	(B)	S(B)	S(B)	Ab(CA)
E_UNABLE_TO_PROCEED				
E_UNCOORDINATED				
E_UNDEFINED_ATT		(6)	(6)	
E_UNRELIABLE_DATA				
E_UNSOLICITED				
E_UNSUPPORTED_OC				
E_UNSUPPORTED_STRAT				
E_UNUSABLE_DATA				
E_VERSION				
E_ZERO_VALUES				

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)		
	Coordinate Shadow Update	Update Shadow	Request Shadow Update
E_ACCESS			
E_ADMIN_LIMIT			
E_ALIAS_DEREF			
E_ALIAS_LOOP			
E_ALIAS_PROBLEM			
E_ARG_BOUNDS			
E_ARG_SYNTAX			
E_ARG_VIOL			
E_ATT_BOUNDS			
E_ATT_OR_VALUE_EXISTS			
E_ATT_SYNTAX			
E_ATT_VALUE			
E_AUTHENTICATION			
E_BUSY	SH(UT)	SH(UT)	SH(UT)
E_CANT_CONSTRUCT		SH(UWP)	
E_CANT_INCORPORATE		SH(UWP)	
E_CHAIN			
E_CREDENTIALS			
E_DBE			
E_DIT_STRUCTURE			
E_DN			
E_DSA			

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)		
	Coordinate Shadow Update	Update Shadow	Request Shadow Update
E_ENTRY_EXISTS			
E_EXTENSION			
E_ILLEGAL_ROOT_OBJ			
E_ILLEGAL_ROOT_VAL			
E_INACTIVE_AGREEMENT	SH(IA)	SH(IA)	SH(IA)
E_INVALID_AGREEMENT	SH(IAID)	SH(IAID)	SH(IAID)
E_LOOP			
E_MATCH			
E_MISSED_PREVIOUS	SH(MP)		SH(MP)
E_MISSING_AVA			
E_MISSING_OBJECT_CLASS			
E_MORE_CURR_UPD_RCD	SH(UAR)		
E_MULTI_DSA			
E_NAMING_VIOLATION			
E_NO_AGMT_W_THIS_DSA	SH(IAID)	SH(IAID)	SH(IAID)
E_NO_ENTRIES_IN_ST			SH(NI)
E_NON_LEAF_OPERATION			
E_NONNAMING_ATTRIBUTE			
E_NOT_SINGLE_VALUED			
E_NO_SUCH_ATT		SH(IIR)	
E_NO_SUCH_OBJECT		SH(IIR)	
E_NO_SUCH_VALUE			
E_OBJECT_CLASS_MOD			
E_OBJECT_CLASS_VIOL			
E_OUTSIDE_UOR		SH(IIR)	

Table 13 - Error actions (continued)

Symptom (See Table 10)	Situation (See Table 11)		
	Coordinate Shadow Update	Update Shadow	Request Shadow Update
E_PREVIOUSLY_COORD	SH(IS)		
E_REFERENCE			
E_SCOPE			
E_PREVIOUSLY_SOLICITED			SH(IS)
E_SYSTEM_PERM	SH(UWP)	SH(UWP)	SH(UWP)
E_SYSTEM_TEMP	SH(UT)		SH(UT)
E_TIMEOUT			
E_TIMESTAMP_MISMATCH	SH(FUR)		SH(FUR)
E_TOO_MANY_UPDATES			SH(FUR)
E_UNABLE_TO_COMPLETE			
E_UNABLE_TO_PROCEED			
E_UNCOORDINATED		SH(IS)	
E_UNDEFINED_ATT			
E_UNRELIABLE_DATA	SH(FUR)		
E_UNSOLICITED		SH(IS)	
E_UNSUPPORTED_OC			
E_UNSUPPORTED_STRAT	SH(US)		SH(US)
E_UNUSABLE_DATA		SH(IIR)	
E_VERSION			
E_ZERO_VALUES			

Table 13 - Notes (continued)

NOTES

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 EARGBOUNDS, a ROSE "Rej" is also admissible.

13 The error is returned when there are no partial results, otherwise a partialOutcomeQualifier 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.

15 If a multicasting DSA receives this error and the matched part of the name is equal to or longer than that indicated by the next RDN to be resolved, name resolution shall be taken as having progressed. The error shall be relayed.

16 If a chaining or multicasting DSA receives this error and the matched part of the name is not equal to or longer than that indicated by the next RDN to be resolved, the error indicates an incompatibility in schema between the DSA and the one to which chaining takes place. Multicasting may continue, and the error in that case may be ignored. A DSA, having received such an error during name resoltuion, may but need not relay it.

Table 13 - Notes (concluded)

NOTES

17 If a DSA generates a chained operation on the basis of a cross reference and receives a serviceError with the problem of invalidReference in response, then it is recommended that the invalid cross reference be removed to eliminate repeated errors. Note that attempting to resolve the correct reference via the returnCrossRefs mechanism should be regarded as nonreliable due to the optional nature of returnCrossRefs. The resolution of an invalidReference due to a superior or subordinate reference is a local administrative issue.

Table 14 - Simple credential fields and protected simple authentication

Simple Credential Field	Equivalent Notation in Directory Documents, Part 8, figure 2
name	A
time1	t_1^A
time2	t_2^A
random1	q_1^A
random2	q_2^A
password	protected2

Annex A (normative)

Maintenance of Attribute Syntaxes

A.1 Introduction

The attribute types defined in the Directory Documents, Part 6, and listed in table 1 have requirements, in DSAs which support them, for underlying algorithms that:

a) check attribute values for syntactical correctness and compliance with pragmatic constraints;

b) match attribute values (comparing for equality, for matching substrings, and for relative ordering).

A.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.

A.3 Checking Algorithms

The subclauses 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.

A.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.

A.3.2 integerSyntax

Local implementations may apply local limitations.

A.3.3 telephoneNumberSyntax

The value of policing further rules is for further study (this applies also to telexNumber, teletexTerminalIdentifier, facsimileTelephoneNumber, G3FacsimileNonBasicParameters, x121Address, and ISDNAddress).

A.3.4 countryName

The value must be checked for compliance with ISO 3166: 1981 (E/F). (Note that from time to time further codes may be allocated.)

A.3.5 preferredDeliveryMethod

The values of the integer elements should not be restricted.

A.3.6 presentationAddress

No further checks should be applied.

A.4 Matching Algorithms

Matching algorithms are conveniently defined in terms of a two-step process:

a) 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.

b) 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.

A.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.)

A.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.

A.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.

Annex B (informative)

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

Annex C (informative)

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).

C.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:

a) Loop detection

b) 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 anyway 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).

C.2 Protocol Support

C.2.1 Usage of ChainingArguments

When using ChainingArguments:[2]

a) *originator* need not be used if requestor in CommonArguments is used;

b) *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);

c) *operationProgress*, *traceInformation*, *aliasDereferenced*, *aliasedRDNs*, *referenceType*, and *timeLimit* shall be generated, accepted, and used in accordance with the Directory Documents;

d) *returnCrossReferences* and info may optionally be generated, and shall always be accepted.

---

[2] In this subclause, the names of protocol elements (within ChainingArguments) are italicized.

C.2.2 Usage of ChainingResults

When using ChainingResults:[3] *crossReferences* and *info* may optionally be generated, and shall always be accepted.

[3] In this subclause, the names of protocol elements (within ChainingResults) are italicized.

Annex D (informative)

Guidelines for Applications Using the Directory

D.1 Tutorial

D.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).

D.1.2 Use of the Directory Schema

D.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:

a) Network management applications may with to define a SMAE object class;

b) 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.

D.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 strong

Authentication User and certification Authority. These two object classes should be considered auxiliary and used in conjunction with other, structural object classes.

D.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:

a) Locally adding attributes to a predefined superclass;

b) Locally making optional attribute types in a predefined superclass mandatory;

c) 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 4 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 4 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 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).

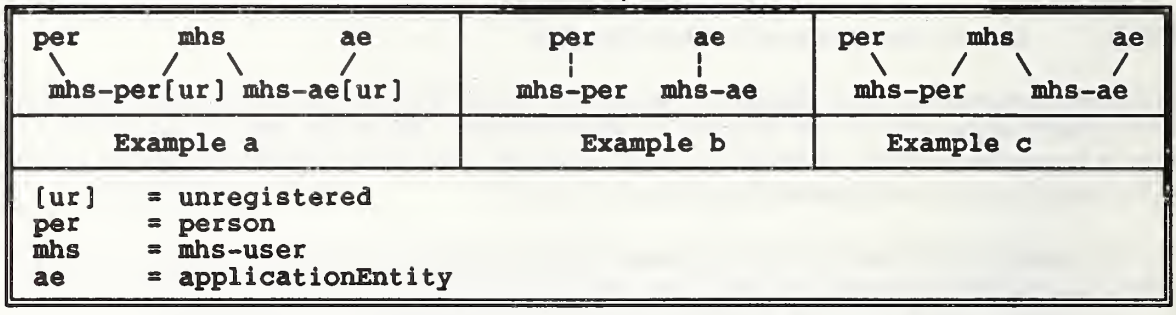


Figure 4 - Three ways of creating two object classes

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.

D.1.2.4 Side Effects of Creating Unregistered Object Classes

This subclause discusses two side effects of creating unregistered object classes.

a) 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.

b) 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.

D.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.

D.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.

D.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.

D.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.

D.4 Use of AETITLE

Applications wishing to make use of the AETitle field to access applicationEntity objects in the Directory are referred to Amendment 1 to ISO8650 for guidance on the purpose and appropriate useage of the AETitle field. In particular, implementors should be aware that:

a) AETitle should be used to uniquely distinguish individual application entities. It is inappropriate for applications to define a fixed AETitle to apply to all its instantiations;

b) The Directory does not perform name resolution on an object identifier (e.g., AETitle name form 2). The Directory does not support lookup based on OID, and AETitle name form 2 does not constitute a Directory Distinguished Name.

Annex E (informative)

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 meaning of the heading and should not be included in the profile).

a) 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)

b) 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.

c) 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.

d) STANDARD APPLICATION SPECIFIC OBJECT CLASSES

Any object classes supported from the relevant standards. For example, the MHS SIG might include mhs-user here.

e) 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.

f) 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.

g) 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.

Annex F (informative)

Bibliography

[ELGA85] ElGamal T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, No. 4, July 1985.

[DIFF76] Diffie W., Hellman M., "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, Nov. 1976

[COPP86] Coppersmith, D., Odlyzko, A., Schroeppel, R., "Discrete Logarithms inGF (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 FactoringIntegers," 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 (p2)," *IEEE Transactions on Information 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.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 12 - OS Security

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Dr. James Galvin, Trusted Information Systems**
SIG Editors: **John Hooder, NAVY, Dr. Mohammad Mirhakkak, MITRE**

Foreword

This part of the Stable Implementation Agreements was prepared by the Security Special Interest Group (SECSIG) of the Open Systems Environment Implementors' Workshop (OIW) hosted by the National Institute of Standards and Technology (NIST). See Procedures Manual for Workshop charter.

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject. There is significant technical change from this text as previously given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as strikeout. New and replacement text will be shown as shaded.

Table of Contents

List of Figures

List of Tables

Part 12 - Security

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

0 Introduction

The relationship between protocols and security is accomplished by developing a security profile that binds these two together. Security profiles define protocol specific implementations of security architectures.

A security profile includes the following items:

a) A grouping of the security services to be offered;

b) The placement of those security services;

c) The selection of mechanisms to support the placed security services.

This part completes this sequence of steps for several generalized security architectures. A generalized security architecture is chosen and tailored to derive a protocol-specific security profile. This part is comprised of protocol-specific security profiles and other supporting functions.

1 Scope

2 Normative References

[**ISO7498-2**] ISO/IEC 7498-2 *Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture, February 1989.*

[**ISO8649**] ISO/IEC 8649: 1988/Amd 1:1990 *Service Definition for the Association Control Service Element, Amendment 1: Peer-Entity Authentication During Association Establishment.*

[**ISO8650**] ISO/IEC 9594-3 *Information Technology - Open Systems Interconnection - The Directory - Part 3: Abstract Service Definition.*

[**ISO8650/1**] ISO/IEC 8650: 1988/Amd 1:1990 *Protocol Specification for the Association Control Service Element, Amendment 1: Peer-Entity Authentication During Association Establishment.*

[**ISO9594-7**] ISO/IEC 9594-7 *Information Processing Systems - Open Systems Interconnection - The Directory - Part 7: Selected Object Classes, 1990.*

[**ISO9594-8**] ISO/IEC 9594-8 *Information Processing Systems - Open Systems Interconnection - The Directory - Part 8: Authentication Framework, 1990.*

[**ISO10021-4**] ISO/IEC 10021-4 *Information Processing Systems - Text Communication - MOTIS -*

Message Transfer System : Abstract Service Definition and Procedures.

[X.509-88] CCITT X.509:1988 *The Directory - Authentication Framework.*

[X.511-88] CCITT X.511:1988 *The Directory - Abstract Service Definition.*

[X.411-84] CCITT X.411:1984 *Message Transfer System - Message Transfer Layer.*

[X.521-88] CCITT X.521:1988 *The Directory - Selected Object Classes.*

3 Definitions

4 Symbols and Abbreviations

5 Architectures

The purpose of this clause is to provide guidance on how to build a security architecture based on an OSI application environment and its threats and vulnerabilities.

A Security Architecture specifies the relationship between the set of security services and mechanisms with which protection from threats and vulnerabilities is achieved. It is designed to respond to assessed vulnerabilities, threats, and risks as identified by a security policy. The establishment of security policies is beyond the scope of the OIW.

5.1 Introduction

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

The sequence of steps by which a security architecture is created for a specific application environment is as follows:

a) Development of threat analysis;

b) Determination of security services;

c) Placement of security services;

d) Selection of mechanisms;

e) Selection of algorithms.

These implementation agreements assume that steps a and b have been completed for the specific application. An introduction to the threat analysis process and the determination of security services is included in Annex H.

Generic OSI application environments are defined in Clause 5.2. Generic security services as defined by ISO 7498-2 are grouped into classes in Clause 5.3. A generalized security architecture for each environment is developed by mapping the security classes onto the functional groups of each environment and providing guidance as to at which layer to support the service in Clause 5.4. Guidance on how to select mechanisms suitable for each security service is presented in Clause 5.5.

It is beyond the scope of these implementation agreements to specify the use of one algorithm over another. Clause 7 presents a set of algorithms suitable for various mechanisms.

5.2 Application Environments

It is useful for the sake of simplification to look at the OSI application environments and to separate them into generic OSI application environments so that security profiles can be developed for each. The environments are: Single Application Association, Application Relay, and Distributed Applications. All applications will operate in one or more of these environments. For example, a Message Handling application that uses a Message Transfer Agent (MTA) to relay mail from one User Agent (UA) to another UA would map to the Application Relay Environment. Likewise a Message Handling application which only includes a UA accessing a Message Store (MS) would map to the Single Application Association Environment.

For each environment, an architectural diagram is provided that portrays the interconnection of the elements. In addition, a set of functional groups are defined each of which is comprised of an interconnected set of elements.

5.2.1 Base Environment

Figure 1 depicts the basic elements of a generic OSI application environment from which all OSI application environments can be derived. In all application environment figures, dashed lines indicate an optional communication path and the double-lined boxes indicate an optional basic element. Ellipses indicate that the previous basic element may be repeated zero or more times.

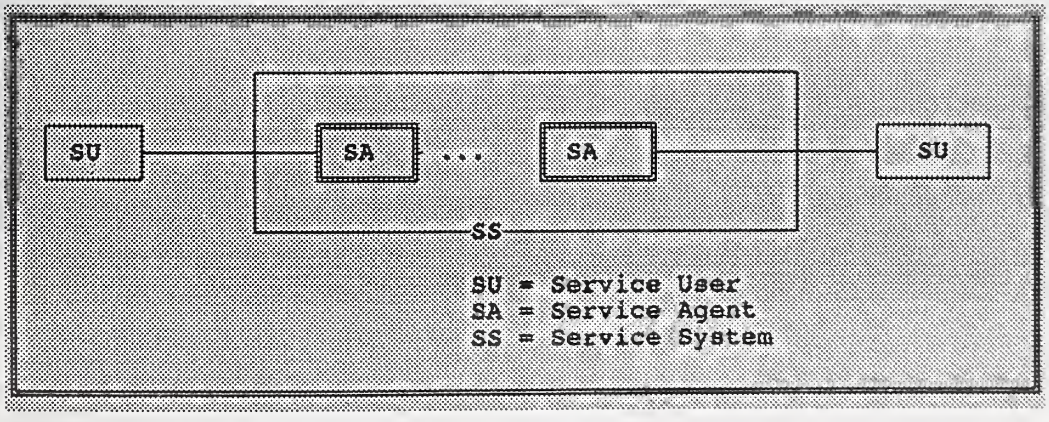


Figure 1 - Basic Elements of a Generic OSI Application Environment

The basic elements are as follows:

Service User (SU): an entity that functions as a service initiator or responder ;

Service Agent (SA): an intermediate entity that actively participates in providing the services between an initiator and a responder;

Service System (SS): zero or more cooperating service agents.

Basic elements that communicate, either through a direct association or indirectly through intermediaries, are classified as a functional group. Functional groups defined in Figure 1 are:

a. f_0: SU -> SU (Service User to Service User directly);

b. f_1: SU = > SU (Service User to Service User indirectly);

c. f_2: SU -> SA (Service User to Service Agent directly);

d. f_3: SU = > SA (Service User to Service Agent indirectly);

e. f_4: SA -> SA (Service Agent to Service Agent directly);

f. f_5: SA = > SA (Service Agent to Service Agent indirectly);

g. f_6: SA -> SU (Service Agent to Service User directly);

h. f_7: SA = > SU (Service Agent to Service User indirectly).

Editor's Note - the "->" notation indicates association security relationship and "= >" indicates relay security relationship.

These definitions and this functional group syntax will be used to define generic OSI application environments. In some applications, these functional groups may have to be combined for the purpose of performing a security analysis.

5.2.2 Single Application Association Environment

The Single Application Association Environment covers applications which are designed to operate over Single Application Associations (as defined in ISO 9545) between one pair of application-entity-invocations (AEIs). This environment specifically includes the case of recovery, i.e. different associations may exist at different times between one pair of AEIs.

Examples of applications to which this environment applies are as follows:

a) FTAM;

b) Network Management;

c) Virtual Terminal.

Applications such as MHS, Directory Services, and TP are only partially covered by this environment because some of their service elements may use store and forward or chaining types of relay functions. The environments that apply to these applications are the Application Relay and Distributed Applications Environments respectively.

5.2.2.1 Architectural Diagram

Figure 2 portrays the architectural diagram for the Single Application Association Environment.

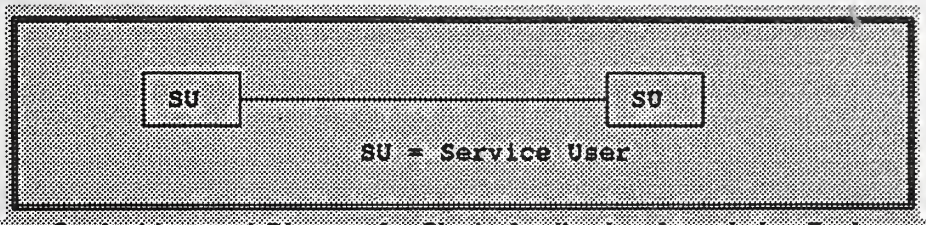


Figure 2 - Architectural Diagram for Single Application Association Environment

5.2.2.2 Functional Groups

The following functional group is defined for the Single Application Association Environment:

a) f_0:SU -> SU.

5.2.3 Application Relay Environment

The Application Relay Environment covers applications which are designed to operate with the active participation of at least one service agent in support of transferring a service user request from an initiator to a responder. When more than one service agent is present, they function sequentially.

An example of an application to which this environment applies is Message Handling Systems.

5.2.3.1 Architectural Diagram

Figure 3 portrays the architectural diagram for the Application Relay Environment. In all application environment figures, dashed lines indicate an optional communication path and the double-lined boxes indicate an optional basic element. Ellipses indicate that the previous basic element may be repeated zero or more times.

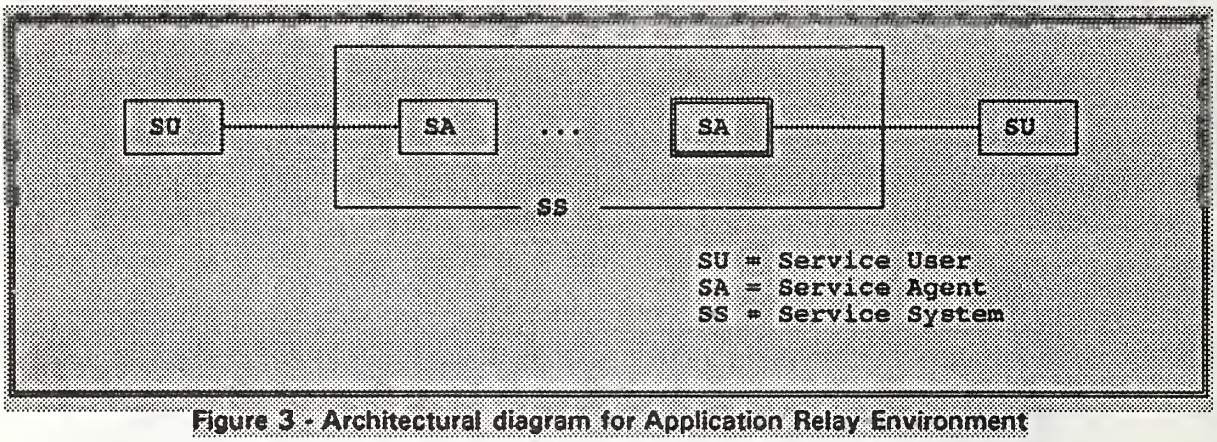


Figure 3 - Architectural diagram for Application Relay Environment

5.2.3.2 Functional Groups

The following functional groups are defined and added for the Application Relay Environment:

a. f_2: SU -> SA;

b. f_3: SU = > SA;

c. f_4: SA -> SA;

d. f_6: SA -> SU.

5.2.4 Distributed Applications Environment

The Distributed Application Environment covers applications which are designed to operate with the active participation of zero or more service agents which may process a service user request. Processing may include modifying, interpreting, or transferring the service user request or its data. When more than one service agent is present, they may function in parallel, sequentially, or both.

5.2.4.1 Architectural diagram

Figure 4 portrays the architectural diagram for the Distributed Applications Environment. In all application environment figures, dashed lines indicate an optional communication path and the double-

lined boxes indicate an optional basic element. Ellipses indicate that the previous basic element may be repeated zero or more times.

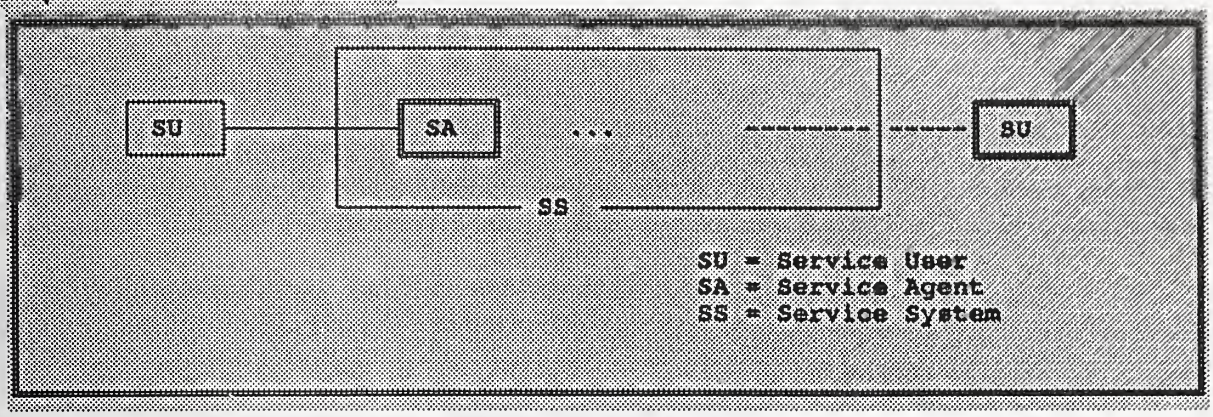


Figure 4 - Architectural diagram for Distributed Applications Environment

5.2.4.2 Functional Groups

The following functional groups are defined and added for the Distributed Applications Environment:

a) f_0: SU -> {SU; ... };

b) f_1: SU = > {SU; ... };

c) f_2: SU -> {SA; ... };

d) f_3: SU = > {SA; ... };

e) f_4: SA -> {SA; ... };

f) f_5: SA = > {SA; ... };

g) f_6: SA -> {SU; ... };

h) f_7: SA = > {SU; ... };

5.3 Security Classes

Security classes are defined to provide a framework on which to build security profiles. Each class specifies the required security services. The services specified in each class are the generic security services as defined by ISO 7498-2. For each application's profile, specific security services are chosen for each class. For example, data integrity is a generic security service for which there exists five distinct data integrity services. One or more specific security services must be specified to meet the requirements of a security class in an application specific security profile.

The classes are organized into two similar hierarchies as shown in Table 1. Each level of each hierarchy is a superset of the security services required of the immediately preceding level. For each level in the hierarchies, the same set of security services are required, except that one hierarchy includes confidentiality services.

Table 1 - Security Classes

SECURITY SERVICES	SECURITY CLASSES	
		ADD CONF
AUTH. & ACCESS CONTROL	S0	S0A
ADD DATA INTEGRITY	S1	S1A
ADD NON-REPUDIATION	S2	S2A

There are two interesting properties of these relationships between the classes. First, each level of the confidentiality hierarchy is a superset of the other hierarchy at the same level and a superset of the confidentiality hierarchy at the immediately preceding level. For example, class S2A is a superset of classes S2 and S1A.

Second, for two entities each supporting a distinct security class in a different hierarchy, the best level of service that can be achieved between them is the class in the non-confidentiality hierarchy at the same level as the lowest class of the two entities. For example, if one entity supports class S2 and the other supports class S1A, the best class of service achievable is S1.

Editor's Note - This is not a mechanism for negotiated services. That is a future work item.

5.3.1 Security Class S0

The Security Class S0 includes implementation of the following security services:

a) S0 = Authentication and Access Control.

The Security Class S0A adds the confidentiality service to the Class S0 as follows:

b) S0A = S0 + Confidentiality.

5.3.2 Security Class S1

The Security Class S1 adds the Data Integrity Service to class S0 as follows:

a) S1 = S0 + Data Integrity.

The Security Class S1A adds the Confidentiality Service to Class S1 as follows:

b) S1A = S1 + Confidentiality

5.3.3 Security Class S2

The Security Class S2 adds the Non-repudiation Service to Class S1 as follows:

a) S2 = S1 + Non-repudiation

The Security Class S2A adds the Confidentiality Service to Class S2 as follows:

b) S2A = S2 + Confidentiality

5.4 Guidelines for OIW Application Profile Development

6 Key Management

[ISO7498-2] defines Key Management (KM) as the "generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy." The Security SIG recognizes that security policies are outside the scope of IAs, and it is inappropriate to make general recommendations in the absence of a KM framework.

7 Security Algorithms

Editor's Note - Implementors are cautioned that security of an algorithm may change at any time. Therefore, the WIA must be consulted in order to determine if there is more current information.

Editor's Note -

The algorithms included here are listed in no particular order (beyond categorization by type of algorithm). It is beyond the scope of these agreements to recommend the use of one algorithm over another. However, if a vulnerability is known to exist a reference will be provided along with a recommendation not to use the algorithm.

This clause references a definitive specification for each algorithm, which includes an object identifier. In general, control of the definitive specification is expected to be outside the scope of the OIW. The benefit of not controlling the specification is that the organization that developed the algorithm is best situated to maintain and have knowledge of the security of the algorithm. Algorithms for which there is no controlling organization are defined in an Annex in this Part.

For each algorithm, its typical usage is stated, its definitive reference is given, and its object identifier is included for reference purposes.Optionally, additional information may be included, for example a reference to known vulnerabilities.

Implementors should be aware that export of products using cryptography may be subject to export restrictions. In general, use of cryptography not involving confidentiality is subject to Commerce Department regulations, while use of cryptography for confidentiality is controlled by (more stringent) State Department regulations. It is the implementor's responsibility to determine any export restrictions which apply to a given product, as the export controls may change from time to time.

Editor's Note - Some of the references are RFCs, Internet Drafts, and PKCS documents. We need to include information on how to access these documents.

~~7.1 Square-Mod-N~~

~~Square-Mod-N is a hash algorithm that is used to compute a fixed size representation of an input stream. It is defined in [X.509] and its object identifier is defined there as:~~

```
~~sqMod-n ALGORITHM~~
~~PARAMETER BlockSize~~
~~::= {hashAlgorithm 1}~~

~~BlockSize ::= INTEGER~~
```

~~Recent research regarding the square-mod-n one-way hash function described in Annex D of [X.509] has revealed that the function is not secure. Its use, therefore, is discouraged.~~

~~**Editor's Note** - We need the reference that identifies its vulnerabilities so we can recommend it not be used.~~

7.2 Message Digests

These message digest algorithms (or hash algorithms) compute a fixed size representation of an input stream. They have different performance characteristics and employ different computational techniques, making each suitable for different applications.

7.2.1 Square-Mod-N

Square-Mod-N is a hash algorithm that is used to compute a fixed size representation of an input stream. It is defined in [X.509] and its object identifier is defined there as:

```
sqMod-n ALGORITHM
PARAMETER BlockSize
::= {hashAlgorithm 1}

BlockSize ::= INTEGER
```

Recent research regarding the square-mod-n one-way hash function described in Annex D of [X.509] has revealed that the function is not secure. Its use, therefore, is discouraged.

Editor's Note - We need the reference that identifies its vulnerabilities so we can recommend it not be used.

7.2.2 MD2

MD2 is a message digest algorithm that employs accepted, traditional computational techniques. Its speed is the slowest of the message digests listed here.

It is defined in Internet Draft [a] and its object identifier is defined there as:

```
md2 ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) digestAlgorithm(2) 2}
```

Editor's Note - There is a Directory SIG OID for this algorithm.

The reference includes a source code implementation of the algorithm written in the C programming language. MD2 is copyrighted and its use may require specific permission or a license. Details are stated in the Internet Draft.

7.2.3 MD4

MD4 is a message digest algorithm that employs non-traditional computational techniques to enhance its speed in software and hardware with native 32-bit arithmetic. Its speed is the fastest of the message digests listed here.

It is defined in Internet Draft [b] and its object identifier is there as:

```
md4 ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) digestAlgorithm(2) 4}
```

This reference includes a source code implementation of the algorithm written in the C programming language.

It is suggested that MD4 be used only with applications for which performance is critical.

Editor's Note - We need to include text from the MD4/5 Internet Drafts which describes the differences between the two algorithms and the preference for MD5.

7.2.4 MD5

MD5 is a message digest algorithm ~~that employs traditional computational techniques with the speed enhancements of MD4~~ which is based on the techniques of MD4, but with additional enhancements to counter proposed attacks. A detailed description of the changes can be found in [c].

~~It~~MD5 is defined in Internet Draft [c] and its object identifier is defined there as:

```
md5 ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) digestAlgorithm(2) 5}
```

This reference includes a source code implementation of the algorithm written in the C programming language.

7.2.5 SHA

This algorithm is the NIST Secure Hash Algorithm [ab]. It is based on concepts similar to those used in MD4 and MD5, and outputs a 160-bit digest.

```
sha ALGORITHM
PARAMETER NULL
::= {algorithm 18}
```

Editor's Note - This and other algorithms may be registered by ISO instead, in which case this text will be adjusted prior to moving to Stable Agreements, or if necessary as Alignment Errata.

7.3 ~~RSA~~Reversible Public Key Algorithms

These algorithms are asymmetric; separate keys are used for encryption and decryption. They also have the property that applying the encipherment function followed by the decipherment function has the same effect as applying the decipherment function followed by the encipherment function. This is useful if a single algorithm is needed to provide both confidentiality (e.g., transport of symmetric keys) and authentication/integrity (e.g., digital signatures).

7.4 ~~RSA~~

RSA is a public key (asymmetric) cryptographic algorithm, typically used in conjunction with message digest (or hash) algorithms to create digital signatures and for confidential distribution of symmetric keys. It may also be used to exchange confidential messages.

The RSA algorithm is defined in [d] and is also described in Annex C of [X.509]. The RSA technology is patented in the United States [e][f].

~~**Editor's Note** - Explain why there are two definitions.~~

According to [X.509], the ASN.1 BIT STRING containing the public key will contain the BER encoding of the modulus and exponent:

```
SEQUENCE {
        n       INTEGER,    -- modulus
        e       INTEGER }   -- public exponent
```

7.4.1 RSA ~~Definition~~(X.509)

RSA is defined in [X.509] and its object identifier is defined there as:

```
rsa ALGORITHM
PARAMETER KeySize
::= {encryptionAlgorithm 1}

KeySize ::= INTEGER
```

The key size specifies the length in bits of the RSA public key modulus.

The definition of this algorithm does not include specification of padding rules. If one assumes that the data is treated as an integer and padded with zero bits on the left, the algorithm is subject to various attacks, such as those described in [ah], which render it unsuitable for some applications, e.g., multi-recipient mail, notarization. In such cases RSAEncryption is preferred.

7.4.2 RSA Encryption

RSA Encryption is defined in PKCS #1 [g] and its object identifier is defined there as:

```
rsaEncryption ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) 1}
```

This algorithm defines various types of block padding depending on whether the block is being encrypted using a public or private key. The padding protects against various attacks documented in the literature.

7.5 Irreversible Public Key Algorithms

These algorithms are not reversible, as defined in section 7.2. Typically, different algorithms are used for encryption and signature. This section defines several signature-only algorithms. Note that these algorithms expand the plaintext, producing output which is significantly larger than the input block or digest. These algorithms are of use in authentication-only systems, and are generally not subject to export restrictions.

7.5.1 El Gamal

ElGamal is a public key (asymmetric) digital signature algorithm. It is defined in [k]. Its object identifier is:

```
ElGamal ALGORITHM
PARAMETER NULL
::= {encryptionAlgorithm 1}
```

Editor's Note - This OID was assigned by the Directory SIG.

In [X.509], the ASN.1 data element subjectPublicKey defined as BIT STRING should be interpreted in the case of ElGamal as being of type:

```
SEQUENCE {
prime INTEGER, -- p
base INTEGER,  -- alpha
key INTEGER -- public key, Y
}
```

Also, in [X.509], the value associated with the ENCRYPTED MACRO should be interpreted in the case of ElGamal as being of type:

```
SEQUENCE {
s INTEGER,
r INTEGER
}
```

The ElGamal technology is patented in the United States [f].

Editor's Note - Should we describe and define OIDs for the message digest with ElGamal signature algorithms? There is a Directory SIG OID for md2WithElGamal.

7.6 Key Exchange

7.6.1 Diffie-Hellman

Diffie-Hellman Key Exchange is a public key (asymmetric) algorithm whereby two parties, without any prior arrangements, can agree upon ~~a secret keysome~~ shared (secret) information. The parties exchange public information which, in conjunction with private information retained by each user, may be used to compute a common value. This value is typically used as a symmetric key, ~~This key could be used~~, for example, to encrypt further communications between the parties.

The Diffie-Hellman Key Exchange is defined in [h] and is also described in [j]. The Diffie-Hellman Key Exchange is patented in the United States [i][f].

The object identifier is defined in PKCS #3 [j] as:

```
dhKeyAgreement ALGORITHM
PARAMETER DHParameter
::= {iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-3(3) 1}

DHParameter ::= SEQUENCE {
prime INTEGER, -- p
base INTEGER   -- g
}
```

7.6.2 Diffie-Hellman with Common Parameters

This version of Diffie-Hellman assumes the use of a common modulus and generator, which are distributed by external means rather than being conveyed in the parameter component of the AlgorithmIdentifier. The patent restrictions in the previous section still apply.

The object identifier is defined as:

```
dhWithCommonModulus ALGORITHM
PARAMETER NULL
::= {algorithm 16}

DHParameter ::= SEQUENCE {
prime INTEGER, -- p
base INTEGER   -- g
}
```

7.7 Signature Algorithms

This section specifies a number of signature algorithms, i.e., hash algorithms combined with appropriate asymmetric encryption algorithms.

~~7.8 Square-Mod-N with RSA~~

~~Square Mod N is a signature algorithm that combines the Square Mod N hash algorithm with the RSA cryptographic algorithm to produce a digital signature. This algorithm is defined in [X.509] and its object identifier is defined there as:~~

```
sqmod-nWithRSA-ALGORITHM
PARAMETER KeyAndBlockSize
::= {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER
```

~~Recent research regarding the square-mod-n one way hash function described in Annex D of [X.509] has revealed that the function is not secure. Its use, therefore, is discouraged.~~

7.9 ~~Message Digests with RSA~~

7.9.1 Message Digests with RSA

The algorithms listed below are signature algorithms that combine a message digest algorithm with the RSA cryptographic algorithm to produce a digital signature.

Editor's Note - The OIDs below have been assigned by the Directory SIG and the Security SIG. Should we explain why they do not appear in a single tree?

7.9.1.1 Square-Mod-N with RSA

Square-Mod-N is a signature algorithm that combines the Square-Mod-N hash algorithm with the RSA cryptographic algorithm to produce a digital signature. This algorithm is defined in [X.509] and its object identifier is defined there as:

```
sqmod-NwithRsa ALGORITHM
PARAMETER KeyAndBlockSize
::= {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER
```

Recent research regarding the square-mod-n one-way hash function described in Annex D of [X.509] has revealed that the function is not secure. Its use, therefore, is discouraged.

7.9.2 ~~MD2 with RSA~~

7.9.2.1 MD2 with RSA

Its object identifier is:

```
md2WithRsa ALGORITHM
PARAMETER NULL
::= {signatureAlgorithm 1}
```

Editor's Note - This OID was assigned by the Directory SIG.

7.9.3 ~~MD4 with RSA~~

7.9.3.1 MD4 with RSA

Its object identifier is:

```
md4WithRSA ALGORITHM
PARAMETER NULL
::= {algorithm 2}
```

7.9.4 ~~MD5 with RSA~~

7.9.4.1 MD5 with RSA

Its object identifier is:

```
md5WithRSA ALGORITHM
PARAMETER NULL
::= {algorithm 3}
```

7.10 ~~Message Digests with RSA Encryption~~

7.10.1 Message Digests with RSA Encryption

The algorithms listed below are signature algorithms that combine a message digest algorithm with the RSA Encryption cryptographic algorithm to produce a digital signature.

7.10.2 ~~MD2 with RSA Encryption~~

7.10.2.1 MD2 with RSA Encryption

MD2 with RSA encryption is defined in PKCS #1 [g] and its object identifier is defined there as:

```
md2WithRSAEncryption ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) 2}
```

7.10.3 ~~MD4 with RSA Encryption~~

7.10.3.1 ~~MD4 with RSA Encryption~~

Its object identifier is:

```
md4WithRSAEncryption ALGORITHM
PARAMETER NULL
::= {algorithm 4}
```

7.10.4 ~~MD5 with RSA Encryption~~

7.10.4.1 ~~MD5 with RSA Encryption~~

MD5 with RSA Encryption is defined in PKCS #1 [g] and its object identifier is defined there as:

```
md5WithRSAEncryption ALGORITHM
PARAMETER NULL
::= {iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4}
```

~~7.11 Diffie-Hellman Key Exchange~~

~~Diffie-Hellman Key Exchange is a public key (asymmetric) algorithm whereby two parties, without any prior arrangements, can agree upon a secret key. This key could be used, for example, to encrypt further communications between the parties.~~

~~The Diffie-Hellman Key Exchange is defined in [h] and is also described in [j]. The Diffie-Hellman Key Exchange is patented in the United States [i][f].~~

~~The object identifier is defined in PKCS #3 [j] as:~~

```
dhkeyagreement ALGORITHM
PARAMETER DHParameter
::= {iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-3(3) 1}

DHParameter ::= SEQUENCE {
prime INTEGER, -- p
base INTEGER -- g
}
```

~~7.12 El Gamal~~

~~ElGamal is a public key (asymmetric) digital signature algorithm. It is defined in [k]. Its object identifier is:~~

```
ElGamal ALGORITHM
PARAMETER NULL
::= {encryptionAlgorithm 1}
```

~~Editor's Note – This OID was assigned by the Directory SIG.~~

~~In [X.509], the ASN.1 data element subjectPublicKey defined as BIT STRING should be interpreted in the case of ElGamal as being of type:~~

```
SEQUENCE {
prime INTEGER, --- p
base INTEGER, --- alpha
key INTEGER -- public key, Y
}
```

~~Also, in [X.509], the value associated with the ENCRYPTED MACRO should be interpreted in the case of ElGamal as being of type:~~

```
SEQUENCE {
a INTEGER,
r INTEGER
}
```

~~The ElGamal technology is patented in the United States [f].~~

~~Editor's Note – Should we describe and define OIDs for the message digest with ElGamal signature algorithms? There is a Directory SIG OID for md2WithElGamal.~~

7.13 ~~Data Encryption Standard~~Symmetric Encryption Algorithms

7.13.1 Data Encryption Standard

The Data Encryption Standard (DES) is a secret key (symmetric) cryptographic algorithm. It is defined in FIPS 46-1 [l]. It is also defined as DEA-1 in ANSI X3.92-1981 [m].

Implementors will also find several other references useful. FIPS PUB 74 [p] provides guidance on the implementation and use of DES and includes a complete specification of the algorithm. SPEC PUB 500-20 [p] describes the design and operation of the NIST (formerly NBS) testbed that is used for the validation of DES implementations. It specifies a set of 291 test cases that have been designed to exercise every basic element of the algorithm, and as a further check on the correctness of an implementation, it specifies an extensive Monte Carlo analysis. SPEC PUB 500-61 describes the design of four maintenance tests for DES implementations. The tests consist of an iterative test procedure

that uses a small program and minimum data. The tests are designed to be independent of implementation and to be fast enough to test devices during actual operation. The tests are defined as four specific stopping points in a general testing process and satisfy four testing requirements of increasing degree of completeness on the thoroughness of testing desired.
There are four modes of operation of the DES, as specified by FIPS 81 [n] and ANSI X3.106-1983 [o]. The modes specify how the data will be encrypted and decrypted. In all cases the key is 64 bits. Use of DES for encryption (i.e., all modes discussed below except DES-MAC) are subject to export controls.

7.13.1.1 DES-ECB

This is the Electronic Codebook mode of operation. Its object identifier is:

```
desECB ALGORITHM
PARAMETER ~~CBCParameter~~NULL
::= {algorithm 6}
```

This mode should be used to encrypt small blocks (e.g., other DES keys). Its use is deprecated for block encryption since it allows cryptanalysis of repeated block values (i.e., the same plaintext in the same place relative to the block), as well as reassembling messages from known blocks.

7.13.1.2 DES-CBC

This is the Cipher Block Chaining mode of operation. Its object identifier is:

```
desCBC ALGORITHM
PARAMETER CBCParameter
::= {algorithm 7}
```

The PARAMETER is needed to specify the Initialization Vector, which need not be kept secret.

This mode should be used to encrypt multiple blocks, where the full message is available. The random IV prevents codebook analysis of the start of the chain. The IV may be public.

This mode will propagate a single bit error in one plaintext block into all succeeding blocks, and will propagate a single bit error in the ciphertext into a garbled plaintext block on decryption as well as a single bit error in the next plaintext block.

The following padding mechanism from [w] should be used if the data to be encrypted is octet aligned, unless the security policy dictates otherwise:

The input to the DES CBC encryption process must be padded to a multiple of 8 octet, in the following manner. Let n be the length in octets of the input. Pad the input by appending 8-(n mod 8) octet to the end of the message, each having the value 8-(n mod 8), the number of octets being added. In hexadecimal, the possible paddings are: 01, 0202, 030303, 04040404, 0505050505, 060606060606, 07070707070707, and 0808080808080808. All input is padded with 1 to 8 octets

to produce a multiple of 8 octets in length. The padding can be removed unambiguously after decryption.

> **Editor's Note** - If adding the padding rules would cause existing implementations to break, this should be registered as a separate algorithm identifier. Note, however, that [FIPS 81] specifies its own padding rules for padding binary data, in the absence of application-defined rules such as those above; those rules require an indication (which could be conveyed as an algorithm PARAMETER) of whether the data has been padded or not.

7.13.1.3 DES-OFB

This is the Output Feedback mode of operation. Its object identifier and parameters are:

```
desOFB ALGORITHM
PARAMETER FBParameter
::= {algorithm 8}
```

The parameters are needed to specify an IV and the number of feedback bits.

This mode may be used to encrypt multiple blocks where the error extension properties of DES-CBC are undesirable. A single bit error in the ciphertext will cause only a single bit error in the output plaintext.

7.13.1.4 DES-CFB

This is the Cipher Feedback mode of operation. Its object identifier and parameters are

```
desCFB ALGORITHM
PARAMETER FBParameter
::= {algorithm 9}
```

The parameters are needed to specify an IV and the number of feedback bits.

This mode may be used when the plaintext is made available in pieces, e.g., a character (8-bit CFB) or a bit (1-bit CFB) at a time. This mode will propagate a single bit error in one plaintext block into all succeeding blocks, and will propagate a single bit error in the ciphertext into a single-bit error in the corresponding plaintext character as well as garbling of the next 8 bytes or so of output (the exact amount depends on the feedback size).

7.13.1.5 DES-MAC

DES-MAC is a Message Authentication Code algorithm (cryptographic checksum) based on the DES that uses a single 64-bit DES key.

It is specified in FIPS 113 [s] and is equivalent to the binary mode defined in ANSI X9.9-1986 [t]. Its object identifier and parameter are:

```
desMAC ALGORITHM
PARAMETER MACParameter
::= {algorithm 10}
```

The parameter is needed to specify the MAC length in bits.

DES-MAC is equivalent to DES-CBC using an all zero Initialization Vector (IV), with all but the last cipher output block discarded. Separate keys (where one may simply be a variant of the other) should be used if both DES-CBC encrypting and MACing the same data.

Editor's Note - We need to include the reference which specifies the vulnerability when the same key is used to DES-CBC encrypt and MAC the same data, and recommends the use of separate keys.

7.13.1.6 DES-EDE

The DES algorithm in Encrypt-Decrypt-Encrypt (EDE) mode, as defined by [af] for encryption and decryption with pairs of 64-bit keys, might be used for key or MAC encryption when symmetric key management is employed. (The mechanism is subject to the same constraints as DES ECB, but is cryptographically stronger.) Given the pair of keys, the data is enciphered with the first key, deciphered with the second key, and enciphered again with the first key to perform encryption; the process is reversed for decryption. Note that if both keys are the same, the result is equivalent to a single encryption under the single key. The key may be represented as a single 128-bit string with the first 64 bits being the first key and the last 64 bits being the second key.

```
desEDE ALGORITHM
PARAMETER NULL
::= {algorithm 17}
```

7.13.2 RC2-CBC

RC2-CBC is a symmetric block encryption algorithm. It is proprietary to RSA Data Security, Inc., and a license from them is required to use the algorithm. The algorithm uses an 8-byte key and operates on an 8-byte block, with cipher block chaining as in DES. The recommended padding is as described above for DES-CBC: the final block is padded to an 8-byte boundary by appending 8 - (n mod 8) bytes, each having the value 8 - (n mod 8), where n is the total number of bytes being encrypted. The speed is comparable to DES.

```
rc2CBC ALGORITHM
PARAMETER RC2-CBCParameter
::= { iso(1) member-body(2) US(840) rsadsi(113549) encryptionAlgorithm(3) 2}

RC2-CBCParameter ::= CHOICE {IV, SEQUENCE {version RC2Version, IV}}

-- with IV only, version defaults to 65

IV ::= OCTET STRING -- 8 octets
RC2Version ::= INTEGER -- 0 to 255, defined by RSADSI
```

The version number relates to the security level. Different versions of RC2 provide different security levels, some of which are exportable.

7.13.3 RC-4

RC-4 is a symmetric block encryption algorithm. It is proprietary to RSA Data Security, Inc., and a license from them is required to use the algorithm. The RC4 key size is variable, 1 to 256 bytes; the block size is one byte. RC4 is a stream cipher, and it exclusive-ors a pseudorandom sequence generated from the key to encrypt or decrypt; a given key should therefore be used only once. RC4 is very fast.

```
rc4 ALGORITHM
PARAMETER NULL
id= { iso(1) member-body(2) US(840) rsadsi(113549) encryptionAlgorithm(3) 4}
```

7.14 ASN.1

7.14.1 Distinguished Encoding Rules

In order to allow verification of digital signatures produced by the SIGNED and SIGNATURE MACROs of [ISO9594-8], it is necessary to define a set of distinguished encoding rules to produce an unambiguous encoding of a given abstract syntax value. [ISO9594-8] defines a number of such encoding rules (8.7), but is, unfortunately, underspecified in the following areas:

a) Ordering of SET OF components;

b) Handling of unused trailing zero bits;

c) Invocation and designation of new character sets in some of the character string types.

The following rules remove these ambiguities:

a) The [ISO9594-8] distinguished encoding rules are always used;

b) For SET OF types, components are sorted into ascending order of the distinguished encodings of the components;

c) For BIT STRINGS with unused trailing bits, if the type definition that specifies the bits have significance, then they are included in the encoding; otherwise they are not;

d) For those character strings which allow it, escape sequences are generated to invoke and designate new register entries only when the register entry for the character currently being encoded is different from that currently designated for G0, C0, or C1. All designations shall be into G0 or C0. (It is assumed that all characters have entries in the ISO Registry of Coded Character Sets.)

NOTE - Rules b,c, and d are taken from [ISO/CD8825-3] (Nov. 1990), the ASN.1 Distinguished Encoding Rules. Other features of [ISO/CD8825-3], which conflict with [ISO9594-8] (e.g., length encoding for constructors), are NOT used by this IA.

It is recommended that whenever the SIGNED or SIGNATURE macro is to be applied to an object, the object should be transferred in its distinguished encoded form. In this way, when the resources required to encode or decode an object exceed the resources required to apply the SIGNED or SIGNATURE macro, a receiving entity may apply the macro immediately, thus realizing enhanced performance. However, if the macro application is unsuccessful, the object must be distinguished encoded and the macro re-applied to determine its actual success or failure.

8 Lower Layers Security

9 Upper Layers Security

This clause addresses the provision of security services in the Upper Layers. The Upper Layers Security Model specifies the interactions among the Upper Layers in providing and using security services [ISO/CD10745].

9.1 Security Mechanisms

9.1.1 Peer Entity Authentication

ACSE authentication extensions [ISO8649][ISO8650/1] support two-way authentication through the definition of a new functional unit. When this functional unit is employed, additional parameters are provided by the A-ASSOCIATE service to indicate this requirement and convey authentication information between entities. The ASN.1 definition for this information is given below:

from [ISO8650/1]:

```
Authentication      ::= SEQUENCE {
  mechanism-name        [0] IMPLICIT OBJECT IDENTIFIER OPTIONAL,

Mechanism-name ::= OBJECT IDENTIFIER
        --This field shall be present if authentication-value is of type ANY.

aAuthentication-value  [1] =  CHOICE {
        charstring              [0] IMPLICIT GraphicString,
        bitstring               [1] IMPLICIT BIT STRING,
        external                [2] IMPLICIT EXTERNAL,
        other                   [3] ANY DEFINED BY m-- Defined by Mechanism-name }}

        --The abstract syntax of authentication-value is determined by the authentication-mechanism
        --used during association establishment. The authentication-mechanism is either explicitly
        --denoted by the OBJECT IDENTIFIER value for Mechanism-name, or it is know implicitly by
        --prior agreement between the communicating partners. If "other" is chosen, then
        --"Mechanism-name" must be present in accordance with ISO 8824.
```

Figure 1 - A-ASSOCIATE Authentication Information

These agreements define the following mechanisms for use with this ACSE functional unit:

simple-strong authentication mechanism.

9.1.1.1 Simple-Strong Authentication

9.1.1.1.1 Operation

The operation of the simple-strong authentication mechanisms are based upon [ISO9594-3] and [ISO9594-8] standards. The sending system is the entity requesting authentication of its identity, and the receiving system is the entity performing the authentication. The sending system supplies data for the ACSE authentication field of the A-ASSOCIATE primitive. The receiving ACSE obtains the ACSE authentication data from the A-ASSOCIATE PDU, and it performs the authentication check. If the check is successful, the association formation succeeds or fails depending upon other circumstances and parameters. The use of the ACSE authentication fields support both the simple and strong credentials variants of the [ISO9594-8] authentication exchanges.

Certificates for use with strong authentication must be compatible with [ISO9594-8].
Certificates procured for use with Internet Privacy Enhanced Mail [u][v][w][x] are completely compatible with [ISO9594-8] and may (subject to licensing restrictions) be used by the strong authentication mechanism. However, Privacy Enhanced Mail uses only a subset of the suggested [ISO9594-7] name forms, and might not support certain name forms of interest to specific OIW applications. Examples include Application Entity names and certain name forms defined by the North American Directory Forum in NADF-123 [y].

9.1.1.1.2 Data Structure

Mechanism Name

The following is the ASN.1 description of the authentication data structure for simple or strong authentication:

```
simple-strong-auth-mechanism OBJECT IDENTIFIER ::= {iso (1)
              identified-organization (3)
              oiw (14)
              secsig (3)
              authentication-mechanisms (3)
              simple-strong-identity-authentication (1)
             }
```

Authentication Value

The authentication value is conveyed in the other option of the authentication-value field of ACSE authentication.

```
Authentication-Value ::=
 SEQUENCE OF DirectoryAbstractService.Credentials
```

This data type is defined in ASN.1 module DirectoryAbstractService of [ISO9594-3] as modified through resolution of Directory Defect Report Numbers 9594/052 and 063. The semantics of all fields are as specified in clause 8.1.2.1 of [ISO9594-3].

The Authentication-Value is defined as a SEQUENCE because it is permitted to pass credentials for multiple entities in the authentication value. It is the responsibility of the application to determine the specific meaning and use of multiple credentials in such a case. It is anticipated that specific applications (e.g., Network Management) would provide specifications for handling multiple credentials within their own clauses of this Part.

This authentication mechanism may employ any registered authentication algorithm; however, it is recommended that the algorithms identified in clause 7 be used.

9.1.1.1.3 Options

For the Simple Credentials option of Credentials, the following agreements apply. Conforming implementations are not required to employ the OPTIONAL validity sequence of the SimpleCredential data element. Receiving implementations that do not employ the validity sequence must reject an authentication value which does contain this sequence. Conforming implementations shall employ the optional password field of the SimpleCredential data element.

Note that the password may be hashed using one way functions and the other validity fields. Password is either cleartext, Protected1 or Protected2 according to [ISO9594-8].

9.1.1.2 External Authentication Mechanisms

Externally defined authentication exchanges may employ the external [2] option of the authentication-value field of ACSE authentication. In this case it is recommended that the mechanism-name be omitted, with the particular mechanism in use being implied by the abstract syntax identified in the external construct.

9.1.1.2.1 Kerberos Version 5

One instance of an external authentication mechanism is the Kerberos mechanism defined in [z]. The Kerberos specification assigned the following object identifier to an abstract syntax suitable for use in this way:

[TBD]

10 Message Handling System (MHS) Security

All current MHS security relevant text appears in Part 8, clause 10.

11 Directory Services Security

12 Network Management Security

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

12.1 Threats

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

The primary threats to be protected against are the following:

a) The masquerading of a manager or agent entity;

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

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

The secondary threats to be protected against are the following:

a) All primary threats;

b) The disclosure of CMIP data units;

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

12.2 Security Services

12.2.1 Basic Security Services

The security services required to counter primary threats are:

a) Peer entity authentication;

b) Data origin authentication;

c) Connectionless integrity.

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

Connectionless integrity insures that data units originating from an authenticated source are not modifiable without detection. When combined with a strong data origin authentication mechanism, the ability to fabricate new data units is also countered. Connectionless integrity may be provided at either the application level of the protocol or within one of the lower levels of the protocol (i.e., transport or network).

12.2.2 Enhanced Security Services

The security services required to counter secondary threats are:

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

b) Connectionless confidentiality;

c) Connection integrity with or without recovery.

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

12.3 Security Mechanisms

12.3.1 Peer Entity Authentication

Peer Entity Authentication will use the ACSE authentication mechanism and associated data types as defined in clause 9 of this Part of the IAs. The specific authentication mechanism to be supported is the Simple-Strong Authentication defined in 9.1.1.1.

Support of ACSE authentication is optional.

12.3.2 Connectionless Integrity

Editor's Note - Proposed text for this clause appears in WIA Part 12, clause 12.3.2.

Annex A (normative)

ISPICS Requirements List

Annex B (normative)

Errata

Table 2 - SIA Part 12 changes

NO. OF ERRATA	TYPE	REFERENCED DOCUMENT	CLAUSE	NOTES

Annex C (normative)

TBD

Annex D (informative)

Security Algorithms

```
OIWSECSIGAlgorithmObjectIdentifiers  {iso(1) identified-organization(3)
                                      oiw(14) secsig(3)
                                      oIWSECSIGAlgorithmObjectIdentifiers(1)}
DEFINITIONS =
BEGIN

EXPORTS
-- to be determined

IMPORTS
-- none

-- category of information object
-- defining our own here; perhaps the definition should be imported from
-- {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

algorithm       OBJECT IDENTIFIER ::= {iso(1) identified-organization(3)
                                      oiw(14) secsig(3) algorithm(2)}

-- macros

-- taken from {joint-iso-ccitt ds(5) modules(1) authenticationFramework(7)}
ALGORITHM MACRO         ::=
BEGIN
TYPE NOTATION         ::= "PARAMETER" type
VALUE NOTATION        ::= value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

-- algorithms

md4WithRSA ALGORITHM
    PARAMETER NULL
    ::= {algorithm 2}

md5WithRSA ALGORITHM
    PARAMETER NULL
    ::= {algorithm 3}

md4WithRSAEncryption ALGORITHM
    PARAMETER NULL
    ::= {algorithm 4}

desECB ALGORITHM
    PARAMETER NULL
    ::= {algorithm 6}

desCBC ALGORITHM
    PARAMETER CBCParameter
    ::= {algorithm 7}

CBCParameter ::= IV

desOFB ALGORITHM
    PARAMETER FBParameter
    ::= {algorithm 8}

desCFB ALGORITHM
    PARAMETER FBParameter
    ::= {algorithm 9}
```

```
FBParameter ::= SEQUENCE {
    iv IV,
    numberOfBits NumberOfBits
}

NumberOfBits ::= INTEGER    -- Number of feedback bits (1 to 64 bits)

    Editor's Note - Check FIPS PUB 81 for allowed ranges of feedback
    bits and specify ranges here as a comment.

IV ::= OCTET STRING  -- 8 octets

desMAC ALGORITHM
    PARAMETER MACParameter
    ::= {algorithm 10}

MACParameter ::= INTEGER    -- Length of MAC (16, 24, 32, 40, 40 or 64 bits)
```

Editor's Note - Check FIPS PUB 113 for allowed

```
END -- of Algorithm Object Identifier Definitions
```

~~Annex E (informative)~~

~~Bibliography~~

Annex F (normative)

Bibliography

REFERENCES FOR SECURITY ALGORITHMS

[a] Kaliski, B., The MD2 Message-Digest Algorithm, Internet Draft draft-rsadsi-kaliski-md2-00.txt, July 1, 1991.

[b] Rivest, R. and S. Dusse, The MD4 Message-Digest Algorithm, Internet Draft draft-rsadsi-rivest-md4-00.txt, July 1, 1991.

[c] Rivest, R. and S. Dusse, The MD5 Message-Digest Algorithm, Internet Draft draft-rsadsi-rivest-md5-01.txt, July 10, 1991.

[d] Rivest, R. L., A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, Volume 21, Number 2, February 1978, pp. 120-126.

[e] Rivest, Ronald L., Adi Shamir and Leonard M. Adleman, Cryptographic Communications System and Method, United States Patent No. 4,405,829, September 20, 1983.

[f] Fougner, R.B., Public Key Standards and Licenses, Internet Request for Comments (RFC) 1170, January 1991.

[g] RSA Data Security, Inc., PKCS #1: RSA Encryption Standard, Version 1.4, June 3, 1991.

[h] Diffie, W., and M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, IT-22, pp. 644-654, 1976.

[i] Hellman, Martin E., Bailey W. Diffie and Ralph C. Merkle, Cryptographic Apparatus and Method, United States Patent No. 4,200,770, April 29, 1980.

[j] RSA Data Security, Inc., PKCS #3: Diffie-Hellman Key-Agreement Standard, Version 1.3, June 3, 1991.

[k] ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, IT-31, Number 4, July 1985, pp. 469-472.

[l] Federal Information Processing Standards Publication (FIPS PUB) 46-1, Data Encryption Standard, U.S. Department of Commerce/National Bureau of Standards,

Supersedes FIPS PUB 46, January 15, 1977, Reaffirmed January 22, 1988.

[m] ANSI X3.92-1981, Data Encryption Algorithm, American National Standards Institute, Approved December 30, 1980.

[n] Federal Information Processing Standards Publication (FIPS PUB) 81, DES Modes of Operation, U.S. Department of Commerce/National Bureau of Standards, December 2, 1980.

[o] ANSI X3.106-1983, Data Encryption Algorithm - Modes of Operation, American National Standards Institute, Approved May 16, 1983.

[p] Federal Information Processing Standards Publication (FIPS PUB) 74, Guidelines for Implementing and Using the NBS Data Encryption Standard, U.S. Department of Commerce/National Bureau of Standards, April 1, 1981.

[q] Gait, Jason, Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard, Special Publication 500-20, U.S. Department of Commerce/National Bureau of Standards, Issued November 1977, Revised September 1980.

[r] Gait, Jason, Maintenance Testing for the Data Encryption Standard, Special Publication 500-61, U.S. Department of Commerce/National Bureau of Standards, August 1980.

[s] Federal Information Processing Standards Publication (FIPS PUB) 113, Computer Data Authentication, U.S. Department of Commerce/National Bureau of Standards, May 30, 1985.

[t] American National Standard X9.9-1986, Financial Institution Message Authentication (Wholesale), American Bankers Association, April 7, 1986.

[u] Linn, John, Privacy Enhancement for Internet Electronic Mail: Part I -- Message Encipherment and Authentication Procedures, Internet Draft draft-ietf-pem-msgproc-01.txt, September 1991.

[v] Kent, Steve, Privacy Enhancement for Internet Electronic Mail: Part II -- Certificate-Based Key Management, Internet Draft draft-ietf-pem-keymgmt-00.txt, June 1991.

[w] Balenson, David. M, Privacy Enhancement for Internet Electronic Mail: Part III -- Algorithms, Modes, and Identifiers, Internet Draft draft-ietf-pem-algorithms-00.txt, August 1991.

[x] Kaliski, Burton. S, Privacy Enhancement for Internet Electronic Mail: Part IV -- Notary,

Co-Issuer, CRL-Storing and CRL-Retrieving Services, Internet Draft draft-ietf-pem-notary-00.txt, July 1991.

[y] North American Directory Forum, A Naming Scheme for c = US, Request for Comments 1255, September 1991.

[z] Kohl, John and B. Clifford Neuman, The Kerberos Network Authentication Service, Internet Draft cat-kerberos-00.txt, June 1991.

Annex G (informative)

ElGamal

The information in this subclause includes a tutorial description of the ElGamal scheme for digital signature using the notation defined in the Directory Documents, [ISO9594-8]. It is intended that much of the tutorial information provided in this subclause will be moved to the security agreements sometime in the future.

G.1 Background

The ElGamal digital signature scheme is based on earlier work done by Diffie and Hellman [b] in which it was suggested that a likely candidate for a one-way function is the *discrete exponential function*

$$f(x) = \alpha^x (\mathrm{mod} p) \quad (1)$$

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 $1 \leq \alpha \geq p$ and $\{\alpha \bmod p, \alpha^2 \bmod p, \ldots, \alpha^{p-1} \bmod p\}$ is equal to the set $\{1, 2, \ldots, p\text{-}1\}$. In algebraic terminology, such an α is called a *primitive element*. References on the topic of primitive roots and elements are [aa] and [ab].

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

$$\exp(\sqrt{cb \ln b}) \quad (2)$$

where b is the number of bits in p, and c is estimated at $c = .69$ according to [ac]. 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.

G.2 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

$$X_p = \alpha^{X_s} (\mathrm{mod} p) \quad (3)$$

where

a) p is a prime satisfying the requirements listed in 12.2.2.4.

b) α is a primitive element mod p.

c) Note that p and α could be used globally, but because they should be easily changeable (see 12.2.2.4 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

$$\alpha^{h(m)} = (A_p)^r r^s \pmod{p} \tag{4}$$

where h is a one-way hash function.

Compute the signature (r, s) as follows.

a) 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., $\gcd(k,p-1) = 1$).

b) Compute r such that

$$r = \alpha^k \pmod{p} \tag{5}$$

c) Use r to solve for the corresponding s as follows.

1) rewrite eq (4) using eq (5) and the definition of the public key to get

$$\alpha^{h(m)} = \alpha^{(A_s)r} \alpha^{ks} \pmod{p} \tag{6}$$

Combining exponents, get

$$\alpha^{h(m)} = \alpha^{(A_s)r+ks} \pmod{p} \tag{7}$$

eq (7) implies that

$$h(m) = (A_s)r + ks \pmod{p-1} \tag{8}$$

Note that eq (8) has a single solution for s because k was chosen such that $\gcd(k,p-1) = 1$. See [ad] for supporting theorem.

2) now solve for s and get

$$s = I(h(m) - (A_s)r) \pmod{p-1} \tag{9}$$

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.

G.3 Verification

The recipient receives Ap, m, r, s, a, and p and computes both sides of eq (4) and then compares the results.

G.4 Known Constraints on Parameters

The following list of constraints is the result of a search of current literature and may not be complete:

a) p must be prime;

b) p must be large.

Note that Expression (2) can be used to speculate on the level of security afforded by crypto systems 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.

$$\exp(\sqrt{cb\ln b}) = 3 \times 10^{22} \qquad (10)$$

We get $b \sim 606$. This is the number of bits in the desired p. So, the magnitude of the desired p is about 2^{806} 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 subclause. 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.

c) p should occasionally be changed. This requirement is discussed in [ae] and is related to the discovery of new algorithms for computing discrete logarithms in $GF(p)$.

d) p-1 must have at least one large prime factor. This requirement is discussed in [ae] and is imposed by the Silverman-Pohlig-Hellman algorithm p 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.

e) 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 [af]for details.

~~Annex H (informative)~~

~~ANNEX FOR SECURITY ALGORITHMS~~

```
OIWSECSIGAlgorithmObjectIdentifiers  {iso(1) identified-organization(3)
                                     oiw(14) secsig(3)
                                     oIWSECSIGAlgorithmObjectIdentifiers(1)}
DEFINITIONS ::=
BEGIN

EXPORTS
-- to be determined

IMPORTS
-- none

-- category of information object
-- defining our own here; perhaps the definition should be imported from
-- {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

algorithm          OBJECT IDENTIFIER ::= {iso(1) identified-organization(3)
                                         oiw(14) secsig(3) algorithm(2)}

-- macros

-- taken from {joint-iso-ccitt ds(5) modules(1) authenticationFramework(7)}
ALGORITHM MACRO          ::=
BEGIN
TYPE NOTATION          ::= "PARAMETER" type
VALUE NOTATION         ::= value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

-- algorithms

md4WithRSA ALGORITHM
    PARAMETER NULL
    ::= {algorithm 2}

md5WithRSA ALGORITHM
    PARAMETER NULL
    ::= {algorithm 3}

md4WithRSAEncryption ALGORITHM
    PARAMETER NULL
    ::= {algorithm 4}

desECB ALGORITHM
    PARAMETER NULL
    ::= {algorithm 6}

desCBC ALGORITHM
    PARAMETER CBCParameter
    ::= {algorithm 7}

CBCParameter ::= IV

desOFB ALGORITHM
    PARAMETER FBParameter
    ::= {algorithm 8}

desCFB ALGORITHM
    PARAMETER FBParameter
    ::= {algorithm 9}
```

```
FBParameter ::= SEQUENCE {
    iv IV,
    numberOfBits NumberOfBits
}

NumberOfBits ::= INTEGER  -- Number of feedback bits (1 to 64 bits)

    Editor's Note -- Check FIPS PUB 81 for allowed ranges of feedback
    bits and specify ranges here as a comment.

IV ::= OCTET STRING  -- 8 octets

desMAC ALGORITHM
    PARAMETER MACParameter
    ::= {algorithm 10}

MACParameter ::= INTEGER  -- Length of MAC (16, 24, 32, 40, 48 or 64 bits)
```

Editor's Note -- Check FIPS PUB 113 for allowed

```
END -- of Algorithm Object Identifier Definitions
```

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 13 - OS Security

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: Dr. James Galvin, Trusted Information Systems
SIG Editor: John Hooder, NAVY, Dr. Mohammad Mirhakkak, MITRE

Foreword

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

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject. There is no significant technical change from this text as previously given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

Part 13 - OS Security

0 Introduction

This part is reserved for future stable Security agreements. Previous Stable Agreements may be found in part 12 of this document.

Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 14 - Virtual Terminal

Output from the December 1992 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Luke Lucas, Control Data Systems, Inc.**
SIG Editor: **Scott Wattum, Digital Equipment Corporation**

Foreword

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

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

Three normative annexes are given.

Future changes and additions to this version of these Implementor Agreements will be published as change pages. Deleted and replaced text will be shown as struck. New and replacement text will be shown as shaded.

Table of Contents

List of Tables

Part 14 - Virtual Terminal

0 Introduction

The OSI Implementors' 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.

1 Scope

1.1 Phase Ia agreements

The Telnet profile is intended to support the following usage:

a) a simple line at a time or character at a time dialogue;

b) 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.

1.2 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.

1.3 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.

The S-mode Paged Application profile is intended to provide a Forms capability for the existing base of block mode terminals. It contains a subset of the functionality provided by the Forms profile.

See Working Agreements regarding other Phase II profiles.

2 Normative references

ISO 9040:1990, *Information technology - Open Systems Interconnection - Virtual Terminal Basic Class Service.*

ISO 9041-1:1990, *Information technology - Open Systems Interconnection - Virtual Terminal Basic Class Protocol - Part 1: Specification.*

ISO 9834-4, *Information technology - Open Systems Interconnection*
- Procedures for the Operation of OSI Registration Authorities
- Part 4: Register of VTE Profiles.

ISO 9834-4, *Information technology - Open Systems Interconnection*
- Procedures for the Operation of OSI Registration Authorities
- Part 5: Register of VT Control Object Definitions.

3 Status

This version of the agreements was completed in December 1991.

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.

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.

3.3 Status of phase II

Phase II is still in progress and includes the remaining profile work for the Scroll profile.

The S-mode Paged Application Profile is being progressed as PDISP 11187-2 (AVT-23 S-mode Paged Application Profile).

The X.3 profile of phase II was stabilized December 15, 1989.

The Generalized Telnet profile of phase II was stabilized December 13, 1991.

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

4 Errata

Editor's Note - "Defect Report" material may be included here, including versions of implementor agreements to which it applies.

Table 1 - Technical errata

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.
12/91-1	Telnet Profile. Change x-absolute value from "no" to "yes."
03/92-1	Generalized Telnet Profile. Add conformance statement regarding the requirement to accept negotiation of Suppress GoAhead.
03/92-2	Generalized Telnet Profile. Rework Definitive Note 8, expanding the repertoire negotiation capability to allow negotiation for the use any one of a number of non-binary repertoires.
03/92-3	X.3 Profile. Correct processing of terminal break so that it aligns with the procedures of ccitt X.29.
12/92-1	Generalized Telnet Profile. Rework Definitive Note 8, to clarify repertoire negotiation capability.
12/92-2	Generalized Telnet Profile. Add Informative Note 4, to clarify situations where repertoire negotiation capability beyond switching to binary would be used.
12/92-3	Generalized Telnet Profile. Remove item C from Definitive Note 3.
12/92-4	Generalized Telnet Profile. Clarify action of VT-BREAK in Informative Note 3.

Table 2 - Alignment errata

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.

Table 3 - Editorial errata

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.
12/90-1	X3 Profile. Changes were made to correct editorial errors discovered during the progression of the EWOS X3 Profile Functional Standard.
09/91-1	Scope, Status, and References clauses were updated.
09/92-1	Status clause was updated.
12/92-1	Scope and Status clauses were updated.
12/92-2	Headings and Table entries were updated.
12/92-3	S-mode Paged Application profile. Created Section 8.6 to reference the S-mode Paged Application Profile.
12/92-4	Telnet-1988 profile. A note was added to clarify the future of the profile.

5 Conformance

Conformant VT implementations are required to support the ISO 9041 Clause 13 requirements plus the additional conformance requirements identified below.

Table 4 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:

a) "Mandatory" indicates that the facility must be provided by all implementations which conform to these agreements;

b) "Optional" indicates that the VT facility is not required to meet minimum conformance requirements but has been identified as providing additional useful capabilities;

c) "Profile Dependent" indicates that the requirement for the facility, if any, is included in the profile definitions;

d) "Not Addressed" indicates that the VT facility is outside the scope of these agreements.

Table 4 - Conformance Status for VT Facilities

Conformance Status	Mandatory	Optional	Profile Dependent	Not Addressed
Switch Profile[2)]		X		
Multiple Interaction Negotiation[2)]				X
Negotiated Release[2)]				X
Urgent Data[2)]		X		
Break[2)]	X			
Delivery Control[1)]			X	
Enhanced Access Rules[2)]			X	
Structured COs[2)]			X	
Blocks[2)]				X
Fields[2)]			X	
RIOs[2)]			X	
S-mode			X	
A-mode			X	
Mode Switching Capability		X		

1) It is not anticipated that new profiles will use quarantined delivery control.
2) 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.

6 Protocol

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 Table 4.

All protocol elements required by the Urgent Data functional unit are selected if this functional unit is used. See Table 4.

6.2 Mapping of protocol elements

Mapping of protocol elements on to ACSE or Presentation Services is as defined in ISO 9041.

6.3 Protocol data unit structure

Protocol data unit structure is as defined in ISO 9041.

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.

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.

7.1.1 Entry number

To be supplied by Registration Authority.

7.1.2 Name of sponsoring body

OSI Implementors' Workshop (OIW), VTSIG.

7.1.3 Date

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

7.1.4 Identifier

oiw-vt-co-misc-sa OBJECT IDENTIFIER ::= {oiw-vt-co-misc sa(0)}

7.1.5 Descriptor value

"OIW VT CO for conveying Sequenced Application Signals"

7.1.6 CO parameters

CO-structure	1
CO-priority	"normal"
CO-category	"symbolic"
CO-size	11

7.1.7 CO Values and Semantics

Table 5 lists the allowed symbolic values together with the integers used to reference these values in the ASN.1 update syntax of ISO 9041:

Table 5 - SA/UA CO values and semantics.

Symbolic Value	Integer Value
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.

7.1.8 Additional information

None.

7.1.9 Usage

Defined in profile.

7.2 Unsequenced Application (UA)

This is a Control object used to convey urgent signals from the application to the terminal.

7.2.1 Entry number

To be supplied by Registration Authority.

7.2.2 Name of sponsoring body

OSI Implementors' Workshop (OIW), VTSIG.

7.2.3 Date

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

7.2.4 Identifier

oiw-vt-co-misc-ua OBJECT IDENTIFIER::= {oiw-vt-co-misc ua(1)}

7.2.5 Descriptor value

"OIW VT CO for conveying Unsequenced Application Signals"

7.2.6 CO parameters

CO-structure	1
CO-priority	"urgent"
CO-category	"symbolic"
CO-size	11

7.2.7 CO values and semantics

Same as in SA.

7.2.8 Additional information

None.

7.2.9 Usage

Defined in profile.

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.

7.3.1 Entry number

To be supplied by Registration Authority.

7.3.2 Name of sponsoring body

OSI Implementors Workshop (OIW), VTSIG.

7.3.3 Date

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

7.3.4 Identifier

oiw-vt-co-misc-st OBJECT IDENTIFIER ::= {oiw-vt-co-misc st(2)}

7.3.5 Descriptor value

"OIW VT CO for conveying Sequenced Terminal Signals"

7.3.6 CO parameters

CO-structure 1
CO-priority "normal"
CO-category "Integer"
CO-size 65535

7.3.7 CO values and semantics

The values of the CO are composite, with values from Table 6 giving meaning to the values in the hex range 00-FF when added to them.

Table 6 - ST/UT CO composite values

hex value	meaning
100	special key - labeled[1)]
400	function key depressed
400	control key depressed
800	shift key depressed
1000	alt key depressed
1) possible special key values are as defined by the STCO ASN.1 module.	

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.

7.3.8 Additional information

None.

7.3.9 Usage

Defined in profile.

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.

7.4.1 Entry number

To be supplied by the Registration Authority.

7.4.2 Name of sponsoring body

OSI Implementors Workshop (OIW), VTSIG

7.4.3 Date

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

7.4.4 Identifier

oiw-vt-co-misc-ut OBJECT IDENTIFIER ::= {oiw-vt-co-misc ut(3)}

7.4.5 Descriptor value

"OIW VT CO for conveying Unsequenced Terminal Signals"

7.4.6 CO parameters

CO-structure	1
CO-priority	"urgent"
CO-category	"integer"
CO-size	65535

7.4.7 CO values and semantics

Same as in ST.

7.4.8 Additional information

None.

7.4.9 Usage

Defined in profile.

8 OIW defined profiles

These profiles are defined using the conventions specified in Annex A of ISO 9040.

8.1 Telnet profile

OIW VTE-Profile Telnet-1988 (r1, r2)

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.

Note: This profile is superceded by the Generalized-Telnet profile. The text for this profile will not be maintained beyond its current state.

8.1.2 Association requirements

8.1.2.1 Functional units

The Urgent Data Functional Unit is optional, but should be used whenever available.

8.1.2.2 Mode

This is an A-mode profile.

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   = "yes",      *(See Definitive Note 4)*
                        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   = "yes",      *(See Definitive Note 4)*
                        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."

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.

8.1.5 Profile dependent control object information

This profile does not reference any Control Objects which are not defined within this profile.

8.1.6 Profile notes

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:

Table 7 - KB/DI CO value definitions

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 in Table 8.

Table 8 - NI/NA CO value definition

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. This is the only instance when backward explicit addressing is permitted.

The TELNET EL (erase line) command 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. This is the only instance when absolute 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.

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.

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).

The transparent character set, see Definitive Note 8 in clause 8.1.6.1.

8.2 Transparent profile

OIW VTE-Profile Transparent-1988 (r1)

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.

8.2.2 Association requirements

8.2.2.1 Functional units

No additional functional units are required by this profile.

8.2.2.2 Mode

This is an A-mode profile.

8.2.3 Profile body

```
Display-objects *(double occurrence)* =
{
        {
        display-object-name  = D1,
        do-access            = "WACA",
```

```
        dimensions              = "one",
                x-dimension     =
                        {
                        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."
```

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.

8.2.5 Profile dependent control object information

This profile does not reference any Control Objects.

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.

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.

8.3 Forms profile

OIW VTE-Profile Forms-1989 (r1,r2, . . . r28)

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:

a) defining and using screen forms;

b) defining field validation and field entry rules;

c) controlling and validating field entry.

This VTE-profile includes support of an optional terminal-end locally attached printer.

8.3.2 Association requirements

8.3.2.1 Functional units

The following VT functional units are required for operation with this profile:

a) Enhanced access-rules;

b) Structured COs;

c) Fields;

d) Reference Information Objects.

The following VT functional units are optional for operation with this profile:

Urgent Data

8.3.2.2 Mode

This is an S-mode profile.

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 clause 7)*
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 clause 7)*

```
        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 clause 7)*
        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 clause 7)*
        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."
```

8.3.4 FIXED field entry instruction definitions - non-parametric

8.3.4.1 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.

8.3.4.2 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.

8.3.4.3 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.

8.3.4.4 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.

8.3.4.5 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.

8.3.4.6 Echo off

Received field entry characters should not be echoed. This FEI can never be violated.

8.3.4.7 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.

8.3.4.8 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.

8.3.5 DYNAMIC field entry instruction definitions - parametric

8.3.5.1 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.

8.3.5.2 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.

8.3.5.3 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.

8.3.5.4 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.

8.3.5.5 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.

8.3.5.6 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.

8.3.5.7 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.

8.3.5.8 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.

8.3.5.9 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.

8.3.5.10 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.

8.3.6 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. Table 9 defines the sets of conflicting FEIs.

Table 9 - 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

8.3.7 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)*
```

8.3.8 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.

Table 10 - FEICO "mandatory-feico" initial content

FEIR	Value
FEIR00	-- not used --
FEIR01	optionalField
FEIR02	mandatoryField
FEIR03	selectableField { }
FEIR04	protectedField
FEIR05	fillField
FEIR06	echoReceivedCharacter
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 --

8.3.9 Field entry event definitions

The Field Entry Events for the mandatory FEPCO are defined in the following subclauses. 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.

8.3.9.1 FEE00

Not used.

8.3.9.2 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.

8.3.9.3 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.

8.3.9.4 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.

8.3.9.5 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.

8.3.9.6 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.

8.3.10 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 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.

8.3.10.1 FEC00

Not used.

8.3.10.2 FEC01 no previous field

The current field has no currently defined previous field, in the sense of 20.3.3.4 of ISO 9040.

8.3.10.3 FEC02 no next field

The current field has no currently defined next field, in the sense of 20.3.3.4 of ISO 9040.

8.3.10.4 FEC03 start of field

The current location for the next character entry is at the first location in the current field.

8.3.10.5 FEC04 end of field

The current location for the next character entry is at the last location in the current field.

8.3.10.6 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.

8.3.10.7 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.

8.3.10.8 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.

8.3.10.9 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.

8.3.10.10 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.

8.3.10.11 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.

8.3.10.12 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.

8.3.10.13 FEC12 not FEC (FEC)

This condition is satisfied precisely when the FEC given as its parameter is not satisfied.

8.3.10.14 FEC13 and FECs (set of FEC)

This condition is satisfied when each of the conditions in the set comprising its parameter is satisfied.

8.3.10.15 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.

8.3.10.16 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.

8.3.10.17 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.

8.3.11 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.

8.3.11.1 FER00

Not used.

8.3.11.2 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.

8.3.11.3 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.

8.3.11.4 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.

8.3.11.5 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.

8.3.11.6 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.

8.3.11.7 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.

8.3.11.8 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.

8.3.11.9 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.

8.3.11.10 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.

8.3.11.11 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.

8.3.11.12 FER11 visual indication

Present a visual indication in response to Field Entry Instruction violations.

8.3.11.13 FER12 audible indication

Present an audible indication in response to Field Entry instruction violations.

8.3.11.14 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.

8.3.11.15 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.

8.3.11.16 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.

8.3.11.17 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.

8.3.12 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 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.

Table 11 - FEPCP "mandatory-fepco" initial content

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 --

8.3.13 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-nullrlo 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.

8.3.14 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.

8.3.14.1 Sequenced Application CO

This Control Object is defined in 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.

8.3.14.2 Unsequenced Application CO

This Control Object is defined in 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.

8.3.14.3 Sequenced Terminal CO

This Control Object is defined in 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.

8.3.14.4 Unsequenced Terminal CO

This Control Object is defined in 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.

8.3.15 Profile notes

8.3.15.1 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 8.3.4 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

Subattribute "h" values

= "F"	51	Framed

= "C"	52	Encircled
= "N" *	54	Not framed, not encircled
= " "	-	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 7.3 and 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 12 - 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 12 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 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, 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 8.3.10.

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.

8.3.15.2 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.

8.3.16 Specific conformance requirements

For further agreement.

8.4 X3 profile

OIW VTE-Profile X3-1989 (r1, r2, r3, r4, r5, r6)

8.4.1 Introduction

This profile provides support for CCITT X.3 PAD compatible operation.

The purpose of this profile is two-fold:

a) 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;

b) to facilitate a gateway function between ISO-VTP and X.3.

8.4.2 Association requirements

8.4.2.1 Functional units

The Structured CO Functional Unit is mandatory.

The Urgent Data Functional Unit is optional.

8.4.2.2 Mode

This is an A-mode profile.

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 the 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 action 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 C0 )*
                             = "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 C0 )*
                             = "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 C0 )*
                             = "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 = "NSAC",
CO-priority = "urgent",
CO-trigger = "not-selected",
CO-category = "symbolic",
CO-size = 2
},

{ *(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 = "NSAC",
CO-priority = "normal",
CO-trigger = "selected",
CO-category = "symbolic",
CO-size = 2
},

```
{ *( 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 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-element-id = 1,
CO-category   = "character",
CO-repertoire-assignment *( VTS Transparent Set )*
              = <ESC> 2/5 2/15 4/2,
CO-size       = 4 },
{       *( User Data )*
CO-element-id = 2,
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 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 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."
```

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. If the terminal-system, i.e., that which includes the equivalent of the PAD function, establishes the VTE-profile then the value of r1 should be "WACI." If the system not including the PAD function establishes the VTE-profile then the value of r1 should be "WACA." 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."

8.4.5 Profile notes

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:

Table 13 - PAD CO data element 1 value definition

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:

Table 14 - PAD CO data element 3 value definition

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:

Table 15 - PAD CO data element 7 value definition

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.

When boolean 1 is "true" and booleans 3 and 5 are "false" and a Break is received from the device, the terminal system updates the BO CO with the symbolic value "alone."

When booleans 1 and 3 are "true" and boolean 5 is "false" and a Break is received from the device, the terminal system updates the BO CO with the symbolic value "prepare" followed by an update to the BI CO with the symbolic value "unconfirmed."

When booleans 1, 3 and 5 are all "true" and a Break is received from the device, the terminal system updates the BO CO with the symbolic value "prepare" followed by an update to the BI CO with the symbolic value "confirmed" and discards all display object updates from the application system until it receives an update to the PAD CO selecting element-id 8.

If boolean 1 is "false," then booleans 3 and 5 must be "false."

If boolean 3 is "false," then boolean 5 must be "false."

Table 16 - BI CO values and semantics

Symbolic Value	Integer Value
unconfirmed	0
confirmed	1

Table 17 - BO CO values and semantics

Symbolic Value	Integer Value
alone	0
prepare	1

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:

Table 18 - PAD CO data element 13 value definition

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:

Table 19 - PAD CO data element 19 value definitions

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 20 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:

Table 20 - PAD CO data element 20 value definition

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), 0/8 (BS)
5	Do not echo 1/11 (ESC), 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), 0/3 (ETX)
7	Do not echo the editing characters defined by data elements 16, 17, and 18 of the PAD CO
8	Do not echo 7/15 (DEL) or any of the other characters belonging to C0 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:

Table 21 - PAD CO data element 21 value definition

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 to 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 update 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.

26. The application system can update the BI CO and the terminal system shall send a Break to the device. If the symbolic value of the update is "confirmed," the terminal system shall respond with an update to the PAD CO selecting element-id 8.

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:

Table 22 - CCITT Simple Standard profile

data element	value	meaning
1	1	possible to return control to 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 may 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 may 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.

Table 23 - CCITT Transparent Standard profile

data element	value	meaning
1	0	control may not be returned to the terminal-system
2	1."false"	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 the device
6	1."false"	service signals are never 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."false"	device may not 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

8.4.6 Specific conformance requirements

None.

8.5 Generalized Telnet profile

OIW VTE-Profile Generalized Telnet-1991 (r1,r2)

8.5.1 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. This profile references the ARPA Internet TELNET standards documents for the semantics of option negotiation and the values of symbolic constants.

8.5.2 Association requirements

8.5.2.1 Functional units

The Structured Control Objects Functional Unit is required. The Urgent Data Functional Unit is optional, but should be used whenever available.

8.5.2.2 Mode

This is an A-mode profile.

8.5.3 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    = "yes",      *(See Definitive Note 5)*
                        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   = *(implicitly defined by r2)*,
        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      = "yes",        *(See Definitive Note 5)*
                        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   = *(implicitly defined by r2)*,
        repertoire-assignment           = profile-argument-r2,
        repertoire-assignment           = <ESC> 2/5 2/15 4/2
        },
},

Control-objects = *(multiple occurrence)*
{
        { *(SYNCHRONIZE)*
                CO-name         = SY,
                CO-category     = "symbolic",
                CO-access       = "NSAC",
                CO-size         = 1,
                CO-priority     = "urgent"
        },
        { *(DISPLAY-SIGNAL)*
                CO-name         = DI,
                CO-category     = "symbolic",
                CO-size         = 255,
                CO-access       = "WACA",
```

```
        CO-priority     = "normal",
        CO-trigger      = "selected"
    },
    { *(KEYBOARD-SIGNAL)*
        CO-name         = KB,
        CO-category     = "symbolic",
        CO-size         = 255,
        CO-access       = "WACI",
        CO-priority     = "normal",
        CO-trigger      = "selected"
    },
    { *(NEGOTIATION BY INITIATOR)*
        CO-name         = NI,
        CO-structure    = 2,
            *(DO/DONT)*
            CO-element-id   = 1,
            CO-category     = "boolean",
            CO-size         = 256,
            *(WILL/WONT)*
            CO-element-id   = 2,
            CO-category     = "boolean",
            CO-size         = 256,
        CO-access       = "WACI",
        CO-priority     = "normal",
        CO-trigger      = "selected"
    },
    { *(NEGOTIATION BY ACCEPTOR)*
        CO-name         = NA,
        CO-structure    = 2,
            *(DO/DONT)*
            CO-element-id   = 1,
            CO-category     = "boolean",
            CO-size         = 256,
            *(WILL/WONT)*
            CO-element-id   = 2,
            CO-category     = "boolean",
            CO-size         = 256,
        CO-access       = "WACA",
        CO-priority     = "normal",
        CO-trigger      = "selected"
    },
    { *(SUBNEGOTIATION BY INITIATOR)*
        CO-name         = SBI,
        CO-structure    = 2,
            *(TELNET OPTION)*
            CO-element-id   = 1,
```

```
                CO-category     = "symbolic",
                CO-size         = 256,
                *(SUBNEGOTIATION)*
                CO-element-id   = 2,
                CO-category     = "character",
                CO-repertoire-assignment = <ESC> 2/5 2/15 4/2,
                *(Virtual Terminal Service Transparent Set)*
                CO-size         = 1024,
        CO-access       = "WACI",
        CO-priority     = "normal",
        CO-trigger      = "selected"
    },
    { *(SUBNEGOTIATION BY ACCEPTOR)*
        CO-name         = SBA,
        CO-structure    = 2,
                *(TELNET OPTION)*
                CO-element-id   = 1,
                CO-category     = "symbolic",
                CO-size         = 256,
                *(SUBNEGOTIATION)*
                CO-element-id   = 2,
                CO-category     = "character",
                CO-repertoire-assignment = <ESC> 2/5 2/15 4/2,
                *(Virtual Terminal Service Transparent Set)*
                CO-size         = 1024,
        CO-access       = "WACA",
        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   = SBA,
            *(SYNC, NEGOTIATE-ACCEPTOR,
            DISPLAY-SIGNAL, SUBNEGOTIATE-ACCEPTOR)*
    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-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   = SBI,
                *(SYNC, NEGOTIATE-INITIATOR,
                KEYBOARD-SIGNAL, SUBNEGOTIATE-INITIATOR)*
        device-default-CO-access        = "WACI",
        device-default-CO-priority      = "normal"
        *(other device object parameters assume corresponding
        DO values)*
        }
Type of delivery control = "simple-delivery-control."
```

8.5.4 Profile argument definitions

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 repertoires for both display objects. This argument is optional, and may occur a number of times in an ordered list to provide for negotiation of values 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. The VTE-parameter repertoire-capability equals the number of occurrences of this profile argument plus one. The default is a single occurrence of the value designating the full U.S. ASCII set. This argument is identified by the identifier for repertoire assignment for display object D.

8.5.5 Profile dependent CO information

8.5.6 Profile notes

8.5.6.1 Definitive notes

1. Sending a KB or DI control object update is the equivalent of sending a TELNET "IAC <command>" sequence. The symbolic value in the KB or DI control object update is equal to the TELNET command code as specified in the TELNET Assigned Numbers.

   The following values must be recognized:

SYMBOLIC	NAME	VALUE
DM	Data Mark	242
BRK	Break	243
IP	Interrupt Process	244
AO	Abort output	245
AYT	Are You There	246
GA	Go ahead	249

   The following values, corresponding to TELNET commands, are excluded from KB and DI control object updates:

SYMBOLIC	NAME	VALUE
SE	End Subnegotiation	240
EC	Erase character	247
EL	Erase Line	248
SB	Subnegotiation	250
WILL	Will	251
WONT	Won't	252
DO	Do	253
DONT	Don't	254
IAC	Escaped IAC	255

   The NI and NA control objects are used in place of the DO, DONT, WILL, WONT commands.

   The SBI and SBA control objects are used in place of the SB <suboptions> SE command sequence.

   The EC and EL commands are replaced by display object updates.

   The IAC is not needed because commands are not embedded in the text.

The recognition of values corresponding to TELNET commands defined in a TELNET option will be dependent upon the successful negotiation of the TELNET option that defines the additional TELNET command. Unrecognized values shall be ignored.

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 with symbolic value DM. 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 is received with symbolic value DM. If a VT-BREAK is received after an SY CO update has been received and prior to the corresponding DI or KB CO update with symbolic value of DM, 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 the TELNET specification 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;

    NI and NA are structured control objects consisting of two boolean data elements. For full symmetry, both NI and NA have the same value definitions. The first boolean data element stands for DO/DONT and the second boolean data element stands for WILL/WONT. The ordinal position of the boolean value in the data element corresponds to the TELNET option number plus one. This allows the ordinal position of bits 1-256 in the boolean object to represent the TELNET options values of 0-255. DO is represented as a "true" boolean value in CO-element-id 1. DONT is represented as a "false" boolean value in CO-element-id 1. WILL is represented as a "true" boolean value in CO-element-id 2. WONT is represented as a "false" boolean value in CO-element-id 2.

4. The SBI and SBA control objects provide subnegotiation for TELNET options, and correspond to the TELNET command sequence "IAC SB <TELNET option code> <subnegotiation> IAC SE". Element id 1 contains the TELNET option code, and element id 2 contains the octets that comprise the subnegotiation. The specification for the TELNET option defines the semantics of the value in element id 2.

5. The TELNET EC (erase character) command will be mapped to a pointer relative (x: = x-1) update and an erase current update. This is the only instance when backward explicit addressing is permitted.

    The TELNET EL (erase line) command 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 pointerupdate to set x = 1. This

is the only instance when absolute explicit addressing is permitted.

6. 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.

7. The VT next X-array update operation will be sent in place of the TELNET NVT "CR, LF" sequence.

8. When a party wants to change the repertoire assignment for the display object to which it has access, it does so by issuing an update to the modal value for the character repertoire attribute. It should be noted that no mechanism exists for a party to request a change of repertoire assignment for the display object to which it does not have access. In the specific case of the TCP/IP TELNET binary option, you must first complete a successful TELNET negotiation to do so. Then the party with the access rights to the display object in question is required to perform the secondary attribute modal update. If a negotiation to change the "binary" repertoire is refused, the current repertoire will remain in effect. When a negotiation to quit using the "binary" repertoire succeeds, the party with the access rights to the display object in question is required to perform the corresponding secondary attribute modal update to switch to the explicit modal default value.

9. While the "binary" repertoire is being used no mapping to the pointer addressing or erase operations will be done.

10. 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 ISO 2022 defined character set designating escape sequences <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.

11. 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 other 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.

8.5.6.2 Informative notes

1. Users of this profile should refer to the TELNET specification (MIL-STD-1782) and RFCs:

   854 Protocol Specification
   855 Options Specification

   or their successors 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. This profile is derived from the Telnet-1988 profile. The negotiation control objects, NA and NI, have been changed to model the DO/DONT WILL/WONT negotiation of TELNET options. The size of the elements of the NA and NI negotiation control objects equals the range of TELNET option numbers, including the numbers presently assigned and those reserved for future options. An implementation can refuse options that it doesn't support. This allows implementations to maintain interoperability while new TELNET options are incorporated. The CO-category of the KB and DI control objects have been changed from "boolean" to "symbolic." A "Go-Ahead" will be signaled by a control object update to the DI or KB control object with symbolic value of GA; therefore, the GA control object has been dropped.

3. If the "go ahead" facility has been negotiated then following a VT-BREAK, only the association acceptor has the right to send data. VT-BREAK causes all negotiated TELNET option values to be reset to their initial values. If negotiated values are to be restored, they must be renegotiated.

4. It is recognized that some implementations may have character set repertoire requirements that are currently outside the scope of TCP/IP TELNET. Implementations such as Gateways that require access to these facilities via non-standard extensions to TCP/IP TELNET will be able to maintain conformance with VT by utilizing profile argument R2 and mapping in both directions between repertoire change requests and appropriate modal attribute updates.

8.5.7 Specific conformance requirements

The following character sets are required:

- The G0 character set for U.S. ASCII (values 32-126);
- The full U.S. 7-bit ASCII (values 0-127);
- The transparent character set, see Definitive Note 8 in section 14.8.5.6.1.

Negotiation to Suppress GoAhead must be accepted.

8.6 S-mode paged application profile

See PDISP 11187-2 (AVT-23 S-mode Paged Application Profile).

Annex A (normative)

Specific ASE requirements

For specific ASE Requirements identified by the Upper Layer SIG for Virtual Terminals, see Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 5 - Upper Layers.

Annex B (normative)

Clarifications

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.

Annex C (normative)

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) }
```

Profiles 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) }

oiw-vt-pr-generalizedTelnet     OBJECT IDENTIFIER ::=
        { oiw-vt-pr     generalizedTelnet(5) }
```

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) }
```