U.S. Department of Commerce
National Bureau of Standa

NBS PUBLICATIONS

A11102 748484

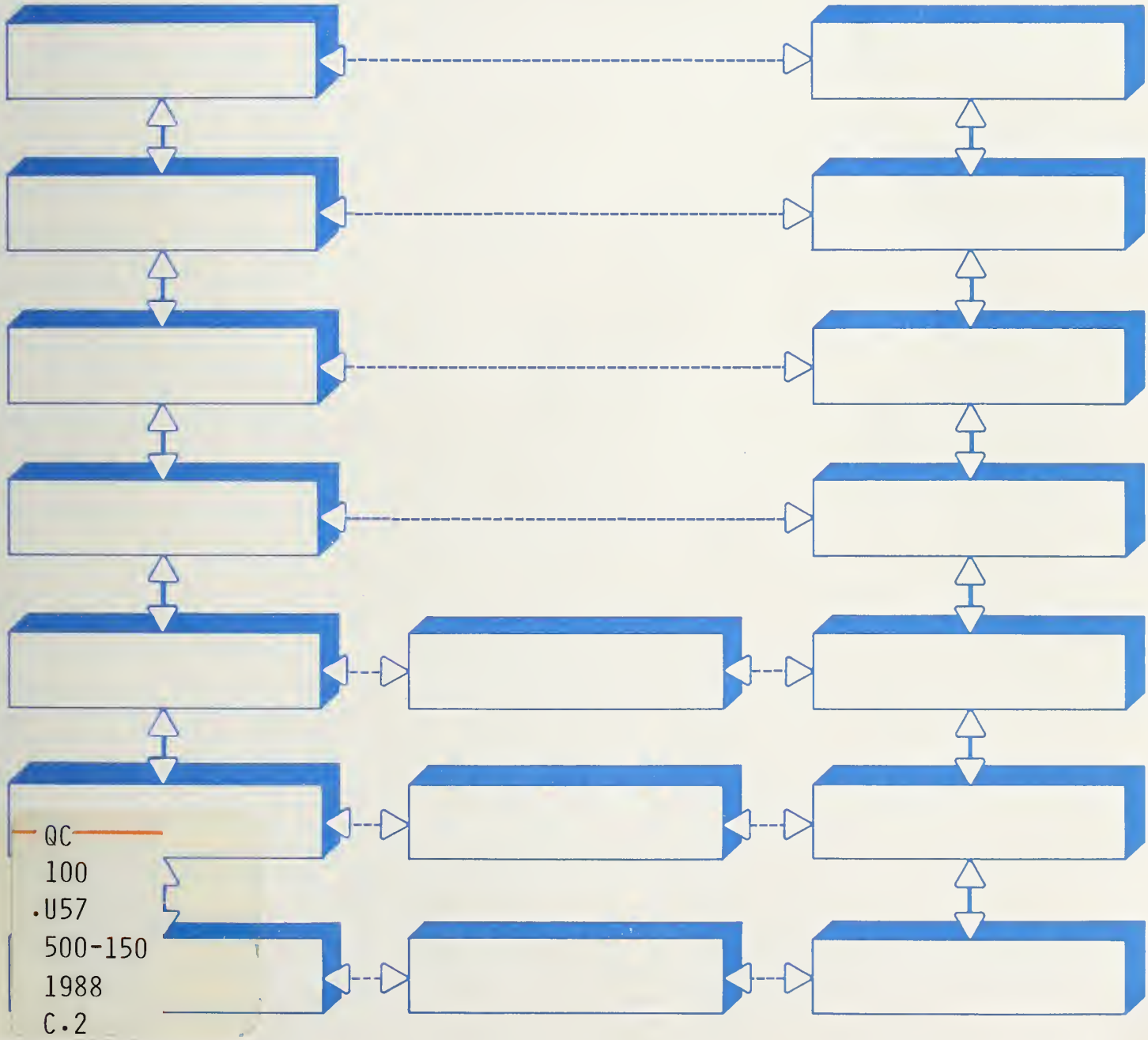
NAT'L INST OF STANDARDS & TECH R.I.C.
A11102748484
/Stable Implementation agreements for op
QC100 .U57 NO.500-150 1988 V19 C.1 NBS-P

Computer Science and Technology

l Publication 500-150

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 1 Edition 1 December 1987

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



QC
100
.U57
500-150
1988
C.2

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research to assure international competitiveness and leadership of U.S. industry, science and technology. NBS work involves development and transfer of measurements, standards and related science and technology, in support of continually improving U.S. productivity, product quality and reliability, innovation and underlying science and engineering. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; provides calibration services; and manages the National Standard Reference Data System. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[3]

The Institute for Computer Sciences and Technology

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following divisions:

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

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-cutting scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Ceramics
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2]Some divisions within the center are located at Boulder, CO 80303.
[3]Located at Boulder, CO, with some elements at Gaithersburg, MD

Research Information Center
National Bureau of Standards
Gaithersburg, Maryland 20899

NBSE
QC100
.U57
no.500-150
1987
c 2

Computer Science And Technology

NBS Special Publication 500-150

Stable Implementation Agreements for Open Systems Interconnection Protocols Version 1 Edition 1 December 1987

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

Workshop Chairman
Robert Rosenthal, NBS

Special Interest Group (SIG) Chairmen

Directory Services	J.J. Cinecoe	Wang
FTAM	Klaus Truoel	DFN
Lower Layers	Michael Gering	IBM
Management	Paul Brusil	MITRE
ODA/ODIF	Frank Dawson	IBM
OSINET	Edward Strum	IBM
Security	Dennis Branstad	NBS
Upper Layers	Michael Ellis	HP
Virtual Terminal	Rick Wilder	MITRE
X400	John Stidd	Xerox

U.S. Department of Commerce
C. William Verity, Secretary

National Bureau of Standards
Ernest Ambler, Director

Issued January 1988

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 88-600500
National Bureau of Standards Special Publication 500-150
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-150, 257 pages (Jan. 1988)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1988

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

Table of Contents

1. GENERAL INFORMATION 1
1.1 PURPOSE OF THIS DOCUMENT 1
1.2 PURPOSE OF THE WORKSHOP 1
1.3 WORKSHOP ORGANIZATION 2

2. SUB NETWORKS 1
2.1 INTRODUCTION 1
2.2 SCOPE AND FIELD OF APPLICATION 1
2.3 STATUS 1
2.4 ERRATA 1
2.5 LOCAL AREA NETWORKS 1
2.5.1 IEEE 802.2 LOGICAL LINK CONTROL 1
2.5.2 IEEE 802.3 CSMA/CD ACCESS METHOD 2
2.5.3 IEEE 802.4 TOKEN BUS ACCESS METHOD 3
2.5.4 IEEE 802.5 TOKEN RING ACCESS METHOD 4
2.6 WIDE AREA NETWORKS 5
2.6.1 CCITT RECOMMENDATION X.25 5

3. NETWORK LAYER 1
3.1 INTRODUCTION 1
3.2 SCOPE AND FIELD OF APPLICATION 1
3.3 STATUS 1
3.4 ERRATA 1
3.5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS) 1
3.5.1 Provision of CLNS over Local 1
3.5.2 Provision of CLNS over X.25 Subnetworks 1
3.5.3 Agreements on Protocols 2
3.5.3.1 ISO 8473 2
3.5.3.2 Subnetwork Dependent Convergence Function . . 3
3.6 CONNECTION-MODE NETWORK SERVICE (CONS) 3
3.6.1 Mandatory Method of Providing CONS 3
3.6.2 Additional Option: Provision of CONS over X.25 1980 Subnetworks 3
3.6.3 Agreements on Protocols 4
3.6.3.1 ISO 8878 4
3.6.3.2 Subnetwork Dependent Convergence Protocol (ISO 8878/Annex A) 4
3.7 ADDRESSING 4
3.8 ROUTING 5
3.8.1 Static Routing 5
3.8.2 End System to Intermediate System 5
3.9 MIGRATION CONSIDERATIONS 6
3.9.1 X.25-1980 6
3.10 CONFORMANCE 7

4. TRANSPORT 1
4.1 INTRODUCTION 1
4.2 SCOPE AND FIELD OF APPLICATION 1
4.3 STATUS 1
4.4 ERRATA 1

4.5 TRANSPORT CLASS 4 1
4.5.1 Transport Class 1
4.5.2 Protocol Agreements 1
4.5.2.1 Rules for Negotiation 1
4.5.2.2 TRANSPORT CLASS 4 SERVICE ACCESS POINTS OR SELECTORS 3
4.5.2.3 Retransmission Timer 3
4.5.2.4 Keep-Alive Function 4
4.6 TRANSPORT CLASS 0 6
4.6.1 Transport Class 0 Overview 6
4.6.2 Protocol Agreements 6
4.6.2.1 TRANSPORT CLASS 0 SERVICE ACCESS POINTS . . . 6
4.6.3 Rules for Negotiation 6

5. UPPER LAYERS 1
5.1 INTRODUCTION 1
5.1.1 References 1
5.2 SCOPE AND FIELD OF APPLICATION 1
5.3 STATUS 1
5.4 ERRATA 2
5.4.1 ISO Defect Reports 2
5.4.2 Session Defects 2
5.5 ASSOCIATION CONTROL SERVICE ELEMENT 2
5.5.1 Introduction 2
5.5.2 Services 3
5.5.2.1 ACSE Services 3
5.5.2.2 Use of Presentation Layer Services 3
5.5.3 Protocol agreements 3
5.5.3.1 Application Context 3
5.5.3.2 Implementation Information 4
5.5.3.3 AE Title 4
5.6 PRESENTATION 4
5.6.1 Introduction 4
5.6.2 Services 5
5.6.2.1 Presentation Services 5
5.6.2.2 Use of Session Layer Services 6
5.6.3 Protocol Agreements 6
5.6.3.1 Transfer Syntaxes 6
5.6.3.2 Abstract Syntaxes 6
5.6.3.3 Presentation Context Identifier 6
5.6.3.4 Mode-selector Position in SET 7
5.6.3.5 Default Context 7
5.6.3.6 P-Selectors 7
5.6.3.7 Provider Abort Parameters 7
5.6.3.8 Provider Aborts and Session Version 7
5.6.3.9 CPC-type 7
5.6.3.10 Presentation-context-definition-result-list . 8
5.6.3.11 RS-PPDU 8
5.6.4 Presentation ASN.1 Encoding Rules 8
5.6.4.1 Invalid Encoding 8
5.6.4.2 Protocol-version, Presentation-requirements . 8
5.6.4.3 Presentation-selector 8

5.7 SESSION 8
5.7.1 Introduction 8
5.7.2 Services 9
5.7.2.1 Session Services 9
5.7.2.2 Use of Transport Services 9
5.7.3 Protocol Agreements 9
5.7.3.1 Concatenation 9
5.7.3.2 Segmenting 10
5.7.3.3 Reuse of Transport Connection 10
5.7.3.4 Use of Transport Expedited Data 10
5.7.3.5 Use of Session Version Number 10
5.7.3.6 Receipt of Invalid SPDUs 11
5.7.3.7 Invalid SPM Intersections 11
5.7.3.8 S-Selectors 11
5.8 Universal ASN.1 Encoding Rules 12
5.8.1 Tags 12
5.8.2 Definite length 12
5.8.3 EXTERNAL Type 12
5.9 CONFORMANCE 12
5.10 APPENDIX A: RECOMMENDED PRACTICES 13

6. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 2 1
6.1 INTRODUCTION 1
6.2 SCOPE AND FIELD OF APPLICATION 1
6.3 STATUS 2
6.4 ERRATA 2
6.5 ASSUMPTIONS 2
6.6 PRESENTATION AGREEMENTS 4
6.7 SERVICE CLASS AGREEMENTS 4
6.8 FUNCTIONAL UNIT AGREEMENTS 4
6.9 FILE ATTRIBUTE AGREEMENTS 4
6.9.1 Mandatory Group 5
6.9.2 Optional Groups 6
6.10 DOCUMENT TYPE AGREEMENTS 6
6.10.1 Character Sets 9
6.10.1.1 IA5 Character Set 10
6.10.1.2 8859-1 Character Set 12
6.10.2 Document Type Negotiation Rules 12
6.10.2.1 Connection Establishment 12
6.10.2.2 File Creation 12
6.10.2.3 File Opening 12
6.10.3 Relationship Between DUs, DEs and Document Types . 13
6.11 F-CANCEL ACTION 14
6.12 IMPLEMENTATION INFORMATION AGREEMENTS 14
6.13 DIAGNOSTIC AGREEMENTS 14
6.14 CONCURRENCY 16
6.15 REQUESTED ACCESS 16
6.16 SECURITY 17
6.16.1 Optional Password Support 17
6.16.2 Access Passwords 17
6.16.3 Implementation Responsibilities 17
6.17 REQUIREMENT FOR CONFORMANT IMPLEMENTATIONS 17

6.17.1 Interoperable Configurations 18
6.17.2 Relationship to ISO 8571--The FTAM Standard . . . 19
6.17.3 Requirements for Document Type Support 19
6.17.4 Initiators 20
6.17.5 Responders 21
6.17.6 Senders 22
6.17.6.1 Initiator Senders 23
6.17.6.2 Responder Senders 23
6.17.7 Receivers 23
6.17.7.1 Initiator Receivers 24
6.17.7.2 Responder Receivers 24
6.17.8 Minimum Ranges 24
6.18 IMPLEMENTATION PROFILES 27
6.18.1 General Requirements for the Defined Implementation Profiles 28
6.18.2 Use of Lower Layer Services 28
6.18.3 Document Type Requirements for the Defined Implementation Profiles 28
6.18.4 Parameters for the Defined Implementation Profiles 29
6.18.5 Parameter Ranges for the Defined Implementation Profiles 30
6.18.6 File Attribute Support for Implementations 30
6.19 PROVISION OF SPECIFIC FUNCTION 33
6.19.1 Implementation Profile T1: Simple File Transfer . 33
6.19.2 Implementation Profile T2: Positional File Transfer 33
6.19.3 Implementation Profile T3: Full File Transfer . . 34
6.19.4 Implementation Profile A1: Simple File Access . . 34
6.19.5 Implementation Profile A2: Full File Access . . . 35
6.19.6 Implementation Profile M1: Management 36
6.20 HARMONIZATION 36
6.21 APPENDIX A: FTAM DOCUMENT TYPES 37

7. CCITT 1984 X.400 BASED MESSAGE HANDLING SYSTEM 1
7.1 INTRODUCTION 1
7.2 SCOPE 2
7.3 STATUS 3
7.4 ERRATA 3
7.5 PRMD to PRMD 3
7.5.1 Introduction 3
7.5.2 Service Elements and Optional User Facilities . . 4
7.5.2.1 Classification of Support for Services . . . 4
7.5.2.1.1 Support (S) 5
7.5.2.1.2 Non Support (N) 5
7.5.2.1.3 Not Used (N/U) 6
7.5.2.1.4 Not Applicable (N/A) 6
7.5.2.2 Summary of Supported Services 6
7.5.2.3 MT Service Elements and Optional User Facilities 6
7.5.2.4 IPM Service Elements and Optional User Facilities 8
7.5.3 X.400 Protocol Definitions 9
7.5.3.1 Protocol Classification 10

7.5.3.2 General Statements on Pragmatic Constraints . 10
7.5.3.3 MPDU Size 11
7.5.3.4 P1 Protocol Elements 11
7.5.3.4.1 P1 Envelope Protocol Elements 11
7.5.3.5 ORName Protocol Elements 16
7.5.3.6 P2 Protocol Profile (Based on [X.420]) . . . 18
7.5.3.6.1 P2 Protocol - Heading 19
7.5.3.6.2 P2 Protocol - BodyParts 21
7.5.3.6.3 P2 BodyPart Protocol Elements 23
7.5.4 Reliable Transfer Server (RTS) 25
7.5.4.1 Implementation Strategy 25
7.5.4.2 RTS option selection 25
7.5.4.3 RTS Protocol Options and Clarifications . . . 26
7.5.4.4 RTS Protocol Limitations 29
7.5.5 Use of Session Services 31
7.5.6 Data Transfer Syntax 31
7.6 PRMD to ADMD and ADMD to ADMD 31
7.6.1 Introduction 31
7.6.2 Additional ADMD Functionality 33
7.6.2.1 Relay Responsibilities of an ADMD 33
7.6.2.2 P1 Protocol Classification Changes 34
7.6.2.3 O/R Names 34
7.6.2.4 P1 ADMD Name 35
7.6.3 Interworking with Integrated UAs 35
7.6.4 Differences with Other Profiles 35
7.6.4.1 TTC Profile 35
7.6.4.2 CEPT Profile 36
7.6.5 Connection of PRMDs to Multiple ADMDs 36
7.6.6 Connection of an ADMD to a Routing PRMD 36
7.6.7 Management Domain Names 37
7.6.8 Envelope Validation Errors 37
7.6.9 Quality of Service 38
7.6.9.1 Domain Availability 38
7.6.9.1.1 ADMD Availability 38
7.6.9.1.2 PRMD Availability 38
7.6.9.2 Delivery Times 38
7.6.10 Billing Information 39
7.6.11 Transparency 39
7.6.12 RTS Password Management 40
7.6.13 For Further Study 40
7.7 INTER and INTRA PRMD CONNECTIONS 40
7.7.1 Introduction 40
7.7.2 The Relaying PRMD 41
7.7.2.1 Relay Responsibilities of a PRMD 41
7.7.2.2 Interaction with an ADMD 41
7.7.3 Intra PRMD Connections 42
7.7.3.1 Relay Responsibilities of an MTA 42
7.7.3.2 Loop Suppression within a PRMD 43
7.7.3.3 Routing Within a PRMD 44
7.7.3.3.1 Class Designations 44
7.7.3.3.2 Specification of MTA Classes 46
7.7.3.3.3 Consequences of Using Certain Classes of

MTAs 46
7.7.3.4 Uniqueness of MPDUidentifiers Within a PRMD . 47
7.7.4 Service Elements and Optional User Facilities . . 47
7.7.5 X.400 Protocol Definitions 48
7.7.5.1 Protocol Classification 48
7.7.5.2 P1 Protocol Elements 48
7.7.5.3 Reliable Transfer Server (RTS) 51
7.8 ERROR HANDLING 51
7.8.1 MPDU Encoding 52
7.8.2 Contents 52
7.8.3 Envelope 52
7.8.3.1 Pragmatic Constraint Violations 52
7.8.3.2 Protocol Violations 52
7.8.3.3 O/R Names 52
7.8.3.4 TraceInformation 53
7.8.3.5 InternalTraceInfo 54
7.8.3.6 Unsupported X.400 Protocol Elements 54
7.8.3.6.1 deferredDelivery 54
7.8.3.6.2 PerDomainBilateralInfo 55
7.8.3.6.3 ExplicitConversion 55
7.8.3.6.4 alternateRecipientAllowed 55
7.8.3.6.5 contentReturnRequest 55
7.8.3.7 Unexpected Values for INTEGER Protocol Elements 55
7.8.3.7.1 Priority 55
7.8.3.7.2 ExplicitConversion 56
7.8.3.7.3 ContentType 56
7.8.3.8 Additional Elements 56
7.8.4 Reports 56
7.9 MHS USE OF DIRECTORY SERVICES 56
7.9.1 Directory Service Elements 56
7.9.2 Use of Names and Addresses 57
7.10 CONFORMANCE 58
7.10.1 Introduction 58
7.10.2 Definition of Conformance 58
7.10.3 Conformance Requirements 60
7.10.3.1 Introduction 60
7.10.3.2 Initial Conformance 60
7.10.3.2.1 Interworking 61
7.10.3.2.2 Service 61
7.11 APPENDIX A: INTERPRETATION OF X.400 SERVICE ELEMENTS . . 62
7.12 APPENDIX B: RECOMMENDED X.400 PRACTICES 66
7.12.1 RECOMMENDED PRACTICES IN P2 66
7.12.2 RECOMMENDED PRACTICES IN RTS 66
7.12.3 RECOMMENDED PRACTICES FOR ORName 67
7.12.4 POSTAL ADDRESSING 70
7.12.5 EDI use of X.400 71
7.12.5.1 Introduction and Scope 71
7.12.5.2 Model 71
7.12.5.3 Protocol Elements Supported for EDI 73
7.12.5.4 Addressing and Routing 73
7.12.6 USA Body Parts 74
7.13 APPENDIX C: RENDITION OF IA5Text AND T61String CHARACTERS 75

7.13.1 GENERATING AND IMAGING IA5Text 75
7.13.2 GENERATING AND IMAGING T61String 75
7.14 APPENDIX D: DIFFERENCES IN INTERPRETATION DISCOVERED THROUGH TESTING OF THE MHS FOR THE CeBit 87 DEMONSTRATION 76
7.14.1 ENCODING OF RTS USER DATA 76
7.14.2 EXTRA SESSION FUNCTIONAL UNITS 76
7.14.3 MIXED CASE IN THE MTA NAME 77
7.14.4 X.410 ACTIVITY IDENTIFIER 77
7.14.5 ENCODING OF PER RECIPIENT FLAG AND PER MESSAGE FLAG 77
7.14.6 ENCODING OF EMPTY BITSTRINGS 78
7.14.7 ADDITIONAL OCTETS FOR BITSTRINGS 78
7.14.8 APPLICATION PROTOCOL IDENTIFIER 78
7.14.9 INITIAL SERIAL NUMBER IN S-CONNECT 78
7.14.10 CONNECTION DATA ON RTS RECOVERY 78
7.14.11 ACTIVITY RESUME 78
7.14.12 OLD ACTIVITY IDENTIFIER 79
7.14.13 NEGOTIATION DOWN TO TRANSPORT CLASS 0 79
7.15 APPENDIX E: WORLDWIDE X.400 CONFORMANCE PROFILE MATRIX . 80
7.16 APPENDIX F: INTERWORKING WARNINGS 91

8. DIRECTORY SERVICES PROTOCOLS 1
8.1 INTRODUCTION 1
8.2 SCOPE AND FIELD OF APPLICATION 1
8.3 STATUS AND REFERENCES 3
8.4 Use of Directories 3
8.5 Directory ASEs, Application Contexts, and Ports 4
8.6 Schemas 5
8.6.1 Maintenance of Structures and Naming Rules 5
8.6.2 Maintenance of object classes and subclasses . . . 6
8.6.3 Maintenance of Attribute Types 6
8.6.4 Maintenance of Attribute Syntaxes 6
8.7 Classification of Support for Attribute Types 6
8.7.1 Mandatory Support 7
8.7.2 Optional Support 7
8.8 Introduction to Pragmatic Constraints 7
8.9 General Constraints 8
8.9.1 Character Sets 8
8.9.2 APDU Size Considerations 8
8.9.3 Service Control (SC) Considerations 8
8.9.4 Priority Service Control 9
8.10 Constraints on Operations 9
8.10.1 Filters 10
8.10.2 Errors 10
8.11 Constraints on Attribute Types 10
8.11.1 Attribute Values 10
8.12 Conformance 14
8.12.1 DUA Conformance 14
8.12.2 DSA Conformance 15
8.12.3 Directory Systems Conformance Classes 15
8.12.4 Authentication Conformance 16
8.12.5 Authentication Conformance Classes 17

8.13 Distributed Operations 17
8.13.1 Referrals and Chaining 17
8.14 Underlying Services 17
8.14.1 ROSE 17
8.14.2 Session 18
8.15 Access Control 18
8.16 Authentication 18
8.17 Test Considerations 18
8.17.1 Major elements of Architecture 18
8.17.2 Search Operation 19
8.18 Errors 20
8.18.1 Permanent vs. Temporary Service Errors 20
8.19 APPENDIX A Definitions 21
8.20 APPENDIX B Attributes and Object Classes 22
8.21 APPENDIX C The Use of ROSE 27
8.22 APPENDIX D Guidelines 28
8.23 APPENDIX E Glossary 29

9. SECURITY 1
9.1 Definitions 1
9.2 Matrix of Security Services and OSI Layers 2

10. OBJECT IDENTIFIER: STRUCTURE AND ALLOCATION 1
10.1 Specific ASE Requirements for ACSE Presentation and Session 5
10.1.1 FTAM 5
10.1.1.1 Phase 2 5
10.1.2 MHS 8
10.1.2.1 Phase 1 8
10.1.3 DS 9
10.1.3.1 Phase 1 9

11. REFERENCES 1
11.1 ISO 1
11.2 CCITT 6
11.3 IEEE 8
11.4 NBS 9
11.5 MAP 9
11.6 TOP 9
11.7 CEN/CENELEC 10
11.8 SPAG 10

List of Figures

Figure 2.1 LSAP bit pattern 1
Figure 2.2 I-Field Format 5

Figure 4.1 AK exchange on idleconnection 5

Figure 6.1 Model of file transfer/access 2

Figure 7.1 The layered structure of this implementation agreement . . 1
Figure 7.2 This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD; and (D) MTA and MTA 3
Figure 7.3 Interconnection of private domains 4
Figure 7.4 X.409 Definition of Privately Defined BodyParts 22
Figure 7.5 An ADMD may (b) or may not (a) serve as a relay 33
Figure 7.6 Relaying PRMD 41
Figure 7.7 Intra PRMD connections 42
Figure 7.8 MD C must know of A to route the message 42
Figure 7.9 Definition of InternalTraceInfo 43
Figure 7.10 Defined Actions in MTASuppliedInfo 44
Figure 7.11 Example of a confirguration to be avoided 47

Figure 8.1 Structure of this Implementation Agreement 1
Figure 8.2 Stand-alone Directory Model 2
Figure 8.3 Distributed Directory Model 3
Figure 8.4 Access to the Directory 5
Figure 8.5 APDU Exchange 8
Figure 8.6 Logical DSA Application Environment 9

List of Tables

Table 5.1 Session States 14
Table 5.2 Incoming Events 15

Table 6.1 Parameters for FTAM-1, -2, -3 7
Table 6.2 Parameters for NBS-6, NBS-7, NBS-8 8
Table 6.3 FTAM primitive data types 9
Table 6.4 IRV Graphic Character Allocations 11
Table 6.5 Interoperable configurations 19
Table 6.6 Required minimal parameter support 25
Table 6.7 Implementation profile support requirements 32
Table 6.8 Implementation Profiles (NBS) and Profiles (SPAG/CEN-CLC) 36
Table 6.9 Information objects in NBS-6 39
Table 6.10 Information objects in NBS-7 44
Table 6.11 Information objects in NBS-8 48
Table 6.12 Datatypes for keys 50
Table 6.12 Information objects in NBS-9 54
Table 6.14 Basic constraints for NBS Ordered flat 58
Table 6.15 Identity constraints in NBS Ordered flat 59

Table 7.1 Basic MT service elements 6
Table 7.2 MT optional user facilities provided to the UA-selectable on a per-message basis 7
Table 7.3 MT optional user facilities provided to the UA agreed for any contractual period of time 7
Table 7.4 Basic IPM service elements 8
Table 7.5 IPM optional facilities agreed for a contractual period of time 8
Table 7.6 IPM optional user facilities selectable on a per-message basis 9
Table 7.7 Protocol Classifications 10
Table 7.8 P1 protocol elements 12
Table 7.9 ORName protocol elements 17
Table 7.10 P2 heading protocol elements 19
Table 7.11 P2 BodyParts 23
Table 7.12 Checkpoint window size of IP 29
Table 7.13 RTS protocol elements 30
Table 7.14 P1 Protocol Classification Changes for a Delivering ADMD . 34
Table 7.15 Delivery Time Targets 38
Table 7.16 Forced Nondelivery Times 39
Table 7.17 Conformant MTA Classifications 45
Table 7.18 P1 Protocol Elements 49
Table 7B.1 Printable string to ASCII mapping 69
Table 7E.1 Protocol element comparison of RTS 81
Table 7E.2 Protocol element comparison of P1 83
Table 7E.3 Protocol element comparison of P2 88

Table 8.1 Pragmatic Constraints for Selected Attributes. 11

Table 9.1 OSI Layers Desirable for Placing Security 3

Table 10.1 TABLE OF ALLOCATED OBJECT IDENTIFIERS 3

1. GENERAL INFORMATION

1.1 PURPOSE OF THIS DOCUMENT

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

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

2) The agreements have been approved by the NBS Workshop Plenary for progression from the "On-Going" agreements document to the draft stable agreements document and after a period of review have been passed to these final stable agreements. The only changes allowed will be clarifications, errata and the correction of omissions discovered in their implementation.

For these reasons, the agreements are considered advanced enough for use in product and test suite development, as well as for procurement references.

Future releases of these Stable Agreements will add and/or extend functionality offered by this version. When required, new versions will be introduced on a yearly basis. It is the NBS Workshop intent that new versions of this Stable Agreements document will be compatible with the present version. If this proves impractical, the agreements will attempt to provide mechanisms and guidelines which maximize interoperability.

1.2 PURPOSE OF THE WORKSHOP

In February, 1983, at the request of industry, NBS organized the NBS Workshop for Implementors of OSI to bring together future users and potential suppliers of OSI protocols. The workshop accepts as input the specifications of emerging standards for protocols and produces as output agreements on the implementation and testing particulars of these protocols. This process is expected to expedite the development of OSI protocols and promote interoperability of independently manufactured data communications equipment.

1.3 WORKSHOP ORGANIZATION

The Workshop organizes its work through Special Interest Groups (SIGs) that prepare technical documentation. An executive committee of SIG chairpersons led by the overall Workshop chairperson administers the workshop. NBS 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 Bureau of Standards 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:

NBS WORKSHOP FOR IMPLEMENTORS OF OSI
ATTENTION: Larry Keys
Building 225, Room B-217
National Bureau of Standards
Gaithersburg, Maryland 20899

2. SUB NETWORKS

2.1 INTRODUCTION

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

2.2 SCOPE AND FIELD OF APPLICATION

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

2.3 STATUS

This version was completed on December 18, 1987.

Ongoing work for other agreements and for future versions of these agreements is contained in the "On-Going" agreements document, including provisions for ISDN.

2.4 ERRATA

2.5 LOCAL AREA NETWORKS

2.5.1 IEEE 802.2 LOGICAL LINK CONTROL

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

1. Link Service Access Point (LSAP)

   The IEEE 802 committee has assigned the code below to address systems using any ISO network layer protocol. Note that bit zero is transmitted first.

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

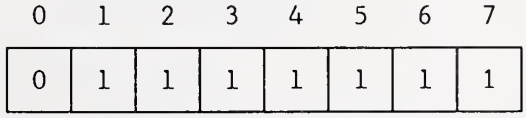


Figure 2.1 LSAP bit pattern

2. Type and Class

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

2.5.2 IEEE 802.3 CSMA/CD ACCESS METHOD

NOTE: **Draft Stable Agreements for this section stated only that, "The 48 bit addressing shall be used." Historically, only 10 BASE 5 technology was considered.**

10 BROAD 36 CSMA/CD agreements are included in these stable agreements for consistency and interoperability with 802.4 broadband systems.

This note flags the following agreements for 10 BROAD 36 CSMA/CD as new material. The 10 BROAD 36 agreements are recommendations only.

The following implementation agreements have been reached with respect to the IEEE 802.3 CSMA/CD Access Method and Physical Layer Specifications for 10 BASE 5:

- o The address length shall be 48 bits

The following is only a recommendation for 10 BROAD 36 Networks:

1. Single Cable Networks

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

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

2. Dual Cable Networks

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

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

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

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

2.5.3 IEEE 802.4 TOKEN BUS ACCESS METHOD

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

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

- Broadband Channel Assignments

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

2.5.4 IEEE 802.5 TOKEN RING ACCESS METHOD

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

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

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

and the:

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

Figure 2.2 I-Field Format

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

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

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

2.6 WIDE AREA NETWORKS

2.6.1 CCITT RECOMMENDATION X.25

The procedures required to describe the DTE/DTE interface and the DTE side of the DTE/DCE interface, for systems attached to subnetworks providing an X.25 interface shall be as defined in ISO 7776 and ISO 8208.

3. NETWORK LAYER

3.1 INTRODUCTION

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

3.2 SCOPE AND FIELD OF APPLICATION

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

3.3 STATUS

This version of the agreements was completed on December 18, 1987.

Ongoing work for other agreements and for future versions of these agreements is contained in the "On-Going" agreements document, including provisions for ISDN, and including specific aspects of using the X.25 PLP over LANs.

3.4 ERRATA

3.5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS)

3.5.1 Provision of CLNS over Local Area Networks (LANS)

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

1. The definition of CLNS shall be as specified in ISO 8348/AD1.

2. The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 3.5.3.1.

3.5.2 Provision of CLNS over X.25 Subnetworks

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

1. The definition of CLNS shall be as specified in ISO 8348/AD1.

2. The protocol used to provide CLNS shall be ISO 8473 with agreements as specified in 3.5.3.1.

3. The necessary subnetwork dependant convergence function shall be as defined in ISO 8473 for operation over X.25 subnetworks, with agreements as specified in 3.5.3.2.

4. The X.25 PLP shall be as defined in ISO 8208.

3.5.3 Agreements on Protocols

3.5.3.1 ISO 8473

1. Subsets of the protocol:

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

   - The non-segmenting subset will not be used. Implementations will not generate data PDUs without a segmentation part. However, implementations will receive and correctly process PDUs which do not contain the segmentation part.

2. Mandatory Functions:

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

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

3. Optional Functions:

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

   - Partial and complete source Routing will not be

supported.[1]

- o Partial record of Route will be supported by Intermediate systems.

- o ISO 8473 will be followed with respect to QOS.

3.5.3.2 Subnetwork Dependent Convergence Function

1. When 8473 SNDC functions are used over X.25 subnetworks, the default throughput class shall be used if this facility is available.

3.6 CONNECTION-MODE NETWORK SERVICE (CONS)

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

3.6.1 Mandatory Method of Providing CONS

When providing CONS using X.25 1984, the following shall apply:

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

- o The mapping of the elements of the CONS to the elements of the X.25 Packet Level Protocol (PLP) is as specified in ISO 8878, Use of X.25 to Provide the Connection-mode Network Service.

- o The general procedures and formats of the X.25 PLP are as specified in ISO 8208, X.25 Packet Level Protocol for Data Terminal Equipment.

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

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

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

---

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

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

3.6.3 Agreements on Protocols

3.6.3.1 ISO 8878

- 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.

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

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

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

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

3.7 ADDRESSING

Address formats supported will conform to ISO 8348 DAD2.

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

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

3.8 ROUTING

3.8.1 Static Routing

End systems and intermediate systems supporting static routing will provide a local mechanism to update and, if necessary, to create the local routing table. Updating and consistency checking will be performed by human operators. The algorithms and data structures used for static routing are not specified in these agreements. Implementors are free to perform these functions in the manner which is most appropriate to their system environment.

3.8.2 End System to Intermediate System

Dynamic routing between end systems and intermediate systems is performed using the protocol described in ISO 9542, End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473. The following agreements apply to the use of this protocol over LANs and point-to-point links.

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

2. Implementations must support both Configuration Information and Route Redirection Information. No subsets are permitted.

3. All timer values must be settable using local system management.

4. Use of checksums must be settable using local system management. Under normal use, checksums will be disabled.

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

6. Both ES and IS implementations must support the 'optimization' described in Clause A.3 of ISO 9542 for system initialization. Its use must be selectable using local system management.

7. This protocol employs the same LSAP as ISO 8473.

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

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

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

10. The ES-IS protocol should be primarily responsible for updating the Network Layer Routing Information Base (RIB). In addition, a management mechanism capable of adding and deleting "static" entries into the 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.

Note: This function enables route table entries to be made which are static in nature. The use of static entries is not intended for normal operations, but rather is provided for test and debug as well as for interoperation with other OSI implementations which do not support the ES-IS protocol.

3.9 MIGRATION CONSIDERATIONS

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

3.9.1 X.25-1980

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

This is an exception to the Agreements for providing the OSI Network Service, granted temporarily for practical reasons. This exception will be removed when it is deemed to be no longer

necessary, in the judgement of the Workshop. While this provision is in effect, it provides an alternative method of using 1980 X.25 to the provisions of 3.6.2

3.10 CONFORMANCE

4. TRANSPORT

4.1 INTRODUCTION

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

4.2 SCOPE AND FIELD OF APPLICATION

Two connection-oriented transport classed have been identified for implementation: classes 0 and 4. Transport class 4 is endorsed for use over CLNS and CONS. Transport class 0 is endorsed for use over CONS.

4.3 STATUS

Completed March 1987.

4.4 ERRATA

4.5 TRANSPORT CLASS 4

4.5.1 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 (i.e., a private MHS, etc.).

4.5.2 Protocol Agreements

The full protocol will be available including expedited data and negotiation at connection establishment. A disconnect request shall be issued in response to a connect request when the maximum number of transport connections is reached or exceeded.

4.5.2.1 Rules for Negotiation

- In general, the ISO rules for negotiation will be used, specifics follow.
- All implementations will send the 16/31 window size/sequence space in the CR TPDU. Implementations must all provide the 16/31 ISO option. Implementations must be able to accept the 4/7 in a CR TPDU.
- The ISO maximum TPDU size is negotiable between 128 and 8K octets, always negotiated downward. The ISO rules are to be followed, allowing any valid size in the CR TPDU. TPDU size negotiation is a local implementation

issue. Each vendor will decide how it is implemented in their end system.

- The security parameter is optional and user defined in the ISO specification. Implementations should not send the security parameter in the CR TPDU; if received the parameter should be ignored.

- Both transports must agree to not use checksum, according to the ISO specifications. Requesting its use is an implementation choice. All implementations must be able to operate with checksum if requested.

- Use of acknowledgement time parameter is optional in ISO 8073. 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 acknowledgement time parameter. The value transmitted should be expressed in units of milliseconds and rounded up to the nearest whole millisecond.

- Throughput, priority, and transit delay are optional in the ISO specification. Do not send in the CR TPDU; ignore in the CC TPDU.

- User data in the CR TPDU and the CC TPDU are optional. No implementation should send; all implementations must be prepared to receive.

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

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

Parameter	Action
TSAP id	Send DR TPDU
TPDU size	ignore parm, use default
Version	ignore parm, usedefault
Protection (Security)	implementation dependent
Checksum	discard CR TPDU
Additional Options	Protocol Error
Alternate Protocol Classes	Protocol Error
Acknowledge Time	ignore parm
Throughput	ignore parm
Residual Error Rate	ignore parm
Priority	ignore parm
Transit Delay	ignore parm

4.5.2.2 TRANSPORT CLASS 4 SERVICE ACCESS POINTS OR SELECTORS

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

4.5.2.3 Retransmission Timer

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

$$t \longleftarrow kE + w$$

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

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

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

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

In this formula, S is the new sample and α is a parameter which can be set to some value between 0 and 1. The value chosen for α determines the relative weighting placed upon the current estimate and the new sample. A large value of α

weights the old estimate more heavily causing it to respond only slowly to variations in the round-trip delay.

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

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

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

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

4.5.2.4 Keep-Alive Function

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

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

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

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

timer. See figure 4.1.

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

- o The window timer must be greater than the round-trip delay. See section 4.1.4.

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

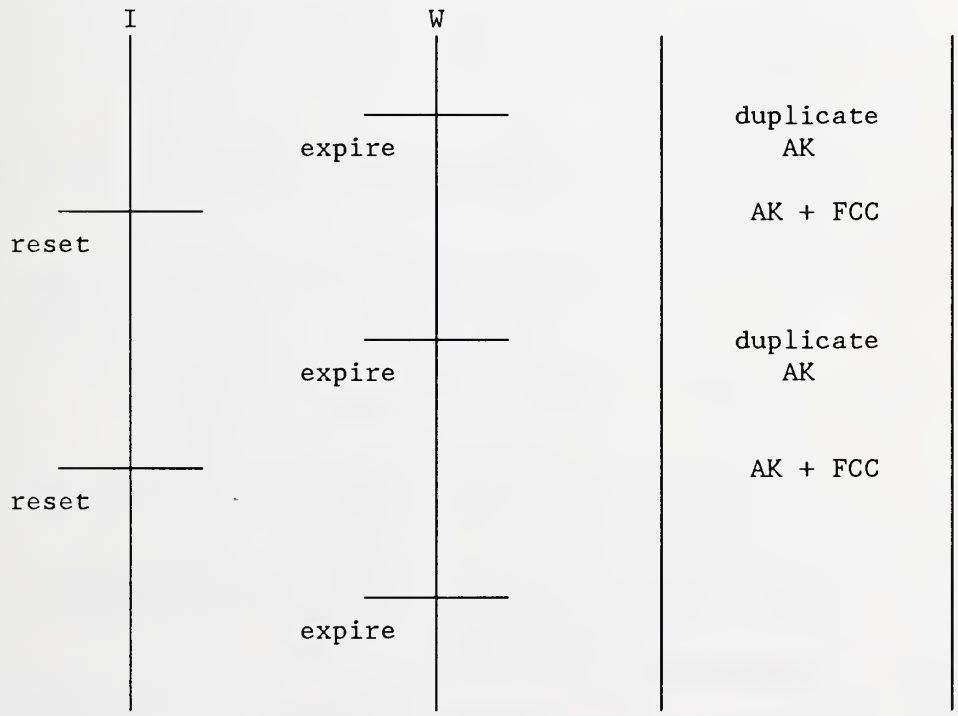


Figure 4.1 AK exchange on idleconnection

4.6 TRANSPORT CLASS 0

4.6.1 Transport Class 0 Overview

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

4.6.2 Protocol Agreements

Transport Class 0 agreements follow:

- The Error (ER) TPDU may be used at any time and upon receipt requires that the recipient disconnect the network connection, and by extension the transport connection.
- The allowed values for the maximum TPDU size are as specified in ISO 8073. They are: 128, 256, 512, 1024, and 2048.
- The class 0 protocol does not support multiplexing. At any instant, one transport corresponds to one network connection.
- It is recommended that the optional timers TS1 and TS2, if implemented, be settable by local system management. Values in the order of minutes should be supported.
- An unlimited TSDU length must be supported.

4.6.2.1 TRANSPORT CLASS 0 SERVICE ACCESS POINTS

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

4.6.3 Rules for Negotiation

The ISO rules for negotiations will be used.

5. UPPER LAYERS

5.1 INTRODUCTION

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

5.1.1 References

All documents referenced in the Upper Layers section of these agreements can be found in the REFERENCES section of this NBS Implementors' Agreements document.

5.2 SCOPE AND FIELD OF APPLICATION

This section does not detail particular conformance statements for ACSE, Presentation, and Session, since what is to be implemented in each case depends on which Application Service Elements (ASE's) and which functional units within each ASE are used with an Application Process. Each ASE's SIG must specify which functional units of each layer it requires. However, the scope of each layer is based on the total indicated requirements of all ASE's for which there is an active NBS SIG. The implementation agreements are not specified beyond that scope.

It is not the intent of this document to specify or reproduce standards, but when a referenced standard is unclear or has known defects, an attempt will be made to remedy the problem herein. Any attempted clarification should be considered as a possible interpretation; the ISO standard still takes precedence if there is any conflict. The situation with respect to defects in a standard is somewhat different; a reported defect may be technically resolved by the appropriate international technical committee with likely approval by the voting members pending for several months. Since relevant defects can't be ignored in an implementation, this document will recommend using defect resolutions which have the tentative approval of the appropriate standards committees.

5.3 STATUS

This document is the first stable version of the NBS UL SIG.

5.4 ERRATA

5.4.1 ISO Defect Reports

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

5.4.2 Session Defects

These defects are listed for the benefit of X.400 implementations.

The following 8326 defect reports have been incorporated into version 1 of Session:

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

The following 8327 defect reports have been incorporated into version 1 of Session:

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

5.5 ASSOCIATION CONTROL SERVICE ELEMENT

5.5.1 Introduction

This section details the implementation requirements for the Association Control Service Element (ACSE) of the Application layer. It is the intent of this section to follow the ISO ACSE standards. Where those specifications are inadequate, this section should provide the necessary information.

5.5.2 Services

5.5.2.1 ACSE Services

The following ACSE service primitives are within the possible scope of an NBS conformant system:

1. A_ASSOCIATE request
2. A_ASSOCIATE indication
3. A_ASSOCIATE response
4. A_ASSOCIATE confirm
5. A_RELEASE request
6. A_RELEASE indication
7. A_RELEASE response
8. A_RELEASE confirm
9. A_ABORT request
10. A_ABORT indication
11. A_P_ABORT indication

5.5.2.2 Use of Presentation Layer Services

ACSE services will make use of Presentation layer services in the manner defined in the ACSE Protocol specification.

5.5.3 Protocol agreements

Implementations shall be based on the ACSE Service definition and the ACSE Protocol specification.

5.5.3.1 Application Context

Specific Application Contexts and their names will be supplied and defined by an appropriate NBS SIG. Other application contexts may be defined and specified as dictated by particular application requirements.

Optional names and specifications are outlined by each application SIG under the heading "Specific ASE Requirements for ACSE, Presentation, and Session". The use of these names implies adherence to the relevant NBS implementors' agreements for a particular application SIG.

The utility of an NBS defined name (which is an OBJECT IDENTIFIER) is left up to the application. An NBS name may or may not be used in the ACSE APDU. The consequence of the name is left up to the application entities and any a priori agreements that they have. In other words, it is up to the application whether this parameter is ignored or validated for correctness. (Note that the consequence of this name must also be dictated by the particular conformance test).

The UL SIG recognizes that this parameter needs further definition by the appropriate standards bodies. Therefore, the use of this parameter for association negotiation is not recommended at this time.

5.5.3.2 Implementation Information

An implementation shall be able to receive at least 128 octets for NBS conformance.

5.5.3.3 AE Title

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. The AE Title maps onto the AP Title and the AE Qualifier.

The AE Qualifier is implied to be nil and is not sent. The value for the AP Title is { 1 3 9999 1 ftam-nil-ap-title(7) } at this time.

5.6 PRESENTATION

5.6.1 Introduction

This section details the implementation requirements for the Presentation layer. It is the intent of this section to follow the ISO Presentation Standards. Where those specifications are inadequate, this section should provide the necessary information.

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

5.6.2 Services

5.6.2.1 Presentation Services

The following functional units are within the possible scope of an NBS conformant system:

> Presentation Kernel - This functional unit supports the basic Presentation services required to establish a Presentation connection, transfer normal data, and release a Presentation connection. This is a non-negotiable functional unit.
>
> The Context Management and Context Restoration functional units are not within the scope of an NBS conformant system and need not be supported.

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. Any service not supported by the Session layer is also not supported by the Presentation layer; see the section on Session Functional Units for the possible Session functional units. The services provided by the Presentation layer are limited by the services provided by the Session layer as defined in the Session service definition ISO/IS 8326 and the Session protocol definition ISO/IS 8327.

5.6.2.2 Use of Session Layer Services

Presentation layer services shall make use of Session layer services in the manner defined in the Presentation Protocol Specification.

5.6.3 Protocol Agreements

Implementations shall be based on the Presentation Service Definition, ISO 8822 and the Presentation Protocol Definition, ISO 8823 .

5.6.3.1 Transfer Syntaxes

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

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

5.6.3.2 Abstract Syntaxes

- Several abstract syntax names may map onto a single transfer syntax name.

- The ACSE abstract syntax shall always be present in the defined context set.

5.6.3.3 Presentation Context Identifier

- The presentation context identifier value shall be encoded in no more than 2 octets.

- Implementations must be able to handle a minimum of 2 presentation contexts per connection.

5.6.3.4 Mode-selector Position in SET

Whenever the Mode-selector value within either a CP-PPDU or CPA-PPDU is normal-mode (1), it shall occur as the first element within the SET.

5.6.3.5 Default Context

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

5.6.3.6 P-Selectors

Local P-selectors shall be a maximum of 4 octets. This applies only to P-selectors in PPDUs whose receipt by an NBS-conformant system normally results in either a P-CONNECT indication or a P-CONNECT confirmation being issued.

5.6.3.7 Provider Abort Parameters

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

5.6.3.8 Provider Aborts and Session Version

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

5.6.3.9 CPC-type

NBS conformant implementations shall not use any CPC-type values in the SS-user data parameter of the S-CONNECT request unless more than one transfer syntax is proposed. Each CPC-type represents a unique transfer syntax. Note: currently only 1 transfer syntax is recognized.

5.6.3.10 Presentation-context-definition-result-list

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

5.6.3.11 RS-PPDU

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

5.6.4 Presentation ASN.1 Encoding Rules

5.6.4.1 Invalid Encoding

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

5.6.4.2 Protocol-version, Presentation-requirements

Protocol-version and Presentation-requirements shall be encoded in their primitive forms, if present.

5.6.4.3 Presentation-selector

Any type defined as a Presentation-selector shall be encoded in its primitive form, if present.

5.7 SESSION

5.7.1 Introduction

This section details the implementation requirements for the Session layer. It is the intent of this section to follow the ISO Session Standards to the fullest extent possible. Where those specifications are inadequate, this section should provide the necessary information.

5.7.2 Services

5.7.2.1 Session Services

The following functional units are within the possible scope of an NBS conformant system:

Kernel

Duplex

Expedited Data

Resynchronize

Exceptions

Activity Management

Half-duplex

Minor Synchronize

Major Synchronize

Typed Data

5.7.2.2 Use of Transport Services

The use of Transport layer services by the Session layer functional units listed in the previous section is as specified in the Transport Protocol Specification, ISO/IS 8073.

5.7.3 Protocol Agreements

Implementations shall be based on the Session service definition ISO/IS 8326 and the Session protocol definition ISO/IS 8327.

5.7.3.1 Concatenation

When a category 0 SPDU is concatenated with a category 2 SPDU, the category 0 SPDU shall contain neither the Token Item field nor User Data. If either a Token Item field or User Data is received in such a concatenated incoming SPDU, the receiving Session Protocol Machine has the option of either properly processing the fields or issuing a provider abort on the connection.

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

5.7.3.2 Segmenting

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

5.7.3.3 Reuse of Transport Connection

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

5.7.3.4 Use of Transport Expedited Data

The use of transport expedited service is as stated in the session protocol specification: if available, transport expedited service must be used.

5.7.3.5 Use of Session Version Number

Session versions 1 and 2 are recognized. Each relevant NBS SIG chooses the version or versions of Session which it requires for a particular implementation phase, and these choices are documented in section 5.9.1.

Session version 2 specifies the use of unlimited user data during connection establishment as dictated by the DAD 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. NBS conformant Session version 2 implementations need only support the maximum data lengths specified in the Specific ASE Requirements section.

5.7.3.6 Receipt of Invalid SPDUs

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

5.7.3.7 Invalid SPM Intersections

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

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

5.7.3.8 S-Selectors

S-selectors shall be a maximum of 16 octets.

5.8 Universal ASN.1 Encoding Rules

5.8.1 Tags

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

5.8.2 Definite length

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

5.8.3 EXTERNAL Type

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.

5.9 CONFORMANCE

In order for an implementation to be in conformance with the NBS implementors' agreements, the following rules shall be adhered to:

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

- Guidelines for implementation of standards' defects will be as per the resolution of such defects by the appropriate ISO standards committee.

5.10 APPENDIX A: RECOMMENDED PRACTICES

Reflect Parameter Values

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
AB-r	44	ABORT - reuse SPDU

Table 2 - continued

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

6. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 2

6.1 INTRODUCTION

This section defines Implementors' Agreements based on ISO File Transfer, Access and Management (FTAM), as defined in ISO 8571. This International Standard has four parts. Part 1 of the IS gives general concepts, Part 2 defines the Virtual Filestore (VFS), Part 3 defines the File Service, and Part 4 defines the File Protocol.

FTAM, as described in the IS, is based on the following ISO documents: ACSE Service and Protocol (ISO 8649, ISO 8650), Presentation Service and Protocol (ISO 8822, ISO 8823), ASN.1 Abstract Syntax Notation and Basic Encoding Rules (ISO 8824, ISO 8825), and Session Service and Protocol (ISO 8326, ISO 8327). These services and protocols are defined architecturally in the OSI Reference Model (ISO 7498). These Agreements provide detailed guidance for the implementor, and eliminate ambiguities in interpretations.

The general agreements reached with respect to the ISO File Transfer, Access and Management Protocol (FTAM) are that the Phase 2 FTAM specification (this section) is based on the International Standard (IS).

These Agreements will serve as a basis for product development. The purpose is to enable systems implemented according to these Agreements to successfully interoperate.

6.2 SCOPE AND FIELD OF APPLICATION

These FTAM Phase 2 Agreements cover transfer of and access to files between the Filestores of two end systems, including the management of a Virtual Filestore. One end system acts in the Initiator role and initiates the file transfer/access, while the other end system acts in the Responder role and provides access to the file in the Virtual Filestore. This paper describes Agreements for the actions and attributes of the Virtual Filestore, and the service provided by the file service provider to file service users, together with the necessary communications between the Initiator and Responder.

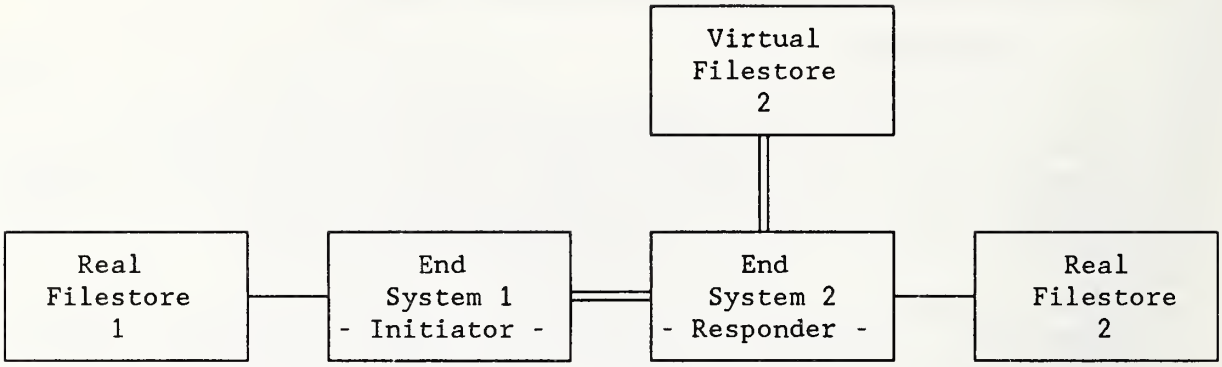


Figure 6.1 Model of file transfer/access

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 FTAM functions for a minimal functionality (Conformant Implementations) and for several Implementation Profiles which are tailored to different classes of user requirements to FTAM.

6.3 STATUS

This version of the FTAM Implementation Agreements was completed December 18, 1987. No further enhancements will be made to this version. See the next section, ERRATA.

Note: These Agreements were updated from the previous March 1987 DIS based Agreements.

6.4 ERRATA

6.5 ASSUMPTIONS

1. FTAM protocol machines must be able to parse and process at a minimum 7K octets of FTAM PCI and FTAM user data (including grouped FPDUs) as they would be encoded with the ASN.1 Basic Encoding Rules. It is recommended, however, that Presentation user data not be restricted in size.

2. In order to maximize interoperability, it is important that the implementations of FTAM service providers do not unnecessarily restrict the service user's ability to generate arbitrary file service requests. Otherwise, they may not be able to work with FTAM Responders whose operation is constrained by their mapping

of the FTAM Virtual Filestore to their local filestore. For example, error procedures should only be invoked when an error actually occurs, not at the point of the specification of options which might result in an error.

3. Implementations must be able to parse all valid optional parameters if they are present in the PDU. Only those optional parameters specified as supported in these Agreements are required to be implemented. If these parameters are not present, a default value is assigned locally. A Responder should not refuse a request solely because a parameter that is optional in the FTAM standard, but is supported in these Agreements, is not present.

4. Consideration of any standardized service interface is not covered by these Agreements.

5. These Agreements define no restrictions for the values used for the <communication quality of service> parameter in <F-INITIALIZE>.

6. FTAM is defined in phases. The Phase 1 FTAM implementation specification is based on the second ISO Draft Proposal, dated April 1985, and the ISO Draft Proposal 8824 and 8825.

   The Phase 2 FTAM specification (this section) is based on the International Standard (IS). THERE IS NO BACKWARD COMPATIBILITY WITH NBS 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 NBS Implementors Agreements or with respect to FTAM profiles of other bodies (see section 6.13 of this document). It is the goal of these Agreements to use the 'user version' mechanism to provide at least one level of backward compatibility for all future NBS FTAM Phases, facilitating backward compatibility for future FTAM products, assuming different new versions of the respective IS's also enable backward compatibility.

6.6 PRESENTATION AGREEMENTS

The following Abstract Syntaxes are recognized in these agreements:

"FTAM FADU"
"FTAM PCI"
"FTAM unstructured text abstract syntax"
"FTAM unstructured binary abstract syntax"
"NBS abstract syntax AS1"
"NBS file directory entry abstract syntax"

The following Transfer Syntax is supported:

"Basic Encoding of a single ASN.1 type"
(See Appendix A, Part 3)

6.7 SERVICE CLASS AGREEMENTS

Implementation Agreements have been reached for the following service classes.

- File Transfer
- File Access
- File Management
- File Transfer and Management
- Unconstrained

6.8 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.

6.9 FILE ATTRIBUTE AGREEMENTS

Implementation of the Kernel Group of file attributes is defined. If the optional Storage Group and Security Group are implemented, aspects of their implementation are defined. Implementation of the Private

Group is not specified.

Responses to an attribute value request shall always include one of the following (as specified in ISO 8571-2, clause 9.4):

- An actual file attribute value.
- A value indicating that no value is available, optionally with a diagnostic.
- No value and an error code, optionally with a diagnostic indicating that the attribute is not supported.

6.9.1 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. Requests using such a single component <filename> value with a sequence of one Graphic String are 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.

6.9.2 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.

6.10 DOCUMENT TYPE AGREEMENTS

These document types are defined.

FTAM-1	"ISO FTAM unstructured text"
FTAM-2	"ISO FTAM sequential text"
FTAM-3	"ISO FTAM unstructured binary"
NBS-6	"NBS-6 FTAM sequential file"
NBS-7	"NBS-7 FTAM random access file"
NBS-8	"NBS-8 FTAM indexed file"
NBS-9	"NBS-9 FTAM file directory file"

Detailed document type definitions are given in Appendix 6A 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 6.1.

Table 6.1 Parameters for FTAM-1, -2, -3

	Universal Class Number	Maximum String Length[6]	String Significance
FTAM-1	General String [1](27) IA5String [2](22)	134 or less	'not-significant'
FTAM-2	Graphic String [3,4](25)	134 or less[5]	'not-significant'
FTAM-3	<not applicable>	512 or less	'not-significant[4]

Notes:

1. The minimum level of support for General String is the IA5 G0 character set and the 8859-1 G0 and G1 character sets, and IA5 C0 set.

2. The support for IA5 String is the IA5 G0 character set and the IA5 C0 set.

3. The minimum level of support for Graphic String is the IA5 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 6.3.

For the use of FTAM-2 only the FADU identities of 'begin', 'end', 'first', and 'next' are required for conformant implementations.

For the document types NBS-6, NBS-7 and NBS-8 parameters are used for which the Agreements apply as specified in Table 6.2.

Table 6.2 Parameters for NBS-6, NBS-7, NBS-8

Parameter	PrimType	String-length	Length-1	Length-2
int	INTEGER	Number of octets required to represent, in 2's complement format, the largest integer to be passed		
bit	BIT STRING	Number of bits in string (non-varying)		
ia5	IA5String	Max number of characters in string		
graphic	Graphic String	Max number of characters in string		
general	General String	Max number of characters in string		
octet	OCTET STRING	Max numbers of octets in string		
private-class-number	Floating Point Number		The minimum number of bits required to be maintained in the mantissa for relative precision	Number of bits required to represent the largest unbiase integer exponen 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 characters 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 6.3.

Table 6.3 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 6A Part 3. It is derived from existing standards IEC 559 and IEEE 754.

6.10.1 Character Sets

Implementation of a character set in FTAM is understood as:

- a transfer syntax is defined for the character set
- document types are defined using the character set in their abstract syntactic definition
- 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 IA5 and ISO 8859-1 shall always be implemented.

6.10.1.1 IA5 Character Set

The International Reference Version (IRV) of IA5 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 6.4.

Table 6.4. IRV Graphic Character Allocations

Graphic	Name	Coded Representation
#	Number sign	2/3
*	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)

6.10.1.2 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 IA5 definition.

6.10.2 Document Type Negotiation Rules

6.10.2.1 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.

6.10.2.2 File Creation

An <F-CREATE request> FPDU must contain a <document type name> value in its <initial attributes> parameter. 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.

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.

6.10.2.3 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.

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.

6.10.3 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.

6.11 F-CANCEL ACTION

When an F-CANCEL is sent or received, the following occurs:

- o no more data is sent,
- o checkpoint numbers are removed, and
- o state of the file is implementation dependent.

Note: When mapping F-CANCEL on P-RESYNCHRONIZE (abandon) it is required that P-SYNC-MINOR be used after F-READ/F-WRITE (see ISO 8571-4 clauses 13, 14).

6.12 IMPLEMENTATION INFORMATION AGREEMENTS

- o The <implementation information> parameter of <F-INITIALIZE> FPDU is not required by these Agreements.

- o It may be used to pass user version information as a series of values, separated by ';'.

- o The following will indicate conformance to the NBS Phase 2 Agreements: NBS-Phase2.

  Note: The list of possible values may be enlarged for future FTAM phases or FTAM profiles of other bodies.

- o 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.

6.13 DIAGNOSTIC AGREEMENTS

- o 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'.)

- o General catch-all diagnostic action is discouraged.

- o The <further details> subfield is supported. It will be encoded as GraphicString, but is restricted to IA5 (IRV, graphic characters) and ISO 8859-1 only.

- o 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.

6.14 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.

6.15 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'.

6.16 SECURITY

6.16.1 Optional Password Support

Users may provide values for <initiator identity> and <filestore password>. Password support in FTAM is not required. If this information is provided, it will be sent to the Responder on the <F-INITIALIZE>.

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.

6.16.2 Access Passwords

Users may provide access passwords. If the information is provided, the passwords will be sent to the Responder in the <access passwords> parameter.

6.16.3 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.

6.17 REQUIREMENT FOR CONFORMANT IMPLEMENTATIONS

This section gives the criteria to be satisfied by every implementation of FTAM that conforms to these Agreements.

Conformance to these Agreements is stated in terms of the different roles occupied by FTAM implementations. The interoperability of certain configurations of these roles motivates this approach. Interoperable configurations of these roles are given in section 6.17.1.

The only function provided by every conformant implementation is the transfer of unstructured binary files in their entirety. It must be recognized that such simple transfer, while commonly understood and generally important, will not support all applications of FTAM. Section 6.18 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.

6.17.1 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 sections 6.17.3 to 6.17.8). 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 sections 6.17.3 to 6.17.8).

These role combinations and this interoperability are shown in Table 6.5 below.

Table 6.5 Interoperable configurations

		Initiator		Responder	
		sender	receiver	sender	receiver
Initiator	sender				x
	receiver			x	
Responder	sender		x		
	receiver	x			

6.17.2 Relationship to ISO 8571--The FTAM Standard

Any implementation in conformance to ISO 8571 (as defined in ISO 8571-4, clause 22 (Conformance)), in addition to the implementation of the minimal protocols and roles enumerated in sections 6.17.3 to 6.17.8, 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.

6.17.3 Requirements for Document Type Support

The document type FTAM-3 shall be supported for purposes of transfer and storage. The details regarding support for FTAM-3 in the FTAM dialogue are given in section 6.10.

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 6.10
- the preferred transfer syntax "Basic Encoding of a single ASN.1 type"

6.17.4 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 section 6.17.8.],
- the grouping protocol and the <threshold> parameter with a value of at least 2 for use in the file transfer class,
- at least one of the read or write protocols [Specific conformance for reading and writing is defined in sections 6.17.6 and 6.17.7.],

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:

- o request the grouping and at least one of the read or write functional units,

- o request the file transfer class with the <service class> parameter,

- o request the document type FTAM-3 using the <document type name> form of the <contents type> parameter,

- o request the <FTAM quality of service> parameter with value 0 and accept in all cases the returned value 0, and

- o 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.

6.17.5 Responders

Every implementation of an FTAM Responder shall support:

- o the kernel protocol and its mandatory parameters with minimum ranges [Minimum required ranges are specified in section 6.17.8.],

- o the grouping protocol and the <threshold> parameter with a value of at least 2 for use in the file transfer class,

- o at least one of the read or write protocols [Specific conformance for reading and writing is defined in sections 6.17.6 and 6.17.7],

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:

- o accept requests for the grouping and at least one of the read or write functional units,
- o accept requests for the file transfer class with the <service class> parameter,
- o accept the document type FTAM-3 using the <document type name> form of the <contents type> parameter,
- o accept requests for an <FTAM quality of service> parameter with any value but may respond with the value 0, and
- o accept requests for a <communication quality of service> consistent with the transport definition in these agreements

as part of the filestore initialization procedures in ISO 8571-4 clause 9.1, FTAM regime establishment.

Responders must be able to operate under all circumstances if the above minimum values are requested on an <F-INITIALIZE request> PDU. Responders must not negotiate upward in the sense described in the standard.

Responders must complete each action requested and supported in a manner consistent with its description in ISO 8571-2 clauses 10 (Actions on complete files) and 11 (Actions for file access), and must interpret each supported attribute in a manner consistent with its definition in ISO 8571-2 clause 12 (File attributes).

Under circumstances where actions cannot be carried out either as requested or consistently with ISO 8571-2 clause 10 (Actions on complete files) and 12 (Actions for file access), the Responder must return at least one diagnostic indicating:

- o if the failure was due to either a protocol or Filestore failure, and then:
  - -- precisely which action failed,
  - -- at least one of the parameters that could not be accommodated with the diagnostic type indicating at least the degree of failure, as given by the action and state result parameter, or

- o that the failure was due to unforeseen system shutdown.

Should the supporting services break down, such that FTAM communication is impossible, the FTAM protocol machine shall notify the user with an <F-P-ABORT indication> and <diagnostic> with identifier 1011, as well as inform the user of any known <further details>.

6.17.6 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.

6.17.6.1 Initiator Senders

Every implementation of an FTAM sender which is also an FTAM Initiator shall support:

- o the write functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU operation	replace
FADU identity	first

and support the applicable procedures, defined in ISO 8571-4 clause 13 (Bulk data transfer initiating entity actions).

6.17.6.2 Responder Senders

Every implementation of an FTAM sender which is also an FTAM Responder shall support:

- o the read functional unit and protocol, and
- o for the document type FTAM-3 the following bulk data transfer specification parameters:

FADU identity	first

Access context	UA

and support the applicable procedures, defined in ISO 8571-4 clause 14 (Bulk data transfer responding entity actions).

6.17.7 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.

6.17.7.1 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:

FADU identity	first
Access context	UA

and support the applicable procedures, defined in ISO 8571-4 clause 13 (Bulk data transfer initiating entity actions).

6.17.7.2 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:

FADU operation	replace
FADU identity	first

and support the applicable procedures, defined in ISO 8571-4

clause 14 (Bulk data transfer responding entity actions).

6.17.8 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 6.6. A conforming implementation may support a wider range of values for any parameter.

Table 6.6 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]	'Not required.'
	all others	As specified in sections 6.17.4 and 6.17.5 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

(Continued on next page.)

Table 6.6 Required minimal parameter support, continued

Parameter		Minimum Range
F_OPEN		
	processing mode	'read' for initiator/receivers 'read' for responder/senders 'replace' for initiator/senders 'replace' for responder/receivers
	contents type	'FTAM-3'
F_READ		
	FADU identity	'first'
	access context	'UA'
F_WRITE		
	FADU operation	'read' for initiator/receivers 'read' for responder/senders 'replace' for initiator/senders 'replace' for responder/receivers
	FADU identity	'first'
F_BEGIN_GROUP		
	threshold[3]	For file transfer (a minimal required function) 2.

For any other supported parameters, minimum ranges are taken from the minimum ranges for the attribute corresponding to each as in ISO 8571-2 Table 4.

Notes:

1. The parameters, functional units, and presentation context management are not ordered, so "minimum value" cannot be formally defined. The above values are those required for conformance to these Agreements but no value conformant to ISO 8571 for use in other applications is regarded to be in violation of these Agreements.

2. Other functional units (and service classes) for defined implementations may also be valid provided that they are implemented in accordance with these Agreements, specifically section 6.17.8.

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.

6.18 IMPLEMENTATION PROFILES

This section defines Implementation Profiles for the specific functions of:

- File Transfer
- File Access
- File Management.

Those definitions are expressed in terms of:

- Document Types
- Attributes
- Service Classes (both service elements and their parameters).

This by no means defines all possible Implementation Profiles.

The following Implementation Profiles are defined:

T1: Simple File Transfer
T2: Positional File Transfer
T3: Full File Transfer
A1: Simple File Access
A2: Full File Access
M1: Management.

Implementation Agreements have been reached for the following service classes. Note that any given implementation may support more than one service class.

- File Transfer
- File Access
- File Management
- Unconstrained
- File Transfer and Management

Support of an Implementation Profile requires adherence to:

1. corresponding definition in 8571-3 clause 8 and any related procedures in 8571-4 clause 8-17,

2. requirements given in sections 6.5-6.18 of these Agreements, and

3. requirements for parameter and attribute support as defined in section 6.17.8.

6.18.1 General Requirements for the Defined Implementation Profiles

- Implementations will be able to act either as Initiator or Responder or both.

- Implementations must support diagnostics as described in section 6.13 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.

6.18.2 Use of Lower Layer Services

- Support for the Presentation Context Management functional unit is not required.

- Implementations will support the Session, Presentation, and ACSE requirements as stated in section 5.

Note: Implementation of the Session Resynchronize functional unit is highly recommended, since the F-CANCEL service may be less effective when mapped to S-DATA.

6.18.3 Document Type Requirements for the Defined Implementation Profiles

Implementations conformant to Implementation Profiles defined in Table 6.7 will support the following document types with the caveats and procedures given. Those document types are defined in Appendix 6A and section 6.10 of these Agreements, and in ISO 8571-2.

- FTAM-1

- FTAM-2

- FTAM-3

- NBS-6

- NBS-7

  Note: Support of this document type entails the naming of FADUs by their position in preorder traversal.

Caveat: Other methods of naming FADUs depend on the system, application, and specific file, and as such are not described here.

- NBS-8

- NBS-9

Support for any document type requires the ability to transfer and store the abstract syntax given in its definition. These Agreements do not specify techniques or formats for storage.

Caveat: Specific abstract syntaxes for the parameterized document types NBS-6,7,8 are not specified in these Agreements.

Any document type supported must be identifiable by its document type name as given in ISO 8571-2 and in Appendix 6A of these Agreements and, where defined, the parameterization scheme given in section 6.10 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.

File security issues related to NBS-9 are subject to the security agreements outlined in section 6.16.

6.18.4 Parameters for the Defined Implementation Profiles

- Use of <shared ASE information> parameter and <charging> parameter is not defined within the scope of these 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.

- Implementations will support the <contents type list> parameter on the <F-INITIALIZE> service element. The initiating service must supply a value for this parameter.

- Implementations will support the <diagnostic> parameter as stated in section 6.13 of these Agreements.

- Implementations will support <initiator identity> parameter on the <F-INITIALIZE> service element. If the initiating service user supplies a value for this parameter, it will be sent on the <request> PDU and the Virtual Filestore will process the parameter. Use must be consistent with section 6.16 of these Agreements.

- Implementations are not precluded from using other parameters for security and/or accounting. Responders must state the semantics applying to <account> and <charging> parameters. The Responder's minimum implementation is to accept but ignore the <account> and to return a <charging> value of zero.

6.18.5 Parameter Ranges for the Defined Implementation Profiles

Parameter ranges for Implementations Profiles are as stated for primitive data types in section 6.10 of these agreements.

6.18.6 File Attribute Support for Implementations

Implementations of the Implementation Profiles will support file attributes or attribute groups in the following ways.

- mandatory
  This feature is mandatory in the ISO 8571-2 standard and shall therefore be implemented by all implementations claiming conformance to these Agreements.

- supported
  This feature shall be implemented by all implementations claiming conformance to these Agreements (for attributes, this implies that at least the minimum range of attribute values, as defined in ISO 8571-2 clause 15, shall be supported). Conformant implementations shall also be able to interwork with other implementations that do not support this feature by negotiating out the corresponding features.

- optionally supported
  Implementations claiming conformance to these Agreements may or may not implement this feature (for attributes, this implies that at least either the minimum range of attribute values, as defined in ISO 8571-2 clause 15, shall be supported or that the 'no value available' result shall be

supplied). If an attribute group with a support level of 'optionally supported' is chosen to be supported, then all the attributes of this group that are classified as 'mandatory' or 'supported' shall be supported.

- o not supported
  This feature is outside the scope of these Agreements.

Kernel Group	mandatory
o Filename	mandatory
o Permitted Actions	mandatory
o Contents Type	mandatory
Storage Group	optionally supported
o Storage Account	optionally supported
o Date and Time of Creation	optionally supported
o Date and Time of Last Modification	optionally supported
o Date and Time of Last Read Access	optionally supported
o Date and Time of Last Attribute Modification	optionally supported
o Identity of Creator	optionally supported
o Identity of Last Modifier	optionally supported
o Identity of Last Reader	optionally supported
o Identity of Last Attribute Modifier	optionally supported
o File Availability	supported
o Filesize	supported
o Future Filesize	optionally supported
Security Group	optionally supported
o Access Control	supported
o Legal Qualifications	optionally supported
Private Group	not supported

Table 6.7 Implementation profile support requirements

Functional Unit	Service Class T	M	A	T&M	UNCST
Kernel	T1,T2,T3	M1	A1,A2		
Read (See note 3.)	T1,T2,T3		A1,A2		
Write (See note 3.)	T1,T2,T3		A1,A2		
Limited File Mgmnt.	SeeNote 6	M1	SeeNote 6	See	See
Enhanced File Mgmnt.		M1			
Grouping	T1,T2,T3	M1			
File Access			A1,A2		
Document Types					
FTAM-1	T1,T2,T3	[M1]	A1,A2		
FTAM-2	T2,T3	[M1]	A1,A2		
FTAM-3	T1,T2,T3	[M1]	A1,A2	Note	Note
NBS-6	[T2],T3	[M1]	[A1],A2	4	5
NBS-7	[T2],T3	[M1]	[A1],A2		
NBS-8	T3	[M1]	A2		
NBS-9	[T1],[T2] [T3]	[M1]			

Notes: to 6.18.3 and Table 6.7

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. If an implementation is capable of supporting Implementation Profile M1 and one of the T-Implementation Profiles, the Initiator may choose to request the <File Transfer and Management> service class. Any implementation so doing must be prepared for the possibility of rejection of the request by the Responder.

5. The support of the <Unconstrained> service class is optional. There are no constraints on this service class beyond those of ISO 8571.

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 6.7 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.

6.19 PROVISION OF SPECIFIC FUNCTION

6.19.1 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:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- o read a complete file

  and/or

- o write (replace, extend) to a file.

6.19.2 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:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-2 "ISO FTAM sequential text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file" (optional)
- o NBS-7 "NBS-7 FTAM random access file" (optional)
- o NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by position

  and/or

- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU.

This Implementation Profile is upwardly compatible to T1 for the transfer of unstructured files.

6.19.3 Implementation Profile T3: Full File Transfer

Implementation Profile T3 provides the function of transferring files at the external file service level for files with an unstructured, flat or general hierarchical constraint set. This includes support of the document types:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-2 "ISO FTAM sequential text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file"
- o NBS-7 "NBS-7 FTAM random access file"
- o NBS-8 "NBS-8 FTAM indexed file"
- o NBS-9 "NBS-9 file directory file" (optional)

This Implementation Profile supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by key or by position

  and/or

- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU.

This Implementation Profile is upwardly compatible to T1 for the transfer of unstructured files.

6.19.4 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:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-2 "ISO FTAM sequential text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file" (optional)
- o NBS-7 "NBS-7 FTAM random access file" (optional)

This Implementation Profile supports file transfer and file access, that is the ability to:

- o read a complete file or FADUs which are identified by position,
- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU,
- o locate and erase within files.

6.19.5 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:

- o FTAM-1 "ISO FTAM unstructured text"
- o FTAM-2 "ISO FTAM sequential text"
- o FTAM-3 "ISO FTAM unstructured binary"
- o NBS-6 "NBS-6 FTAM sequential file"
- o NBS-7 "NBS-7 FTAM random access file"
- o NBS-8 "NBS-8 FTAM indexed file"

This Implementation Profile supports file transfer and file access, that is, the ability to:

- o read from a complete file, or from a series of FADUs which are identified by key or by position,
- o write (replace, extend, insert depending on constraint set and document type) to a file or an FADU,
- o locate and erase within files.

6.19.6 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.

6.20 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 6.8 Implementation Profiles (NBS) and Profiles (SPAG/CEN-CLC)

Implementation Profile	SPAG/CEN-CLC
T1	A/111
T2	A/112
T3	A/113
A1	A/122
A2	A/123
M1	A/13

6.21 APPENDIX A: FTAM DOCUMENT TYPES

Part 1: Document Types
Part 2: Constraint Sets
Part 3: Abstract Syntaxes
Part 4: Object Identifiers and Object Descriptors used in FTAM

Part 1: Document Types

NBS-6 Sequential file document type

1. Entry Number: NBS-6

2. Information objects

Table 6.9 Information objects in NBS-6

document type name	{iso identified-organization icd (9999) organ-code (1) document type (5) sequential (6)} "NBS-6 FTAM sequential file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization icd (9999) organ-code (1) abstract-syntax (2) nbs-as1 (1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}
  Parameter0 ::= universal-class-number-0 [0] INTEGER {univer-time (23),
                                                       gen-time (24),
                                                       boolean (1),
                                                       null (5) }
  Parameter1  ::= [1] SEQUENCE  {
                  universal-class-number-1 INTEGER { int       (2),
                                                     bit       (3),
                                                     ia5      (22),
                                                     graphic  (25),
                                                     general  (27),
                                                     octet     (4)},
                  string-length INTEGER }
  Parameter2  ::= [2] SEQUENCE {
                  private-class-number INTEGER {float (0)},
                  length-1     INTEGER,
                  length-2     INTEGER }
```

file model	{iso standard 8571 file-model (3) hierarchical (1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set (4) sequential-flat (2)} "FTAM sequential flat constraint set"

```
file contents:
                Datatype1 ::= PrimType -- as defined in Annex 6 A, Part 3
                Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                         Enter-Subtree-Data-Element , Exit-
                                         Subtree-Data-Element}
```

3. Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the sequential flat constraint set (see Table 6.9) 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 6.A, Part 3 of this document. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

Note: The string length values are the actual number of characters from the specified character set, they do not include any escape sequences or overhead from the encoding.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in Table 6.9, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in Table 6.9, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2".

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice
2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g. document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 6.9 for all presentation data values transferred. An implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-6 may be simplified to the document type FTAM-3 (allowed only when reading the file). The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

11.2 Access context selection

A document of type NBS-6 may be accessed in any one of the access contexts defined in the sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied at the end of file the transferred material shall be the series of FADUs which would be generated by reading any NBS-6 document with the same parameter values in access context FA.

NBS-7 Random access file

1. Entry number: NBS-7

2. Information objects

Table 6.10 Information objects in NBS-7

document type name	{iso identified-organization icd (9999) organ-code (1) document type (5) random-file (7)} "NBS-7 FTAM random access file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization icd (9999) organ-code (1) abstract-syntax (2) nbs-as1 (1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}
  Parameter0 ::= universal-class-number-0 [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 icd (9999) organ-code(1) constraint-set (4) nbs-ordered-flat(2)} "NBS ordered flat constraint set"

```
file contents:
               Datatype1 ::= PrimType -- as defined in Annex 6 A, Part 3
               Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                        Enter-Subtree-Data-Element }
                                        Exit-Subtree-Data-Element }
```

3. Scope and field of application

The document type defines the contents of a file for storage, for transfer and access by FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the NBS-ordered-flat constraint set (see Table 6.10). These definitions appear in Appendix 6 A, Part 2 of this document.

For a specific file the number of data elements in a data unit is given by the parameters. Each data element is a data type from the set of primitive data types defined in the Annex 6.A, Part 3 of this document. Each data unit contains the same data element types in the same order as all other data units. These types are determined by the parameters 0 through 2.

Note: The string length values are the actual number of characters from the specified character set, they do not include any escape sequences or overhead from the encoding.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in Table 6.10, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in Table 6.10, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of " Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name " asname2".

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice
2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g. document type parameters and transfer syntaxes).

9.3 Sequence of presentation data values

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 6.10 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-7 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

A document of type NBS-7 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>.

11.2 Access context selection

A document of type NBS-7 may be accessed in any one of the access contexts defined in the NBS-ordered-flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied at the end of file the transferred material shall be the series of FADUs which would be generated by reading any NBS-7 document with the same parameter values in access context FA.

NBS-8 Indexed sequential file

1. **Entry Number: NBS-8**
2. **Information objects**

Table 6.11 Information objects in NBS-8

document type name	{iso identified-organization icd (9999) organ-code (1) document type (5) indexed-file (8)} "NBS-8 FTAM indexed file"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization icd (9999) organ-code (1) abstract-syntax (2) nbs-as1 (1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1) } "Basic Encoding of a single ASN.1 type"

```
parameter syntax:
  PARAMETERS ::= SEQUENCE            {DataTypes, KeyType, KeyPosition}

  DataTypes  ::= SEQUENCE OF CHOICE {Parameter0, Parameter1, Parameter2}

  KeyType    ::= CHOICE {Parameter0, Parameter1, Parameter2}

             --  Parameter0, Parameter1, Parameter2, as defined for the
             --  document types NBS-6, NBS-7

  KeyPosition::= position INTEGER
```

file model	{iso standard 8571 file-model (3) hierarchical (1)} "FTAM hierarchical file model"
constraint set	{iso standard 8571 constraint-set (4) ordered-flat (3) } "FTAM ordered flat constraint set"

```
file contents:
               Datatype1 ::= PrimType -- as defined in Annex 6 A, Part 3
               Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                        Enter-Subtree-Data-Element }
                                        Exit-Subtree-Data-Element }
```

3. Scope and field of application

The document type defines the contents of a file for storage, for transfer and access using FTAM.

Note: Storage refers to apparent storage within the Virtual Filestore.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1.

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units, each of which consists of zero, one or more data elements. The order of each of these elements is significant.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set (see Table 6.11). 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 'traversal number' are not required for conformant implementations
- The identities 'next' and 'previous' are allowed for all FADUs

Each data element is a data type from the set of primitive data types defined in Appendix 6A, Part 3 of this document. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the <DataTypes> parameter.

Note: The length values refer to the number of characters from the applicable type, not to the number of octets in the encoding, nor to the line length in any rendition of the document, where these are different.

Each data unit in the file has a key associated with it. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in Appendix 6A, Part 3.

The type and length of the key are defined by the <KeyType> parameter.

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in the following Table 6.12.

Table 6.12 Datatypes for keys

Key Type	Minimum Range (octets)	Order
ASN.1 INTEGER	(1-2)	increasing numeric value
ANS.1 IA5String	(0-16)	lexical order
ASN.1 GraphicString	(0-16)	lexical order
ANS.1 GeneralString	(0-16)	lexical order
ANS.1 OCTET STRING	(0-16)	increasing value
ASN.1 GeneralizedTime		increasing time value
ASN.1 UniversalTime		increasing time value
NBS-AS1 FloatingPointNumber		increasing numeric value

The position of the key in the data unit is specified by the <position> parameter.
position = 0 implies the key is not part of the data
position > 0 specifies the actual data element in the data unit.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 module ISO8571-FADU in ISO 8571, in which each of the file access data units has the abstract syntactic structure of NBS-AS1 as defined by the parameters.

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in Table 6.11, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in Table 6.11, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 **Presentation data values**

The document is transferred as a series of presentation data values, each of which is either

a) one value of the ASN.1 datatype "Datatype1", carrying one of the data elements from the document. All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname1" or

b) a value of "Datatype2". All values are transmitted in the same (but any) presentation context defined to support the abstract syntax name "asname2".

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice
2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between presentation data values in the same presentation context, and boundaries between P-DATA primitives, are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options (e.g. document type parameters and transfer syntaxes).

9.3 **Sequence of presentation data values**

The sequence of presentation data values of type a) and the sequence of presentation data values of types a) and b) is the same as the sequence of data elements within a Data Unit, and Data Units in the hierarchical structure, when flattened according to the definition of the hierarchical file model in ISO 8571-2.

10. **Transfer syntax**

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 6.11 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. **ASE specific specifications for FTAM**

11.1 **Simplification and relaxation**

11.1.1 **Structural simplification**

This simplification loses information.

The document type NBS-8 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

The octet representation of the transferred data is unpredictable. It will usually correspond to the data values as stored in the local Real Filestore of the Responder.

A document of type NBS-8 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>. The traversal order of the FADUs must be maintained.

Note: The traversal order is as reading the file as NBS-8 in key order.

11.2 Access context selection

A document of type NBS-8 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied the transferred material shall be the series of FADU which would be generated by reading any NBS-8 document with the same parameter values in access context FA.

The insertion of a new FADU after an already existing FADU will be indicated via a diagnostic on TRANSFER-END.

11.4 The EXTEND operation

This operation is excluded for the use with this document type.

NBS-9 File directory file

1. Entry Number: NBS-9

2. Information objects

Table 6.12 Information objects in NBS-9

document type name	{iso identified-organization icd (9999) organ -code (1) document-type (5) file directory (9)} "NBS-9 FTAM file directory file"
abstract syntax names:	{iso identified-organization icd (9999) organ -code (1) abstract-syntax (2) nbs-as2 (2)} "NBS file directory entry abstract syntax"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1)} "Basic Encoding of a single ASN.1 type"

parameter syntax

```
PARAMETERS ::= attribute-names [0] IMPLICIT BIT STRING {

            -- Kernel group

                 read_filename (0),
                 read_permitted-actions (1),
                 read_contents-type (2),

            -- Storage group

                 read_storage-account (3),
                 read_date-and-time-of-creation (4),
                 read_date-and-time-of-last-modification (5),
                 read_date-and-time-of-last-read-access (6),
                 read_date-and-time-of-last-attribute-modification(7),
                 read_identity-of-creator (8),
                 read_identity-of-last-modifier (9),
                 read_identity-of-last-reader (10),
                 read_identity-of-last-attribute-modifier (11),
                 read_file-availability (12),
                 read_filesize (13),
                 read_future-filesize (14),

            -- Security group

                 read_access-control (15),
                 read_legal-qualifications (16),

            -- Private group

                 read_private-use (17) }
```

(Continued on next page.)

Table 6.12 Information objects in NBS-9 continued.

file model	{iso standard 8571 file-model (3) hierarchical (1)} FTAM hierarchical file "FTAM hierarchical file model"
constraint-set	{iso standard 8571 constraint-set (4) unstructured (1)} FTAM unstructured "FTAM unstructured constraint set"
File contents: Datatype1 ::= FileDirectoryEntry --As defined by NBS-AS2 in Appendix A, --Part 3 of this document	

3. Scope and field of Application

This document defines the contents of a file for transfer (not for storage) using FTAM.

4. References

ISO 8571, Information Processing Systems - Open Systems Interconnection -File Transfer, Access and Management.

5. Definitions

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1

6. Abbreviations

FTAM File Transfer, Access and Management.

7. Document Semantics

The document consists of one file access data unit, which consists only of zero, one or more data elements of type <FileDirectoryEntry> (defined in NBS-AS2).

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the unstructured constraint set. These definitions appear in ISO 8571-1.

The parameter of the document type is used on <F-OPEN request> to specify the desired attributes of each of the files on the Filestore, when reading the document.

8. **Abstract syntactic structure**

The abstract syntactic structure of the document is a series of file directory entries, each of which is defined by the <FileDirectoryEntry> definition in NBS-AS2.

Additional constraints are defined for this document type: File access actions are restricted to Read. File-directory files may be Selected, Opened, Read, Closed, and Deselected. They may not be Created or Deleted. They may not be Written or Modified (except as a side effect of actions performed on individual files contained within a file directory).

9. **Definition of transfer**

9.1 **Datatype definition**

The file consists of zero or more values of Datatype1 defined in Table 6.13.

9.2 **Presentation data values**

The document is transferred as a series of presentation data values. Each presentation data value shall consist of one value of the ASN.1 data type "Datatype1", carrying one of the file directory entries from the document.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in Table 6.13.

9.3 **Sequence of presentation data values**

The sequence of presentation data values is the same as the sequence of file directory entries within the Data Unit in the file.

10. **Transfer syntax**

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 6.13 for all presentation data values transferred. Implementations shall optionally support other named transfer syntaxes.

11. **ASE specific specifications for FTAM**

11.1 **Simplification and relaxation**

Relaxation is allowed to any bitstring combination of the document type parameter.

Part 2: Constraint Sets

NBS Ordered flat constraint set

1. **Field of application**

   The NBS-ordered flat constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made on an FADU basis by position in the sequence.

2. **Basic constraints**

Table 6.14 Basic constraints for NBS Ordered flat

Constraint set descriptor	"NBS ordered flat constraint set"
Constraint set identifier	{iso identified-organization icd (9999) organ-code (1) 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'.

3. **Structural constraints**

The root node shall not have an associated data unit; all children of the root node shall be leaf nodes and may have an associated data unit; all arcs from the root node shall be of length one.

4. Action constraints

Insert: The <Insert> action is allowed only at the end of file. If the FADU identity is 'end' the new node is inserted following all existing nodes in the file. If the FADU identity is 'traversal number', the number must be at least one greater than the traversal 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 'traversal number' not greater than that of the last existing node, an error will occur.

Erase: The Erase action is only allowed at the root node to empty the file, with FADU identity of 'begin'. The result is a solitary root node without an associated data unit.

Note: It is the intention when using this constraint set to allow for emptying an FADU, i.e. leaving an FADU with a DU of data length 0 (or without a DU); afterwards data may be reinserted into this hole. In order to empty an FADU, the <Replace> operation may be used with new data of length zero (or with an FADU whose <data exists> bit is set to 'false' and no DU). Refilling the hole is accomplished by a <Replace> operation with the new DU (or with the new FADU, whose <data exists> bit is set to 'true' and the new DU).

5. Identity constraints

The FADU identity associated with the file action shall be one of the identities 'begin', 'end', 'first', 'last', 'current', 'next', 'previous' or a 'traversal number' greater than or equal to one. The actions with which these identities can be used are given in the following table.

Table 6.15 Identity constraints in NBS Ordered flat

Action	Begin	End	First	Last	Current	Next	Previous	Traversal
Locate	valid	valid	valid	valid	valid	valid	valid	valid
Read	whole		leaf	leaf	leaf	leaf	leaf	leaf
Insert		valid						leaf
Erase	whole							
Replace	whole		leaf	leaf	leaf	leaf	leaf	leaf

Part 3: Abstract Syntaxes

Abstract Syntax NBS-AS1

Abstract syntax name: {iso identified-organization icd (9999) organ-code (1) abstract-syntax (2) nbs-as1 (1)}
"NBS abstract syntax AS1"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 type NBS-AS1.PrimType

```
NBS-AS1 DEFINITIONS ::=
BEGIN
PrimType ::= CHOICE {  INTEGER,
                       BIT STRING,
                       BOOLEAN,
                       IA5String,
                       GraphicString,
                       GeneralString,
                       OCTET STRING,
                       UTCTime,
                       GeneralizedTime,
                       NULL,
                       FloatingPointNumber }

                       --   The support for IA5String is the IA5 G0
                       --   character set and the IA5 C0 set
                       --   The minimum level of support for
                       --   GraphicString is the IA5 G0 character set and
                            the 8859-1 G0 and G1 sets
                       --   The minimum level of support for
                            GeneralString is the IA5 G0 character set and
                            the 8859-1 G0 and G1 character sets, and IA5
                            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.

Abstract Syntax NBS-AS2

Abstract syntax name: { iso identified-organization icd (9999) organ-code (1) abstract-syntax (2) nbs-as2 (2) }

"NBS file directory entry abstract syntax"

This is an abstract syntax for the set of presentation data values, each of which is a value of the ASN.1 Type NBS-AS2.FileDirectoryEntry.

```
NBS-AS2 DEFINITIONS ::=

BEGIN

FileDirectoryEntry  ::=[PRIVATE 2] Read-Attributes

Read-Attributes     ::=ISO8571-FTAM.Read-Attributes

END
```

For this abstract syntax the following transfer syntax will be used

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

"Basic Encoding of a single ASN.1 type"

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.

Abstract Syntax "FTAM unstructured binary abstract syntax"

This abstract syntax is defined as DataType1 (File Contents) in Table 21 of ISO 8571-2, Annex B.

Part 4: Object Identifiers and Object Descriptors used by FTAM

1 Application Context

{ iso (1) standard (0) 8571 application-context (1) iso-ftam (1) }

"ISO FTAM"

2 Abstract Syntax

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

"FTAM PCI"

{ iso (1) standard (0) 8571 abstract-syntax (2) ftam-fadu (2) }

"FTAM FADU"

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

"FTAM unstructured text abstract syntax"

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

"FTAM unstructured binary abstract syntax"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) abstract-syntax (2) nbs-as1 (1) }

"NBS abstract syntax AS1"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) abstract-syntax (2) nbs-as2 (2) }

"NBS file directory entry abstract syntax"

3 File Model

{ iso (1) standard (0) 8571 file-model (3) hierarchical (1) }

"FTAM hierarchical file model"

4 Constraint Set

{ iso (1) standard (0) 8571 constraint-set (4) unstructured (1) }

"FTAM unstructured constraint set"

{ iso (1) standard (0) 8571 constraint-set (4) sequential-flat (2) }

"FTAM sequential flat constraint set"

{ iso (1) standard (0) 8571 constraint-set (4) ordered-flat (3) }

"FTAM ordered flat constraint set"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) constraint-set (4) nbs-ordered-flat (1) }

"NBS ordered flat constraint set"

5 Document Type

{ iso (1) standard (0) 8571 document-type (5) unstructured-text (1) }

"ISO FTAM unstructured text"

{ iso (1) standard (0) 8571 document-type (5) sequential-text (2) }

"ISO FTAM sequential text"

{ iso (1) standard (0) 8571 document-type (5) unstructured-binary (3) }

"ISO FTAM unstructured binary"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) document-type (5) sequential (6) }

"NBS-6 FTAM sequential file"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) document-type (5) random-file (7) }

"NBS-7 FTAM random access file"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) document-type (5) indexed-file (8) }

"NBS-8 FTAM indexed file"

{ iso (1) identified-organization (3) icd (9999) organization-code (1) document-type (5) file-directory (9) }

"NBS-9 FTAM file directory file"

6 Transfer Syntax

{ joint-iso-ccitt (2) asn1 (1) basic-encoding (1) }

" Basic Encoding of a single ASN.1 type"

7. CCITT 1984 X.400 BASED MESSAGE HANDLING SYSTEM

7.1 INTRODUCTION

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

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

This agreement can be used over any Transport protocol class. In particular, this MHS agreement can be used over the Transport protocol class 0 used over CCITT X.25, described in section 4.6 of this document. In addition, this MHS agreement can be used over the Transport profiles used in support of MAP (Manufacturing Automation Protocol) or TOP (Technical and Office Protocols). Note that the MAP or TOP environment must support the reduced Basic Activity Subset (BAS) as defined in X.410.

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

User Agent Layer	CCITT X.420
Message Transfer Agent Layer	CCITT X.411
Reliable Transfer Service Layer	CCITT X.410
Presentation Layer	CCITT X.410 sec. 4.2
Session Layer	See Section 5.1.1

Figure 7.1 The layered structure of this implementation agreement

7.2 SCOPE

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

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

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

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

With the exception of intra domain connections, this agreement does not cover message exchange between communicating entities within a domain even if these entities communicate via P1 or P2. Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this agreement requires the ability to exchange messages without use of bilateral agreements.

PRMD = Private Management Domain
ADMD = Administration Management Domain

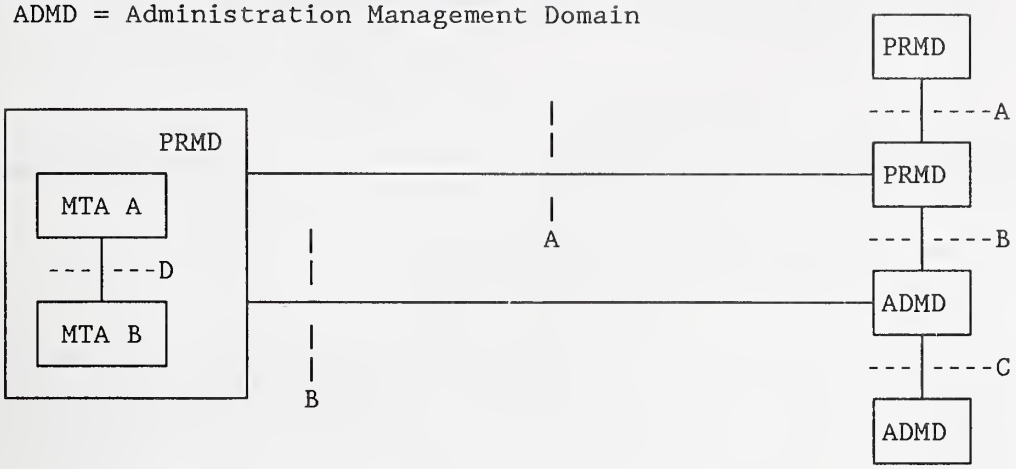


Figure 7.2 This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD; and (D) MTA and MTA

7.3 STATUS

This version of the X.400 based Message Handling System implementation agreements was completed on December 17, 1987. No further enhancements will be made to this version. See the next section--Errata.

7.4 ERRATA

7.5 PRMD to PRMD

7.5.1 Introduction

This section is limited in scope to issues arising from the direct connection (interface A in Figure 7.2) of two PRMDs. "Direct" means that no ADMD or relaying PRMD provides MHS services to facilitate message interchange. "Direct" does not exclude those instances for which Administrations happen to be ADMDs but are not providing X.400 services, that is, they are used only to provide lower layer services such as X.25. Figure 7.3 schematically represents the scope of this section.

These issues relate to the use of the UAL (User Agent Layer) and MTL (Message Transfer Layer) services, protocol elements, recommended practices and constraints. In particular, this section addresses the P1 and P2 protocols and their related services in a direct connection environment. This section describes the minimum level of services

provided by a PRMD. Provision for the use of the remaining services defined in the X.400 Series of Recommendations is beyond the scope of this section.

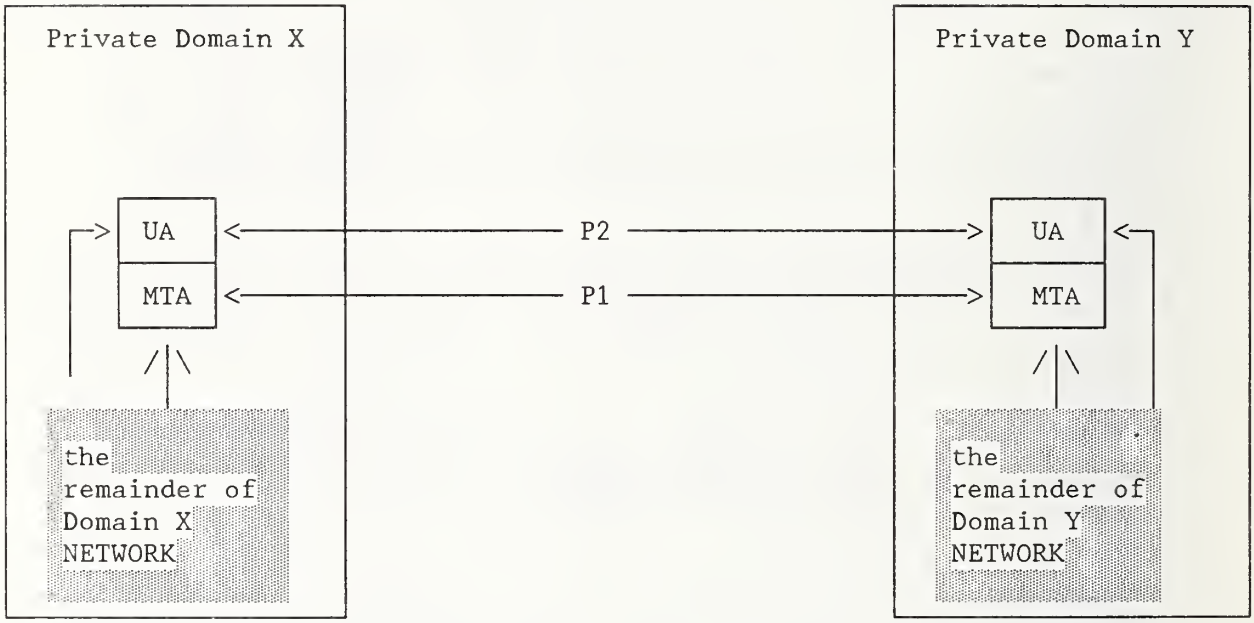


Figure 7.3 Interconnection of private domains

7.5.2 Service Elements and Optional User Facilities

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

7.5.2.1 Classification of Support for Services

The classification of UA and MT-Service elements is used to define characteristics of equipment. Equipment can claim SUPPORT or NON-SUPPORT of a Service; in the case of UA-service elements, a separate classification is given for Origination and Reception.

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

7.5.2.1.1 Support (S)

a) Support means:

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

b) Support for Origination means that:

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

c) Support for Reception means that:

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

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

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

7.5.2.1.2 Non Support (N)

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

7.5.2.1.3 Not Used (N/U)

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

7.5.2.1.4 Not Applicable (N/A)

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

7.5.2.2 Summary of Supported Services

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

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

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

7.5.2.3 MT Service Elements and Optional User Facilities

Tables 7.1 through 7.3 show the message transfer (MT) service elements and optional user facilities.

Table 7.1 Basic MT service elements

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

1 Not applicable to co-resident UA and MTA.

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

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

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

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

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

7.5.2.4 IPM Service Elements and Optional User Facilities

Tables 7.4 through 7.5 show the IPM service elements and optional user facilities.

Table 7.4 Basic IPM service elements

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

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

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

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

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

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

---

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

7.5.3 X.400 Protocol Definitions

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

7.5.3.1 Protocol Classification

The protocol classifications are defined below in table 7.7:

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.

Table 7.7 Protocol Classifications

7.5.3.2 General Statements on Pragmatic Constraints

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

b) The maximum number of recipients in a single MPDU is 32K - 1 (that is, 32767). However, no individual

limits on the number of occurrences (recipients) are placed on the following protocol elements: Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, Obsoletes and Cross References. Additionally, there is no limit on the number of Reply to Users. This is a local matter for the originating system.

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

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

7.5.3.3 MPDU Size

The following agreements govern the size of MPDUs:

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

7.5.3.4 P1 Protocol Elements

7.5.3.4.1 P1 Envelope Protocol Elements

Table 7.8 contains Protocol Elements and their classes.

Table 7.8 P1 protocol elements

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

(Continued on next page.)

Table 7.8 P1 protocol elements, Continued

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

(Continued on next page.)

Table 7.8 P1 protocol elements, Continued

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

(Continued on next page.

Table 7.8 P1 protocol elements, continued

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

Notes on Table 7.8

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

7.5.3.5 ORName Protocol Elements

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

Table 7.9 contains ORName protocol elements.

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

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

Table 7.9 ORName protocol elements

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

(Continued on next page.)

Table 7.9 ORName Protocol Elements, Continued

Notes:

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

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

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

2. At least one of these must be supplied.

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

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

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

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

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

Restriction on length is one of the types of restrictions. The reaction of implementations to a violation of this restriction is not defined by this Profile.

7.5.3.6.1 P2 Protocol - Heading

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

Table 7.10 P2 heading protocol elements

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

(Continued on next page.)

Table 7.10 P2 heading protocol elements, continued

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

(Continued on next page.)

Table 7.10 P2 heading protocol elements, continued

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

7.5.3.6.2 P2 Protocol - BodyParts

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

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

b) Privately Defined BodyParts

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

```
BodyPart  :: =  CHOICE {
  [0]IMPLICIT IA5Text,
            [1]IMPLICIT TLX,
                 .
                 .
                 .
            [234]IMPLICIT UKBodyParts,
                 .
                 .
                 .
            [310]IMPLICIT USABodyParts,
                 .
                 .
                 .                    ;
            Where UKBodyParts and USABodyParts are defined as:

              SEQUENCE {BodyPartNumber, ANY}

              BodyPartNumber ::= INTEGER
```

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

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

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

Figure 7.4 X.409 Definition of Privately Defined BodyParts

7.5.3.6.3 P2 BodyPart Protocol Elements

Table 7.11 P2 BodyParts

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

(Continued on next page.)

Table 7.11 P2 BodyParts, continued

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

(Continued on next page.)

Table 7.11 P2 BodyParts, continued

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

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

7.5.4 Reliable Transfer Server (RTS)

7.5.4.1 Implementation Strategy

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

7.5.4.2 RTS option selection

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

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

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

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

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

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

Initial turn: Kept by the initiator of the association.

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

7.5.4.3 RTS Protocol Options and Clarifications

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

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

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

c) Checkpoint size:

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

d) Window size:

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

e) APDUs should not be blocked into one activity.

f) Only one SSDU shall be transferred:

- Between two adjacent minor synch points.
- Between minor synch points and adjacent S-ACTIVITY-START and S-ACTIVITY-END requests.
- Between S-ACTIVITY-START and S-ACTIVITY-END without checkpoints.

g) A monologue association is defined as follows:

- The RTS user responsible for establishing the association is called the initiator.
- The initiator keeps the initial turn.
- APDUs are transferred in the direction of the initiator to the recipient only.
- There shall be no token passing.
- 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:

- The current session connection is of type "recover".
- 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.
- This must occur as the first activity of the next session connection for the same RTS-association. It must be the first, otherwise X.410 clause 4.5.1 point 1 is violated.

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

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

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

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

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

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

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

- 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."

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

Table 7.12 Checkpoint window size of IP

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

Legend:

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

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

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

7.5.4.4 RTS Protocol Limitations

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

Table 7.13 RTS protocol elements

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

(Continued on next page.)

Table 7.13 RTS protocol elements, continued

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

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

7.5.5 Use of Session Services

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

7.5.6 Data Transfer Syntax

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

7.6 PRMD to ADMD and ADMD to ADMD

7.6.1 Introduction

This section defines the implementation agreements that apply to the interface between two management domains when at least one is an ADMD. A message arriving at an ADMD has either no recipient within that domain or one or more recipients within that domain. In the former case, the ADMD serves as a relay between two or more domains and the actions required of that ADMD are independent of the nature (PRMD or ADMD) of the domains. In the latter case, the ADMD is responsible for delivering messages to the proper recipient(s) within its jurisdiction, and may also be responsible for relaying the message.

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

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

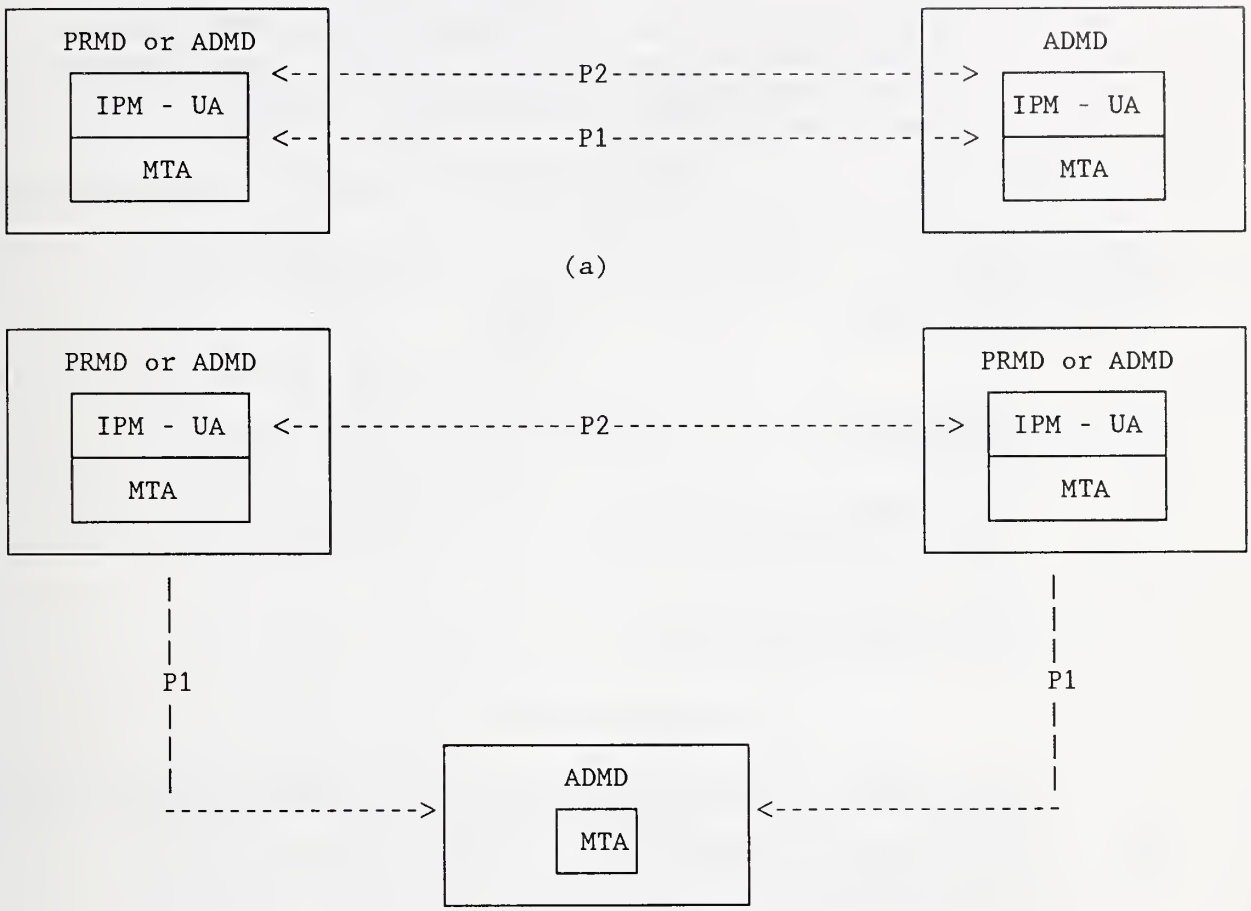


Figure 7.5 An ADMD may (b) or may not (a) serve as a relay

7.6.2 Additional ADMD Functionality

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

7.6.2.1 Relay Responsibilities of an ADMD

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

7.6.2.2 P1 Protocol Classification Changes

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

Table 7.14 P1 Protocol Classification Changes for a Delivering ADMD

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

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

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

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

7.6.2.3 O/R Names

O/R Names shall consist of:

- CountryName,
- AdministrationDomainName.

as well as one of the following:

- PrivateDomainName,
- PersonalName,
- OrganizationName,

- o OrganizationalUnit,
- o UniqueUAIdentifier,
- o X121Address.

and permits the optional inclusion of a

- o DomainDefinedAttributeList.

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

7.6.2.4 P1 ADMD Name

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

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

7.6.3 Interworking with Integrated UAs

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

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

and may optionally include a

- o DomainDefinedAttributeList.

7.6.4 Differences with Other Profiles

7.6.4.1 TTC Profile

There are no outstanding issues regarding interworking between TTC-conformant systems and NBS-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 NBS required MPDU size is 2MB as covered in section 7.5.3.3. Other differences are shown in Appendix E. TTC is currently based on Version 4 of the Implementor's Guide.

7.6.4.2 CEPT Profile

See Appendix 7E.

7.6.5 Connection of PRMDs to Multiple ADMDs

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

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

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

- o Naming,
- o Routing,
- o Replying.

It should be noted that a single PRMD that is connected to more than one ADMD can be represented by more than one combination of country-name, ADMD-name, and PRMD name. For example, it may occur that the PRMD-name for a particular PRMD may take different values, depending on the ADMD-name. Implementors should be aware of the consequences of these possibilities on routing.

7.6.6 Connection of an ADMD to a Routing PRMD

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

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

Note also that upon receipt by the ADMD of a message originating

somewhere within the PRMD collection, that the TraceInformation may contain more than one element.

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

7.6.7 Management Domain Names

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

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

7.6.8 Envelope Validation Errors

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

ADMDs will validate P1 Envelopes in the following areas:

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

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

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

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

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

   MPDU Identifier Validation

   - 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).

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

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

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

7.6.9 Quality of Service

7.6.9.1 Domain Availability

7.6.9.1.1 ADMD Availability

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

7.6.9.1.2 PRMD Availability

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

7.6.9.2 Delivery Times

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

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

Table 7.15 Delivery Time Targets

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

b) The interval(s) between retries and the number of retry attempts that an ADMD uses in attempting delivery to a

PRMD or integrated UA, will be locally determined domain parameters. However, the total elapsed times after which delivery attempts will be stopped are shown in table 7.16. This implies that, after these times, a Non-Delivery Notice will be generated.

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

Table 7.16 Forced Nondelivery Times

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

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

7.6.10 Billing Information

a) All aspects relating to billing, charging, tariffs, settlement, and in particular to the use of the billingInformation field in the delivery report, is subject to bilateral agreement, and shall not be addressed in these implementation agreements.

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

7.6.11 Transparency

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

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

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

A receiving MTA can optionally:

- o Deliver the message
- o Non-Deliver the message.

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

c) P2 extensions shall be relayed transparently by ADMDs.

7.6.12 RTS Password Management

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

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

7.6.13 For Further Study

Issues requiring further study are:

- o Intra-Domain Routing
- o Multi-Vendor Domains

7.7 INTER and INTRA PRMD CONNECTIONS

7.7.1 Introduction

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

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

work is for further study.

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

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

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

7.7.2 The Relaying PRMD

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

7.7.2.1 Relay Responsibilities of a PRMD

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

7.7.2.2 Interaction with an ADMD

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

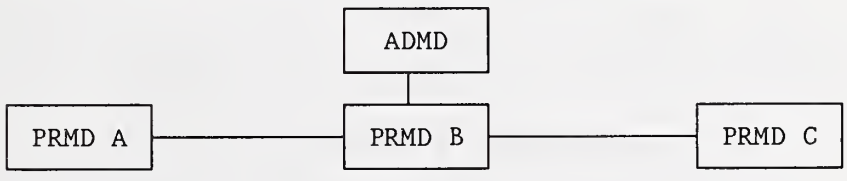


Figure 7.6 Relaying PRMD

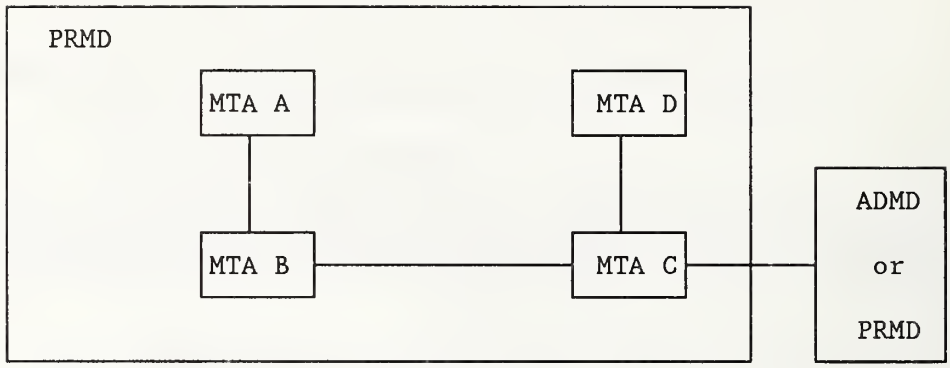


Figure 7.7 Intra PRMD connections

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

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

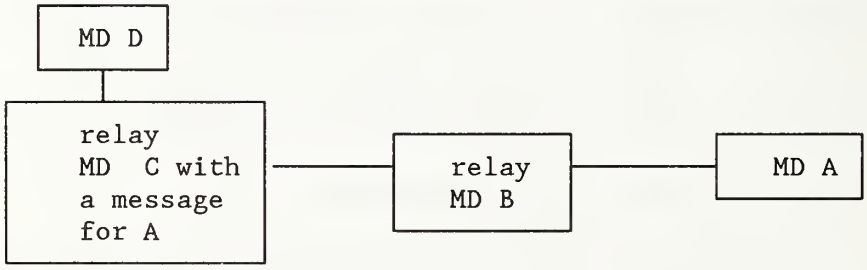


Figure 7.8 MD C must know of A to route the message

7.7.3 Intra PRMD Connections

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

7.7.3.1 Relay Responsibilities of an MTA

The relaying responsibilities of an MTA are the same as those of an ADMD (as specified in sections 7.6.8 and 7.6.2.1) with one exception: loop suppression within the domain is done using the MOTIS InternalTraceInfo protocol element. The MTA must validate the InternalTraceInfo (see section 7.8.3.5 for details on validation). In addition, the PRMD will simply deliver messages routed to it from an ADMD, even if this results in routing a message from the

ADMD to the PRMD to another ADMD (please see section 7.6.6).

7.7.3.2 Loop Suppression within a PRMD

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

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

MTAName ::= PrintableString
```

Figure 7.9 Definition of InternalTraceInfo

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

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

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

Figure 7.10 Defined Actions in MTASuppliedInfo

7.7.3.3 Routing Within a PRMD

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

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

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

7.7.3.3.1 Class Designations

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

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

table 7.17.

Table 7.17 Conformant MTA Classifications

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

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

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

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

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

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

7.7.3.3.2 Specification of MTA Classes

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

7.7.3.3.3 Consequences of Using Certain Classes of MTAs

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

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

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

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

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

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

example is intended to stimulate careful examination of the relationship between network and organizational topologies.

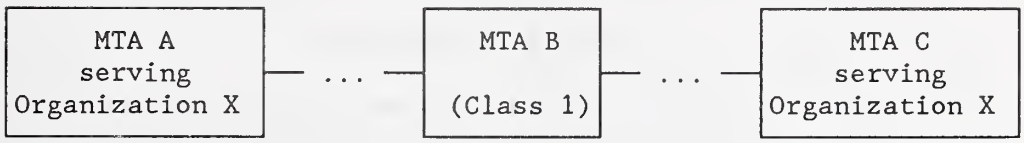


Figure 7.11 Example of a confirguration to be avoided

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

7.7.3.4 Uniqueness of MPDUidentifiers Within a PRMD

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

Two pieces of information, the MTA name and MTA designator, need to be registered within a PRMD to guarantee uniqueness. This registration facility need not be automated. If the MTA name is less than or equal to 12 characters, it is recommended that it also be used as the MTA designator.

7.7.4 Service Elements and Optional User Facilities

A PRMD made up of MTAs which support varying sets of service elements in addition to those required in these agreements may appear to provide inconsistent service for these elements. For example, if one MTA supports deferred delivery and another MTA does not, then deferred delivery can be used by some, but not all, users in the PRMD. Similarly, if one MTA supports return of contents and another does not, then a user outside of the PRMD will receive returned contents for messages sent to one user, but not for messages sent to another user. Note that this same inconsistency occurs when sending to two PRMDs which support different additional optional elements.

7.7.5 X.400 Protocol Definitions

This section describes additions and modifications to section 7.5.3 which are required for implementation of a relaying PRMD or an MTA within a PRMD.

7.7.5.1 Protocol Classification

a) The classification scheme given in section 7.5.3.1 applies to elements passing from one PRMD to another. For both relaying PRMDs, and MTAs in a PRMD, the same classification scheme will be used, but within a PRMD the classification applies to elements passing from one MTA to another.

b) In addition to the classifications given in section 7.5.3.1, a classification of Prohibited has been used.

PROHIBITED = P

This element shall not be used. Presence of this element is a protocol violation.

7.7.5.2 P1 Protocol Elements

Table 7.18 contains protocol elements and their classes. An * signifies that the classification of the protocol element has not changed from Table 7.8.

Table 7.18 P1 Protocol Elements

Element	Class	Restrictions and Comments
UMPDUEnvelope		
MPDUIdentifier	M*	This field needs to be unique within a PRMD. See sections 7.7.3.4 for the method of ensuring uniqueness.
originator	M*	It is recommended that all components of the originator's ORName be included to help ensure that reports can be returned.
TraceInformation	M*	The first MTA in the domain to receive the message adds the TraceInformation. Subsequent MTAs can update the TraceInformation in the event of conversion or deferred delivery. When a message is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This element is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. Elements are always added to the end of the sequence. (See Note 1)
InternalTraceInfo	M	MTANames within a PRMD must be unique. See section 7.7.3.4 for the method of assuring uniqueness Maximum length = 32 characters.
MTAName		
MTASuppliedInfo	M	

(Continued on next page.)

Table 7.18 P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
MTASuppliedInfo		
arrival	M	
deferred	X	This field must be generated by MTAs which perform deferred delivery.
action	M	See section 7.7.3.2 for restrictions on values of this field.
previous	X	This field must be generated by MTAs which perform rerouting.
DeliveryReportEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the report adds the TraceInformation. When a report is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. (See Note 1)
DeliveryReportContent		
intermediate InternalTraceInfo	P	If it were possible to include this field in the delivery report content, an audit and confirmed report could be provided to detect problems within a PRMD. Efforts are being made to add this field to the MOTIS definition.
DeliveredInfo		
typeOFUA	R*	It is the responsibility of the MTA generating the report to generate this element.

(Continued on next page.)

Table 7.18 P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
ProbeEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the probe adds the TraceInformation. When a probe is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. (See Note 1)

Note 1: The M classification is only applicable if an implementation is claiming conformance according to section 7.10.2, point (g) 4th bullet.

7.7.5.3 Reliable Transfer Server (RTS)

In the pUserData of PConnect, the value of applicationProtocol should be 1. This value was chosen because the agreements on intra-domain connections are not strictly P1, nor are they MOTIS. Philosophically, it would be good to choose a new application protocol identifier for these agreements, but this introduces too many practical problems. Since these agreements are closer to P1 than to MOTIS, the value of 1 will be used. This will not cause interworking problems between domains, because the only deviation from P1 is the InternalTraceInfo, which will not be present in messages transferred outside of a domain.

7.8 ERROR HANDLING

This section describes appropriate actions to be taken upon receipt of protocol elements which are not supported in this profile, malformed MPDUs, unrecognized O/R Name forms, content errors, errors in reports, and unexpected values for protocol elements.

7.8.1 MPDU Encoding

The MPDU should have a context-specific tag of 0, 1, or 2. If it does not have one of these tags, it is not possible to figure out who originated the message. Therefore, the way this error is reported is a local matter.

7.8.2 Contents

Once delivery to the UA has occurred, it is not possible to report errors in P2 information to the originator. In addition, it seems unreasonable to insist that the MTA that delivers a message ensure that the P2 content of the message is acceptable. As a result, the handling of content errors is a local matter.

7.8.3 Envelope

This section describes the handling of errors in message envelopes. Some of the error conditions described below may be detected in a recipient's O/R Name. This may limit the reporting MTA's ability to generate a nondelivery notification that accurately reflects the erroneous O/R Name in the ReportedRecipientInfo. This handling of this situation is a local matter.

7.8.3.1 Pragmatic Constraint Violations

In all cases of pragmatic constraint violation, a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of pragmaticConstraintViolation.

7.8.3.2 Protocol Violations

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

b) If a protocol element is expected to be of one type, but is encoded as another, then a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters should be generated.

7.8.3.3 O/R Names

a) The domain that has responsibility for delivering a message should also have the responsibility to send the nondelivery notification if the message cannot be delivered. Therefore, each MTA should only validate

the O/R Names of recipients with responsibility flags set to TRUE. In addition, a nondelivery notification can only be sent if the originator's O/R Name is valid.

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

c) If the O/R Name has all of the appropriate protocol elements and the message still cannot be delivered to the recipient, the following DiagnosticCodes may appear in the nondelivery report: unrecognizedORName, ambiguousORName, and uaUnavailable.

7.8.3.4 TraceInformation

a) Since non-relaying domains need not do loop suppression, domains with responsibility for delivering the message need not be concerned about the semantics of the TraceInformation, that is, arrival time and converted EncodedInformationTypes can be provided to the UA without inspection by the MTAs of the domain as long as the TraceInformation is properly encoded according to X.409.

b) When a message is accepted for relay, the relaying domain must check that a TraceInformation SEQUENCE has been added by the domain that last handled the message. If the appropriate TraceInformation was not added, this should be treated as a protocolViolation (section 7.8.3.2).

c) In addition, the relaying domain must check that the information was added in the sequence defined by the rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid,then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow a more meaningful description of this problem. A request for this

new diagnostic code will be submitted.

7.8.3.5 InternalTraceInfo

This section applies only to MTAs which follow the agreements of section 7.7.

a) When a message is accepted for relay from another MTA in the domain, the relaying MTA must check that an InternalTraceInfo SEQUENCE has been added by the MTA that last handled the message. If the appropriate InternalTraceInfo was not added, this should be treated as a protocolViolation (section 7.8.3.2).

b) In addition, the relaying MTA must check that the information was added in the sequence defined by the rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid, then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow for a more meaningful description of this problem. A request for this new diagnostic code will be submitted.

7.8.3.6 Unsupported X.400 Protocol Elements

The protocol elements defined in X.400 but unsupported by this profile are: the deferredDelivery and PerDomainBilateralInfo parameters of the UMPDUEnvelope, the ExplicitConversion parameter of RecipientInfo, and the alternateRecipientAllowed and contentReturnRequest bits of the PerMessageFlag. Appropriate actions are described below for domains that do not support the protocol elements.

7.8.3.6.1 deferredDelivery

The delivering domain shall do one of the following:

- deliver at once,
- hold for deferred delivery,
- return a nondelivery notification with a ReasonCode of unableToTransfer and a DiagnosticCode of noBilateralAgreement.

7.8.3.6.2 PerDomainBilateralInfo

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

7.8.3.6.3 ExplicitConversion

If ExplicitConversion is requested the message should be delivered if possible. That is, if the UA is registered to accept the EncodedInformationTypes of the message, then the message should be delivered even though the requested conversion could not be performed along the route. If delivery is not possible, then a nondelivery report should be generated with a ReasonCode of conversionNotPerformed with no DiagnosticCode.

7.8.3.6.4 alternateRecipientAllowed

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

7.8.3.6.5 contentReturnRequest

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

7.8.3.7 Unexpected Values for INTEGER Protocol Elements

There are three INTEGERs in the P1 Envelope. Appropriate actions are described below for domains receiving unexpected values for Priority, ExplicitConversion, and ContentType.

7.8.3.7.1 Priority

Additional values for Priority have been suggested by at least one group of implementors as upward compatible changes to the X.400 Recommendations. Therefore, if a PRMD receives an unexpected value for Priority, and this value is greater than one byte in length, a nondelivery report should be generated with a ReasonCode of unableToTransfer and DiagnosticCode of invalidParameters. If the value is less than or equal to one byte, the PRMD can either generate a nondelivery report as previously specified or interpret the Priority as normal and deliver or relay the message.

7.8.3.7.2 ExplicitConversion

When an unexpected value is received for ExplicitConversion, it should be handled as in section 7.8.3.6.3.

7.8.3.7.3 ContentType

If the ContentType is not supported by the delivering MTA, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of contentTypeNotSupported.

7.8.3.8 Additional Elements

In the absence of multilateral agreements to the contrary, receipt of privately tagged elements and protocol elements in addition to those defined in X.400 will result in a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters.

The exceptions to this are the MOTIS elements. The treatment of MPDU's containing these MOTIS extensions is described in Section 7.6.11.

7.8.4 Reports

There is no mechanism for returning a delivery or status report due to errors in the report itself. Therefore the handling of errors in reports is a local matter.

7.9 MHS USE OF DIRECTORY SERVICES

7.9.1 Directory Service Elements

a) Recommendation X.400 recognizes the need of MHS users for a number of directory service elements. Directory service elements are intended to assist users and their UAs in obtaining information to be used in submitting messages for delivery by the MTS. The MTS may also use directory service elements to obtain information to be used in routing messages. Some functional requirements of directories have been identified and are listed below.

   - o Verify the existence of an O/R name.
   - o Return the O/R address that corresponds to the O/R name presented.
   - o Determine whether the O/R name presented denotes a user or a distribution list.

- o Return a list of the members of a distribution list.
- o When given a partial name, return a list of O/R name possibilities.
- o Allow users to scan directory entries.
- o Allow users to scan directory entries selectively.
- o Return the capabilities of the entity referred to by the O/R name.
- o Provide maintenance functions to keep the directory up-to-date.

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

c) Currently, these aspects of directory service elements and procedures are under study by both the CCITT and the ISO. Both organizations are committed to the development of a single Directory Service specification for use by MHS and all other OSI based applications.

Given the incomplete nature of the ongoing activities within the CCITT and the ISO, no implementation details will be provided now for MHS use of Directory Services. Implementation agreements for MHS Use of Directory Services will be issued when current activities within the CCITT and the ISO are stable.

7.9.2 Use of Names and Addresses

a) It is recognized that these agreements enable a wide variety of naming and addressing attributes (see section 7.5.3.5 ORName Protocol Elements) wherein each PRMD may adopt particular routing schemes within its domain.

b) With the exception of the intra-domain connection agreements:

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

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

- o manual directories
- o on-line directories
- o ORName address specifications

- o ORName address translation.

d) Further, each PRMD may adopt naming and addressing schemes wherein the user view may take a form entirely different from the attributes reflected in table 7.9. And, each PRMD may have one user view for the originator form and another for the recipient form, and perhaps other forms of user addressing. In some cases (e.g., receipt notification) these user forms must be preserved within the constraints of these implementation agreements. However, mapping between one PRMD user form to another PRMD user form, via the X.400 ORName attributes of these agreements, is outside the scope of these agreements.

7.10 CONFORMANCE

7.10.1 Introduction

In order to ensure that products conform to these implementation agreements, it is necessary to define the types and degrees of conformance testing that products must pass before they may be classified as conformant. This section defines the conformance requirements and provides guidelines for the interpretation of the results from this type of testing.

This section is incomplete and will be enhanced in future versions of this Agreement. Later versions will reflect the problems of conformance testing and will outline specific practices and recommendations to aid the development of conformance tests and procedures.

7.10.2 Definition of Conformance

For this section, the term conformance is defined by the following:

a) The tests indicated for this section are intended to establish a high degree of confidence in a statement that the implementation under test (IUT) conforms (or does not conform) to the agreements of this section.

b) Conformance to a service element means that the information associated with the service element is made accessible to the user (person or process) whenever this agreement says that this information should be available.

Accessible means that information must be provided describing how a user (person or process):

- o causes appropriate information to be displayed, or
- o causes appropriate information to be obtained.

c) Conformance to P1, P2, and RTS as part of an X.400 OSI application requires that only the external behavior of that OSI system adheres to the relevant protocol standards.

In order to achieve conformance to this section, it is not required that the inter-layer interfaces be available for testing purposes.

d) Conformance to the protocols requires:

- o that MPDUs correspond to instances of syntactically correct data units,
- o MPDUs in which the data present in the fields and the presence (or absence) of those fields is valid in type and semantics as defined in X.400, as qualified by this profile,
- o correct sequences of protocol data units in responses (resulting from protocol procedures).

e) Statements regarding the conformance of any one implementation to this profile are not complete unless a Protocol Implementation Conformance Statement (PICS) is supplied.

f) The term "Implementation Under Test" (IUT) is interchangeable with the term "system" in the definition of conformance, and may refer to:

- o a domain, which may be one or more MTA's with co-located or remote UA's,
- o a single instance of an MTA and co-located UA with X.400 (P1, P2, RTS and session) software,
- o a relaying product with P1, RTS and session software,
- o a gateway product.

g) Claiming Implementation Conformance

- o An implementation which claims to be conformant as an ADMD must adhere to the agreements in sections 7.5 and 7.6.
- o An implementation which claims to be conformant as a PRMD must adhere to the agreements in section 7.5.
- o An implementation which claims to be conformant as a relaying PRMD must adhere to the agreements in section 7.5 and the appropriate sections of 7.7.

- o An implementation which claims to be conformant to the intra-domain connection agreements must adhere to the agreements in section 7.5 and the appropriate sections of 7.7.

7.10.3 Conformance Requirements

7.10.3.1 Introduction

Conformance to this specification requires that all the services listed as supported in sections 7.5, 7.6, and if appropriate, 7.7 of these agreements are supported in the manner defined, in either the CCITT X.400 Recommendations or these agreements. It is not necessary to implement the recommended practices of section 7.12, Appendix B, in order to conform to these agreements.

It is the intention to adopt, where and when appropriate the testing methodology and/or the abstract test scenarios currently being defined by the CCITT X.400 Conformance Group. However, it is recognized that formal CCITT Recommendations relating to X.400 Conformance Testing will not be available until 1988. It is also recognized that aspects of these agreements are outside the scope of the CCITT, and that other organizations will have to provide conformance tests in these cases.

7.10.3.2 Initial Conformance

This section is intended to provide guidelines to vendors who envisage having X.400 products available prior to any formal mechanism, or "Conformance Test Center" being made accessible that would allow for conformance to this product specification to be tested.

It is feasible that vendors and carriers will want to enter bilateral test agreements that will allow for initial trials to be carried out for the purposes of testing initial interworking capabilities. It is equally feasible that for the purposes of testing interoperability, only a subset of this specification will initially be tested.

Note: By claiming conformance to this subset of information the vendor or carrier CANNOT claim conformance to this entire specification.

There are two aspects to the requirements, interworking and service, as described in the following sections.

7.10.3.2.1 Interworking

The interworking requirements for conformance implies that tests be done to check for the syntax and semantics of protocol data elements for a system as defined by the classification scheme of sections 7.5.2.1.1 and 7.7.5.2. For a relay system, the correct protocol elements should be relayed as appropriate. For a recipient system, a message with correct protocol elements must not be rejected where appropriate.

7.10.3.2.2 Service

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

- IPMessage ID - only the PrintableString portion of the IPMessageId needs to be accessible.
- subject,
- primaryRecipients,
- copyRecipients,
- blindcopyRecipients,
- authorizingUsers,
- originator,
- inReplyTo,
- replyToUsers,
- importance,
- sensitivity,
- IA5Text Bodypart.

7.11 APPENDIX A: INTERPRETATION OF X.400 SERVICE ELEMENTS

The work on service element definitions is limited to those that are defined as 'supported' in section 7.5 of this specification. Furthermore it is not the intent of this section to define how information should be made available or presented to a MHS user, nor is it intended to define how individual vendors should design their products. In addition, statements on conformance to a specific service element and the allocation of error codes that are generated as a result of violations of the service should be defined in the sections on conformance and errors as part of the main product specification. The main objective is to provide clarification, where required, on the functions of a service element, and in particular what the original intent of the Recommendations were.

SERVICE ELEMENTS

The following Service Elements defined in X.400 have been examined and require further text to be added to their definitions to represent the proposed implementation of these service elements by the X.400 SIG.

The service element clarifications are to be taken in the context of this profile.

Service elements not referenced in this section are as defined in X.400.

PROBE

A PRMD need not generate probes.

If a probe is addressed to and received by a PRMD, the PRMD must respond with a Delivery Report as appropriate at the time the probe was processed.

DEFERRED DELIVERY

In the absence of bilateral agreements to the contrary, Deferred Delivery and Deferred Delivery Cancellation are local matters (i.e., confined to the originating domain) and need not be provided.

The extension of Deferred Delivery beyond the boundaries of the initiating domain is via bilateral agreement as specified in Section 3.4.2.1 of X.411.

Content Type Indication

It is required that both an originating and recipient domain be able to support P2 content type. The ability for domains to be able to exchange content types other than P2 will depend on the existence of bilateral or multi-lateral agreements.

Original Encoded Information Types Indication

It is required that both an originating and recipient domain be able to support IA5 text. Support for other encoded information types, for the purposes of message transfer between domains, will depend on the existence of bilateral or multi-lateral agreements.

The use of the 'unspecified' form of encoded information type should only be used when the UMPDU content represents an SR-UAPDU or contains an auto-forwarded IM-UAPDU.

The original encoded information type of a message is not meaningful unless a message is converted en route to the recipient. These agreements support only IA5 text, which should not undergo conversion. The original encoded information types should be made accessible to the recipient for upward compatibility with the use of non-IA5 text message body parts.

Registered Encoded Information Types

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

Delivery Notification

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

Disclosure of Other Recipients

This service is not made available by originating MTAE's to UAE's, but must be supported by relaying and recipient MTAE's.

By supporting the disclosure of other recipients the message recipient can be informed of the O/R names of the other recipient(s) of the message, as defined in the P1 envelope, in addition to the O/R Descriptors within the P2 header.

These agreements do not support initiation of disclosure of other recipients, but the information associated with it should be made accessible to the recipient for upward compatibility with support for the initiation of this service element.

Typed Body

As defined in X.400 with the addition of the Private Body Types that are to be supported. At present there is no mechanism provided within X.420 that would allow you to respond to reception of an unsupported body type.

Action taken in this situation is a local matter.

Blind Copy Recipient Indication

It should be considered that the recipient's UA acts on behalf of the recipient, and therefore may choose to disclose all BCC recipients to each other. Therefore it is the responsibility of the originating domain to submit two or more messages, depending on whether or not each BCC should be disclosed to each other BCC.

Auto Forwarded Indication

A UA may choose not to forward a message that was previously auto-forwarded. In addition there is no requirement for an IPM UA that does not support non-receipt or receipt notification to respond with a non-receipt notification when a message is auto-forwarded.

Primary and Copy Recipients Indication

It is required that at least one primary recipient be specified; however, for a forwarded message this need not be present. The recipient UA should be prepared to accept no primary and copy recipients to enable future interworking with Teletex, Fax, etc.

Sensitivity Indication

A message originator should make no assumptions as to the semantic interpretation by the recipients UA regarding classifications of sensitivity. For example, a personal message may be printed on a shared printer.

Reply Request Indication

In requesting this service an originator may additionally supply a date by which the reply should be sent and a list of the intended recipients of the reply. If no such list is provided then the initiator of the reply sends the reply to the originator of the message and any recipients the reply initiator wishes to include. The replytoUsers and the replyBy date may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request. Note that for an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Body Part Encryption

The original encoded information type indication includes the encoded information type(s) of message body parts prior to encryption by the originating domain. The ability for the recipient domain to decode an encrypted body part is a local matter. Successful use of this facility can only be guaranteed if there exists bilateral agreements to support the exchange of encrypted body parts.

Forwarded IP message Indication

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

- If forwarding a private message body part the originator of the forwarded message shall set the original encoded information types in the P1 envelope to undefined for that body part.

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

- See Appendix 7B on recommended practices for the use of the delivery information as part of Forwarded IP-message.

Multipart Body

It is the intent of multipart bodies to allow for the useful and meaningful structuring of a message that is constructed using differing body part types. For example, it is not recommended that a message made up of only IA5 text should be represented as a number of IA5 body parts, each one representing a paragraph of text.

7.12 APPENDIX B: RECOMMENDED X.400 PRACTICES

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

7.12.1 RECOMMENDED PRACTICES IN P2

1. ORDescriptor

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

2. ForwardedIPMessage BodyParts

   ForwardedIPMessage BodyParts should be nested no deeper than eight. There is no restriction on the number of ForwardedIPMessage BodyParts at any given depth.

3. DeliveryInformation

   It is strongly recommended that DeliveryInformation be supplied in both forwarded and autoforwarded message body parts. DeliveryInformation is useful when a message has multiple forwarded message body parts because without it, the EncodedInformationType(s) of the component forwarded messages cannot be deduced easily. DeliveryInformation is useful for autoforwarded messages because the EncodedInformationType of an autoforwarded message is "unspecified" and the EncodedInformationType(s) of the message cannot be determined easily without it. Absence of the EncodedInformationType(s) makes it difficult for a UA to easily determine whether the message can be rendered.

7.12.2 RECOMMENDED PRACTICES IN RTS

1. In the case where S-U-ABORT indicates a temporaryProblem, reestablishment of the session should not be attempted for a "sensible" time period (typically not less than five minutes).

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

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

   - receiving ability jeopardized (value 1)
     Possible meaning: The receiving RTS knows of an impending system shutdown.

   - local ss-User error (value 5)

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

- o irrecoverable procedure error (value 6)
  Possible meaning: The receiving RTS has had to delete a partially received APDU, even though some minor synchronization points have been confirmed.

- o non specific error (value 0)
  Possible meaning: The receiving RTS cannot handle the APDU (for example, because it was too large) and wishes to inform the sending RTS not to try again.

- o sequence error (value 3):
  Possible meaning: The S-ACTIVITY-RESUME request specified a minor synchronization point serial number which does not match the checkpoint data.

3. For purposes of identifying an MTA during an RTS Open, OSI addressing information should be used. This addressing information is conveyed by lower layer protocols and is reflected by the calling and called SSAP parameters of the S-CONNECT primitives.

   MTA validation and identification are related, but separate, functions. The mTAName and password protocol elements of the RTS user data should be used for validation, rather that identification, of an MTA. The RTS initiator and responder may independently require each other to supply mTAName and password.

   The CallingSSUserReference parameter of the S-CONNECT primitives should only have meaning to the entity that encoded it and should not be used to identify an MTA.

7.12.3 RECOMMENDED PRACTICES FOR ORName

Table 7.9 stipulates that the StandardAttributeList must contain either PrivateDomainName or OrganizationName. It is recommended that, for both originator and recipients in a private domain, the PrivateDomainName field be used.

It is recommended that there should be a DomainDefinedAttribute to be used in addressing UAs in existing mail systems, in order to curtail the proliferation of different types of DomainDefinedAttributes used for the same purpose. The syntax of this DomainDefinedAttribute conforms to the CCITT Pragmatic Constraints, and thus has a maximum value length of 128 octets and a type length of 8 octets, each of type Printable String. Only one occurrence is allowed.

This DomainDefinedAttribute has the type name "ID" (in uppercase). It contains the unique identifier of the UA used in addressing within the domain. This DomainDefinedAttribute is to be exclusively used for routing within the destination domain (i.e., once routed to that domain via the mandatory components of the StandardAttributeList); any other components of the StandardAttributeList may be provided. If they conflict delivery is not made.

The contents of this parameter need not be validated in the originating domain or any relaying domain, but simply transferred intact to the next MTA or domain.

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

There are existing mail systems that include a small set of non-Printable String characters in their identifiers. For these systems to communicate with X.400 messaging systems, either for pass-through service or delivery to X.400 users, gateways will be employed to encode these special characters into a sequence of Printable String characters. This conversion should be performed by the gateway according to a common scheme and before insertion in the ID DDA, which is intended to carry electronic mail identifiers. X.400 User Agents may also wish to perform such conversions.

It is recommended that the following symmetrical encoding and decoding algorithm for non-Printable String characters be employed by gateways. The encoding algorithm maps an ID from an ASCII representation to a PrintableString representation. Any non-printable string characters not specified in the table are covered by the category "other" in the table below.

The principal conversion table for the mapping is as follows:

Table 7B.1 Printable string to ASCII mapping

ASCII Character	Printable String Character
% (percent)	(p)
@ (at sign)	(a)
! (exclamation)	(b)
" (quote mark)	(q)
_ (underline)	(u)
((left paren.)	(l)
) (right paren.)	(r)
other	(3DIGIT)

where 3DIGIT has the range 000 to 377 and is interpreted as the octal encoding of an ASCII character.

To encode an ASCII representation to a PrintableString, the table and the following algorithm should be used:

```
IF current character is in the encoding set THEN
   encode the character according to the table above
ELSE
   write the current character;
continue reading;
```

To decode a PrintableString representation to an ASCII representation, the table and the following algorithm should be used:

```
IF current character is not "(" THEN
   write character
ELSE
   {
   look ahead appropriate characters;
   IF composite characters are in the above table THEN
      decode per above table
   ELSE
      write current character;
   }
continue reading;
```

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

7.12.4 POSTAL ADDRESSING

For domains wishing to support postal (or physical) delivery options, the following interim set of "nationally-defined" domain defined attributes are recommended. The CCITT will define Standard Attributes in support of physical delivery in its 1988 Recommendations; this is only an interim solution.

CCITT will also be addressing the services associated with physical delivery. This interim solution does not address the end-to-end service aspects of physical delivery; in particular, the following IPM service elements do not currently extend outside of the X.400 environment:

- o alternate Recipient Assignment
- o PROBE
- o Receipt Notification / Non-Receipt Notifications
- o Grade of delivery

"Delivery" means passing a message from the MTS to the physical delivery system (PDS), and not to the user (or user agent).

The following three DDAs are recommended to be used to specify a postal (or physical) address:

CNTRPC - encodes the country and postal code for postal delivery. The DDA value is of the form "Country?Postalcode" (for example, "USA?22096"). The country field is optional, the postal code is optional; the separator ("?") is not. If both country and postal code are missing, this DDA should not be specified.

PDA 1 - The country and postal code fields are free-form text.

PDA 2 - These two DDA (signifying Postal Delivery Address strings 1 and 2) form a 256 character free-form postal address. Fields are separated by a question mark ("?"). There is no implied separator between PDA1 and PDA2. The meaning of the fields are defined by each domain supporting the physical delivery interface. PDA1 contains the first 128 characters, PDA2 the next 128 characters. If the PDA string is less than 128 characters, PDA2 is not used.

For example, if the domain interprets the PDA fields as lines, the address

Mr. John Smith
Conway Steel
123 Main Street
Reston VA 22096

would be encoded as follows:

type = "PDA1" value = "Mr. John Smith?Conway Steel?123 Main Street?Reston VA"
CNTRPC = "?22096"

7.12.5 EDI use of X.400

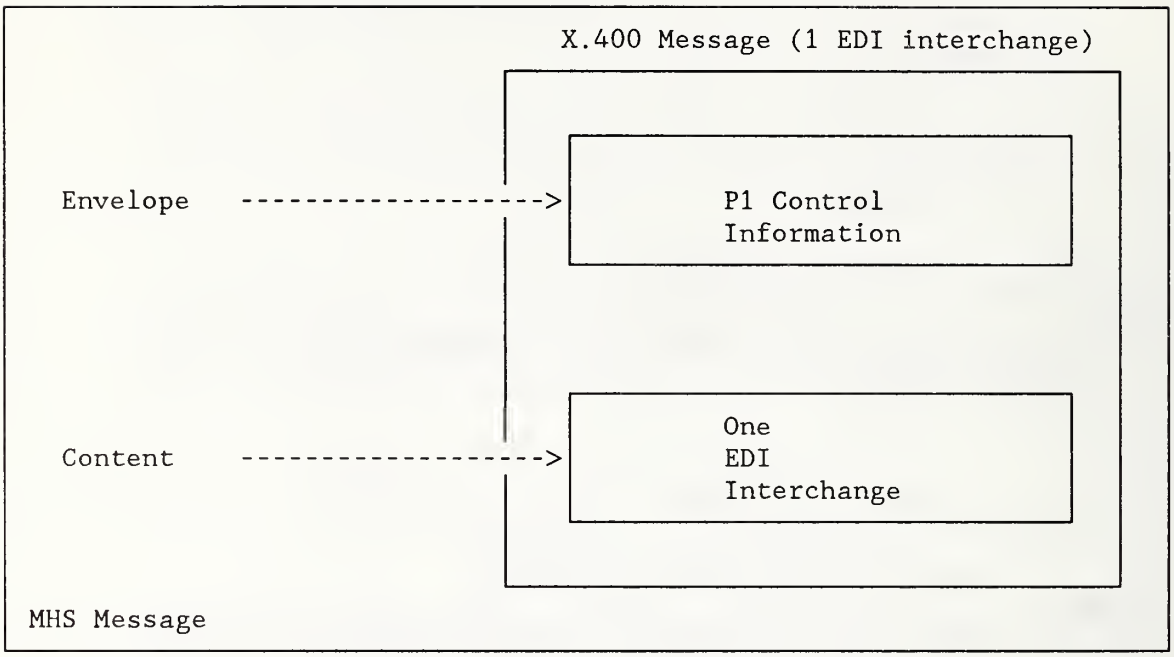
7.12.5.1 Introduction and Scope

This is a guideline for EDI data transfer in an X.400 environment conforming to the NBS agreements. These recommended practices outline procedures for use in transferring EDI transactions between trading partner applications in an attempt to facilitate actual X.400 implementation by EDI users.

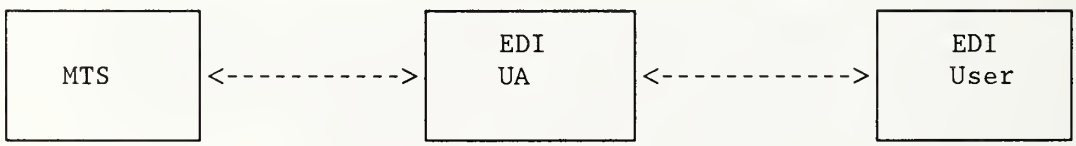
The scope of this guideline is to describe specific recommendations for adopting X.400 as the data transfer mechanism between EDI applications.

7.12.5.2 Model

The MHS recommendations can accommodate EDI through the approach illustrated below. Many Message Transfer (MT) service elements defined in the X.400 recommendations are particularly useful to the EDI application.



This diagram depicts an EDI content (1 EDI interchange) enveloped by the P1 MHS envelope. All the MT Services defined in the X.400 Recommendations may be used for EDI. However, it is not required to support optional or non-essential services to exchange EDI data between EDI users. When an EDI user submits an EDI Trade Document to the EDI User Agent, the EDI UA will submit the EDI content plus P1 envelope to the Message Transfer System (MTS).



The EDI UA must support the essential MT Services as defined in these Agreements; for example, as a minimum, to provide default values for services not elected by the EDI user, such as Grade of Delivery.

Note: MT Services are not necessarily made available by the EDI UA to the EDI user.

7.12.5.3 Protocol Elements Supported for EDI

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

Content Type

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

Original Encoded Information Types

Any EIT defined in the X.400 Recommendations may be used to specify the encoding of EDI content. However, for ANSI X12 EDI applications in particular, it is expected that the "undefined" and "Ia5Text" EIT's will normally be used, with "undefined" used to signify the EBCDIC character set.

7.12.5.4 Addressing and Routing

It is anticipated that connection of some existing systems to an X.400 service for EDI purposes will be by other than X.400 protocols, at least in the short term.

EDI messages entering the X.400 environment will therefore need to have X.400 O/R Names added to identify the origination and recipient trading partners, typically by means of local directory services in the origination domain which will map EDI identifiers/addresses into O/R Names. Such O/R Names will contain Standard Attributes as defined in Table 7.9 and for recipient trading partners will at least identify the destination domain.

In the case of trading partners outside the X.400 environment, it is expected, however, that there will be cases where message delivery will require the provision of addressing information beyond that which can be carried in Standard Attributes. In such cases, Domain Defined Attributes are recommended to be used.

The syntax of this DDA is as defined in Table 7.9, with a single occurrence having the type name "EDI" (uppercase) and a value containing the identifier/address of the trading partner. For ASC X12 purposes, specifically, this value will comprise the 2 digit interchange ID qualifier followed by the interchange ID (max 15 characters). Routing on this DDA shall only occur, if at all, in the destination domain.

7.12.6 USA Body Parts

It is recommended that UAs can generate any USA Body Part, as defined in section 7.5.3.6.2, and that they can receive such body parts as well. reception of USA Body Parts does not imply further processing by the UA, but mérely that the body part is made available, with a indication of its registered body part identifier, to another process or deposition in a file. Generation implies the reverse of this process.

7.13 APPENDIX C: RENDITION OF IA5Text AND T61String CHARACTERS

7.13.1 GENERATING AND IMAGING IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations:

CR LF	to start a new line
CR FF	to start a new page (and line)
LF .. LF	to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect. The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition. Note that X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

7.13.2 GENERATING AND IMAGING T61String

For further study.

7.14 APPENDIX D: DIFFERENCES IN INTERPRETATION DISCOVERED THROUGH TESTING OF THE MHS FOR THE CeBit 87 DEMONSTRATION

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

7.14.1 ENCODING OF RTS USER DATA

The password is defined as an ANY in the X.400 Recommendations, and implementor's groups have decided to use an IA5String for this field. There was some confusion about what the X.409 encoding for this IA5String would be, and the correct encoding is:

```
class:    context specific
form:     constructor
id code:  1
length:   length of contents
contents: (primitive encoding)
     IA5String:     16
     length:        length of contents
     contents:      the password string
class:    context specific
form:     constructor
id code:  1
length:   length of contents
contents: (constructor encoding) left as an exercise for the
          reader
```

Implementations should be prepared to receive any X.409 type for the password because of its definition as an ANY.

7.14.2 EXTRA SESSION FUNCTIONAL UNITS

One vendor proposed more than the required set of functional units on opening the session connection, and the receiver rejected the connection. All debate aside about whether the initiator should have proposed units outside of the required set, or whether the receiver should have rejected the connection, the set of functional units can be negotiated in a straightforward way. The following is recommended.

If the initiator proposes using more than the required set of functional units, the responder should specify the set of functional units that it would like to use (which should include the required set) in the open response. The session implementations will automatically use the intersection of the units proposed by both sides.

If the initiator proposes using less than the required set of functional units, the responder should reject the connection.

Unfortunately, there is not an appropriate RefuseReason for rejecting the connection, so instead of refusing the connection in the response to the S-CONNECT, the receiver should issue an S-U-ABORT with an AbortReason of protocolError. Note that it is valid to issue an S-U-ABORT instead of responding to the S-CONNECT. A problem report has been submitted to the CCITT requesting the addition of a RefuseReason for this situation.

If the responder proposes using less than the required set of functional units, the session connection is established before the initiator can check for this. If too few functional units have been proposed, the initiator should abort the connection using S-U-ABORT, with an abort reason of protocolError.

7.14.3 MIXED CASE IN THE MTA NAME

The MTA name is frequently exchanged over the telephone when two systems are being configured to communicate with one another. In one such telephone exchange, the casing of the MTA name was not specified, the MTA name consisted of both upper and lower case letters, and one of the implementations compared MTA names for equality in a case sensitive manner. Consequently, connections failed until the problem was detected and repaired. It is recommended that the MTA name be compared for equality in a case insensitive manner, and that the password be compared for equality in a case sensitive manner.

7.14.4 X.410 ACTIVITY IDENTIFIER

The X.400 Implementor's Guide recommends that the activity identifier be X.409 encoded, but this is only a recommendation and not a requirement. Consequently, receiving systems cannot assume that the activity identifier will be X.409 encoded.

7.14.5 ENCODING OF PER RECIPIENT FLAG AND PER MESSAGE FLAG

In the definition of the PerRecipientFlag in X.411, there is a statement that the last three bits are reserved, and should be set to zero. It is unclear whether those bits are unused in the X.409 encoding. Receivers should accept encodings with either zero or three unused bits. A problem report has been submitted to the CCITT asking for clarification.

Though there is not any statement in X.411 about the last four bits of the PerMessageFlag, some vendors have encoded this with zero unused bits, and some have encoded it with four unused bits. The PerMessageFlag should be encoded with at least four unused bits.

7.14.6 ENCODING OF EMPTY BITSTRINGS

There are three valid encodings for an empty bitstring: a constructor of length zero, a constructor of indefinite length followed by the end-of-contents terminator, and a primitive of length one with a zero octet as the value.

7.14.7 ADDITIONAL OCTETS FOR BITSTRINGS

Nothing in X.409 constrains an implementation from sending two, three, four, or even more octets for a bitstring that fits into one octet, with the undefined bits set to zero. Note that the number of excess octets is bounded by the pragmatic constraints guidelines of the CCITT X.400 Implementor's Guide for all of the bitstrings in P1.

7.14.8 APPLICATION PROTOCOL IDENTIFIER

If a value other that 1 is received in the applicationProtocol of the pUserData in the PConnect, NBS 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 NBS implementations accept the value of 8883 without supporting the MOTIS service elements, they would be misrepresenting themselves. To make matters worse, CEPT uses a value of 1, but relays MOTIS elements, which means that MOTIS elements will be relayed to implementations using a value of 1 to demonstrate that they do not support MOTIS. Work is continuing to try to find a solution that will allow European implementations to interwork with U.S implementations.

7.14.9 INITIAL SERIAL NUMBER IN S-CONNECT

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

7.14.10 CONNECTION DATA ON RTS RECOVERY

It is clarified that the ConnectionData is identical in both the S-CONNECT.request and the S-CONNECT.response. The value of the ConnectionData is the old Session Connection Identifier.

7.14.11 ACTIVITY RESUME

If an activity is being resumed on a new session connection, it is not clear from X.410 and X.225 whether all four of the called-ss-user reference, the calling-ss-user reference, the common reference, and the additional reference information should be specified in the S-ACTIVITY-RESUME, or whether one of the ss-user-references should be absent. It is also unclear whether the called-ss-user reference should be identical to the calling-ss-

user reference if both are present. Consequently, receivers should be tolerant of this situation. Appropriate problem reports will be submitted to the CCITT asking for clarification.

7.14.12 OLD ACTIVITY IDENTIFIER

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

7.14.13 NEGOTIATION DOWN TO TRANSPORT CLASS 0

For European implementations, X.410 specifies that class 0 transport must be supported. However, it is permissible for an initiator to propose a higher class as the preferred class, provided that class 0 appears as the alternate class in the T-Connect PDU. A responding implementation can choose to use either the preferred or alternate class, but again, must be able to use class 0. In other words, for private to private connections in Europe, class 0 transport is required.

This conflicts with the NBS agreements, since class 0 is only required if one of the partners in a connection is an ADMD.

7.15 APPENDIX E: WORLDWIDE X.400 CONFORMANCE PROFILE MATRIX

Y CONFORMANCE (E)
implies a conformance problem for European products in the U.S.

Y CONFORMANCE (US)
implies a conformance problem for U.S. products in Europe.

- The A/311 profile is specified in Env 41 202, the A/3211 profile in Env 41 201

- No TTC protocol classification for RTS exists.

- The notation X/Y indicates "X" for PRMDs and "Y" for ADMDs, i.e. "M/G" would be Mandatory for PRMDs and Generatable for ADMDs.

Table 7E.1 Protocol element comparison of RTS

RTS element	NBS	A/311	A/3211	PROBLEM Y/N
PConnect	M	M	M	N
DataTransferSyntax	M 0	M 0	M 0	N
PUserData	M	M	M	N
checkpointSize	H	H	H	N
windowSize	H	H	H	N
dialogueMode	H	H	H	N
connectdata	M	M	M	N
applicationProtocol	G 1 H 8883	H 1	R 8883	N
ConnectionData				
Open	G	G	G	N
Recover	G	H	G	N
Open				
RTSUserData	G	G	G	N
Recover				
SessionConnectionID	G	G	G	N
RTSUserData				
MTAName	G	G	G	N
Password	G	G	G	N
null	G	G	G	N
SessionConnectionID				
CallingUserReference	M	M	M	N
CommonReference	M	M	M	N
AdditionalRefInfo	H	H	H	N
PAccept	G	G	G	N
DataTransferSyntax	M 0	M 0	M 0	N

(Continued on next page.)

Table 7E.1 Protocol element comparison of RTS, continued

RTS element	NBS	A/311	A/3211	PROBLEM (Y/N)
PUserData	M	M	M	N
CheckpointSize	H	H	H	N
WindowSize	H	H	H	N
ConnectionData	M	M	M	N
PRefuse	G	G	G	N
RefuseReason	M	M	M	N
SSUserData (in S-TOKEN-PLEASE)	G	G	G	N
AbortInformation (in S-U-ABORT)	G	G	G	N
AbortReason	H	H	H	N
reflectedParameter	X	X	X	N

Table 7E.2 Protocol element comparison of P1

P1 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
ORname					
StandardAttributeList	M	M	M	M	N See Note 4
DomainDefAttributeList	X	X	X	G	Y See Note 5
StandardAttributeList					
CountryName	R	R	R	M	N
		ISO R	R		N
		X.121 H	H		Y Conformance (E)
		Other X	X		Y Prot Vio
AdministrationDomainName	R	R	G	M	N
... if PrintableString		R	G		N
... if numericString		H	H		Y Conformance (E)
X.121 Address	X	X/R	X		Y Conf(US)See Note 1
Terminal ID	X	X/G	X		Y Conf(US)See Note 1
PrivateDomainName	G	G	G	G	N
OrganizationName	G	G	G	G	N
UniqueUAidentifier	X	X/G	X		Y Conf(US)See Note 1
PersonalName	G	G	G	G	N
OrganizationalUnit	G	G	G	G	N
DomainDefinedAttribute	X	X	X	G	N
Type	M	M	M	M	N
Value	M	M	M	M	N
PersonalName					
Surname	M	M	M	M	N
GivenName	G	G	G	G	N
Initials	G	G	G	G	N
GenerationQualifier	G	X	X	X	Y Conformance (E)
GlobalDomainIdentifier					
CountryName	M	M	M	M	N
AdministrationDomainName	M	M	G	M	Y Proto Vio
PrivateDomainIdentifier	R/H	H	R	M/X	N
MPDU					
UserMPDU	G	G	G	G	Y TTC required MPDU size is 32K
DeliveryReportMPDU	G	G	G	G	N
ProbeMPDU	H	H	H	H	N

(Continued on next page

Table 7E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
UserMPDU					
UMPDUenvelope	M	M	M	M	N
UMPDUcontent	M	M	M	M	N
UMPDUenvelope					
MPDUidentifier	M	M	M	M	N
originatorORname	M	M	M	M	N
originalEncodedTypes	G	H	H	G	Y Conformance (E)
ContentType	M	M	M	M	N
UAcontentID	H	H	H	H	N
Priority	G	G	G	G	N
PerMessageFlag	G	G	G	G	N
DeferredDelivery	X	X	X	X	N
PerDomainBilatInfo	X	X	X	X	N
RecipientInfo	M	M	M	M	Y TTC MPDU 32K
TraceInformation	M	M	M	M	N
MOTIS-> LatestDelivery			X		N
MOTIS-> InternalTraceInfo	M/P		P		N
UMPDUcontent	M	M	M	M	N
MPDUidentifier					
GlobalDomainIdent	M	M	M	M	N
IA5string	M	M	M	M	N
PerMessageFlag					
DiscloseRecipients	H	G @ MTL	H	H	Y Conformance (US)
		H at UA	?		Y Conformance (US)
ConversionProhibited	G	G	G	G	N
AlternatRecipAllowed	H	G @ MTL	H	X	Y Conformance (US)
		H at UA	?		Y Conformance (US)
ContentReturnRequest	X	X	X	X	
MOTIS-> redirectionProhibited			X		N
PerDomainBilateralInfo					
CountryName	M	M	M	M	N
AdminDomainName	M	M	G	M	Y Prot Vio
MOTIS-> PrivateDomainName			G		N
BilateralInfo	M	M	M	M	N

(Continued on next page

Table 7E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	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

(Continued on next page

Table 7E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
ProbeEnvelope					
probe	M	M	M	M	N
originator	M	M	M	M	N
ContentType	M	M	M	M	N
UAcontentID	H	H	H	H	N
originalEncInfoTypes	G	H	H	G	Y Conformance (E)
TraceInformation	M	M	M	M	N
PerMessageFlag	G	G	G	G	N
ContentLength	H	H	H	H	N
PerDomainBilatInfo	X	X	X	X	N
RecipientInfo	M	M	M	M	Y TTC 256 max
MOTIS-> InternalTraceInfo	M/P		P		N
RecipientInfo					
RecipientORname	M	M	M	M	N
ExtensionIdentifier	M	M	M	M	N
PerRecipientFlag	M	M	M	M	N
ExplicitConversion	X	X	X	X	N
MOTIS-> OriginatorReqAlternatRecip			X		N
MOTIS-> ReassignmentInfo			X		N
PerRecipientFlag					
ResponsibilityFlag	M	M	M	M	N
ReportRequest	M	M	M	M	N
UserReportRequest	M	M	M	M	N
TraceInformation					
GlobalDomainIdent	M	M	M	M	N
DomainSuppliedInfo	M	M	M	M	N

(Continued on next page

Table 7E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
DomainSuppliedInfo					
arrival	M	M	M	M	N
deferred	X	X	X	X	N
action	M	M	M	M	N
(0=relayed)	G	G	G		N Note: Re-routing not required.
(1=rerouted)	H	H	H		N
MOTIS-> (2=recipientReassigned)			H		N
converted	H	G	H	H	Y Conformance(US)
previous	H	G	G	X	Y Conformance(US) (Note: G is inconsistent with action (relayed) being "H".)
ORname					
EncodedInformationTypes					
BitString	M	M	M	M	N See Note 3
G3NonBasicParameters	X	X	X	X	N
TeletexNonBasicParams	X	R	X	X	Y Conformance(US)
PresentationAbilities	X	X	X	X	N
DeliveryReportMPDU	G	G	M	G	N
DeliveryReportEnvelop	M	M	M	M	N
DeliveryReportContent	M	M	M	M	N
DeliveryReportEnvelope					
report	M	M	M	M	N
originator ORname	M	M	M	M	N
TraceInformation	M	M	M	M	N
InternalTraceInfo	M/P		P		N

(Continued on next page

Table 7E.3 Protocol element comparison of P2

P2 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
UAPDU					
IM_UAPDU	G	G	G	G	N
SR_UAPDU	X	X	X	X	N
IM_UAPDU					
Heading	M	M	M	M	N
Body	M	M	M	M	N
Heading					
IPmessageID	M	M	M	M	N
Originator ORname	R	R	R	M/G	N
AuthorizingUsers	H	H	H	H	Y TTC 16 max
PrimaryRecipients	G	G	G	G	Y TTC 256 max
CopyRecipients	G	G	G	G	Y TTC 256 max
BlindCopyRecipients	H	H	H	H	Y TTC 256 max
InReplyTo	G	G	G	G	N
Obsoletes	H	H	H	H	Y TTC 8 max
CrossReferences	H	H	H	H	Y TTC 8 max
Subject	G	G	G	G	N
ExpiryDate	H	H	H	H	N
ReplyBy	H	H	H	H	N
ReplyToUsers	H	H	H	H	Y TTC 32 max
Importance	H	H	H	H	N
Sensitivity	H	H	H	H	N
Autoforwarded	H	H	H	H	N
MOTIS-> CirculationList			X		N
MOTIS-> ObsoletingTime			X		N
IPmessageID					
ORname	H	H	H	H	N
PrintableString	M	M	M	M	N
ORdescriptor					
ORname	H	H	H	┐	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

(Continued on next page

Table 7E.3 Protocol element comparison of P2, continued

P2 Protocol	NBS	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

(Continued on next page

Table 7E.3 Protocol element comparison of P2, continued

P2 Protocol	NBS	A/311	A/3211	TTC	PROBLEM (Y/N)
BODYPART SUPPORT					
o IA5 Text	G	G	G		N See Note 7
o TLX	X	X	X		N
o Voice	X	X	X		N
o G3FAX	X	X	X		N
o TIFO	X	X	X		N
o TTX	X	X/H	X		Y Conf(US)See Note 2
o VideoTex	X	X	X		N
o NationallyDefined	X	X	X		N
o Encrypted	X	X	X		N
o ForwardedIPmessage	H	H	H		N
o SFD	X	X	X		N
o TIFI	X	X	X		N
MOTIS-> o ODA			X		N
MOTIS-> o ISO6937 Text			H		N

Note 1: It should be noted that the A/311 profile states: For routing all ADMDs should support all Form 1 Variants of O/R Name. All PRMDs should support at least Form 1, Variant 1 form of OR Name.

Note 2: It should also be noted that the A/311 profile requires that all ADMDs should support the reception of Teletex body parts for delivery to their own UAEs.

Note 3: An A/3211 implementation may generate MOTIS encoded information types. See 7.6.11.

Note 4: Only Form 1 Variant 1 of O/Rname shown for TTC, but TTC defines other forms and variants. Form 1 Variant 1 recommended for PRMDs and ADMDs, Form 1 Variant 2 also recommended for ADMDs.

Note 5: DDA's can be used to specify recipients in any Japanese domains other than TTC. Assignment of DDAs for UAs within TTC domains is not recommended.

Note 6: One of [DeliveredInfo/NonDeliveredInfo] must be present. TTC encodes this as shown. Other profiles represent this by classifying both protocol elements as generatable. A similar situation exists with the P2 ORdescriptor.

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

7.16 APPENDIX F: INTERWORKING WARNINGS

ADMD name is to be encoded as a single space when configurations with no ADMD's are present. It should be noted that this may change in January 1988 so that the ADMD name is encoded as a zero length element in such cases.

The NBS agreements allow implementation to generate MPDUs with no body parts. Such MPDUs will be rejected by European-conformant systems. (Note this situation may change in January 1988)

In order to optimize the number of recipients you can read and reply to, it is advisable to be able to generate all standard O/R name attributes.

8. DIRECTORY SERVICES PROTOCOLS

8.1 INTRODUCTION

This is an Implementation Agreement developed by the Implementor's Workshop sponsored by the U.S. National Bureau of Standards 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 Directories based on the 1988 Recommendations/International Standard from the CCITT and ISO joint working groups on Directories (hereafter referenced as Directory documents). Versions of this document will stay consistent with the latest drafts of those Directory Documents. Figure 8.1 displays the structure of this Implementation Agreement.

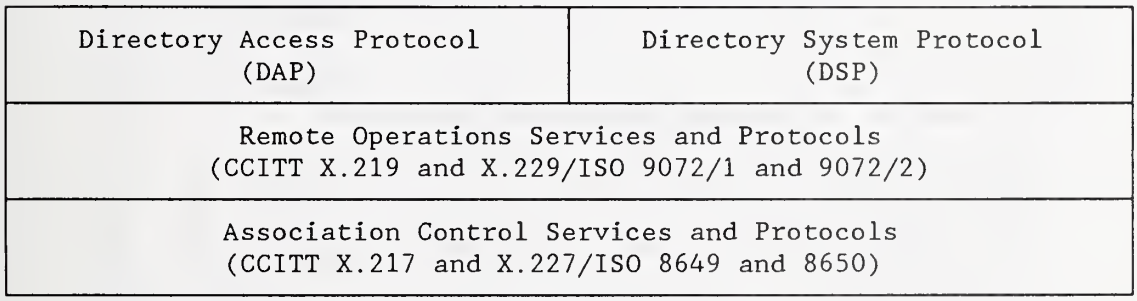


Figure 8.1 Structure of this Implementation Agreement

The Directory User Agents (DUAs) and Directory Service Agents (DSAs) provide access to the Directory on behalf of humans and applications such as Message Handling and File Transfer, Access, and Management. See the Scope and Field of Application section for more information on the model used in Directories.

This document covers both the Directory Access Protocol (DAP) and the Directory System Protocol(DSP) defined in the Directory documents. A good working knowledge of the Directory documents is assumed by this chapter. All terminology and abbreviations used but not defined in this text may be found in those documents.

8.2 SCOPE AND FIELD OF APPLICATION

Stand alone and distributed directories can both be accommodated in this Agreement by the appropriate choice of protocols and pragmatic constraints from those specified. Figure 8.2 illustrates a stand alone directory and Figure 8.3 illustrates a distributed directory.

This agreement does not cover interaction between co-located entities, such as a co-resident DUA and DSA. It also does not specify the interface between a user (person or application) and a DUA. Bilateral agreements between a DUA and DSA or DSA and DSA may be implemented in addition to the requirements stated in this document. Conformance to this agrement requires the ability to interact without the use of bilateral agreements other than those required in the Directory documents.

The logical structure of the Directory Information Base (DIB) is described in the Directory documents. The manner in which a local portion of the DIB is organized and accessed by its DSA is not in the scope of this agreement.

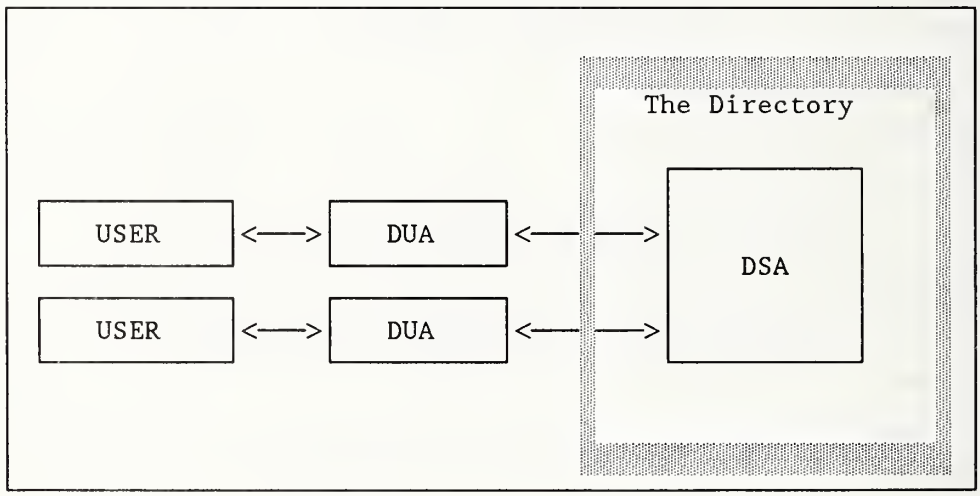


Figure 8.2 Stand-alone Directory Model

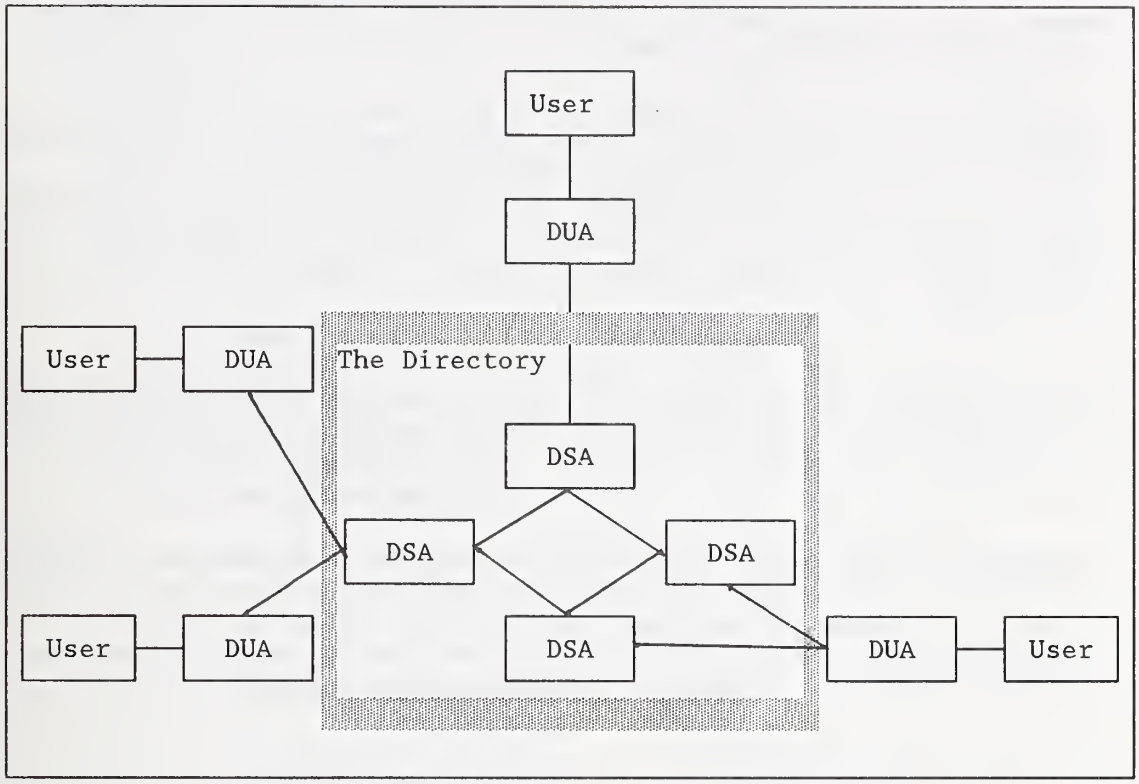


Figure 8.3 Distributed Directory Model

8.3 STATUS AND REFERENCES

This Implementation Agreement is conformant with references found in the References section of this document.

8.4 Use of Directories

Given the rapid multiplication and expansion of OSI applications, telecommunication systems and services, there is growing need for users of, 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, Directories 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, Directories, 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. The Directory permits the imposition of rules by means of the Directory Schema (Section 8.6 below); but the Directory Schema itself should be capable of alteration by Directory management.

8.5 Directory ASEs, Application Contexts, and Ports

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.

Thus, each Directory application context defines the operations needed by two communicating Directory entities.

Access to the services provided by an application context is through one or more directory ports. The point of access is called an Access Point (see Figure 8.4). Each access point corresponds to a particular combination of port types.

The following ASEs are described in the Directory Document:

- Directory Read ASE
- Directory Chained Read ASE
- Directory Search ASE
- Directory Chained Search ASE
- Directory Modify ASE
- Directory Chained Modify ASE

ROSE and ACSE also form part of the Directory Application Contexts.

The following Application Contexts are described in the Directory Document:

- Directory Access
- Directory System

The following Ports are described in the Directory Document:

- Read Port
- Modify Port
- Search Port
- Chained Read Port
- Chained Search Port
- Chained Modify Port

The ports cited above are to be specified for a particular Access Point to the Directory as illustrated by Figure 8.4.

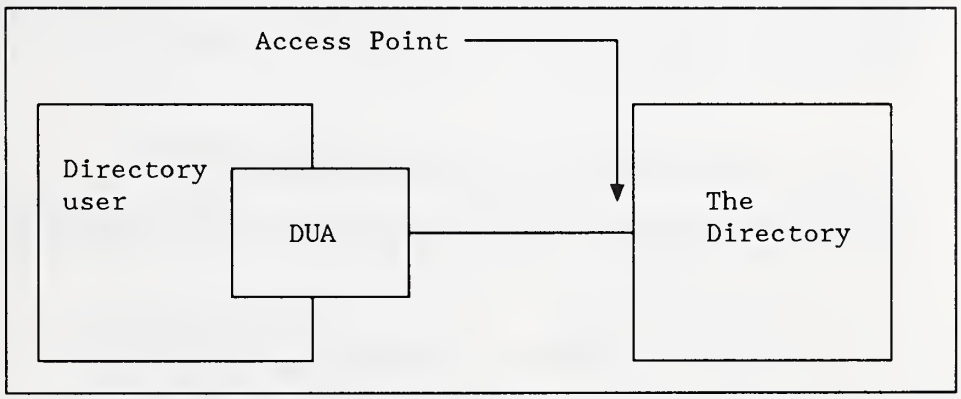


Figure 8.4 Access to the Directory

8.6 Schemas

There are four (4) major topics that relate to schemas:

8.6.1 Maintenance of Structures and Naming Rules

DSAs shall be capable of supporting the object classes and the associated structure and naming rules defined in the Directory documents, part 7, annex B, in the sense that DITs may be constructed and within these rules shall be supported.

8.6.2 Maintenance of object classes and subclasses

DSAs shall be able to support the storage and use of the following object classes defined in the Directory documents part 7, annex B:

top	alias
country	application entity
locality	organizational-person
organization	residential-person
organizational unit	application-process
DSA	

Use of the modification and local extension mechanisms provided by registered and unregistered object classes is optional.

8.6.3 Maintenance of Attribute Types

DSAs shall be able to support the storage and use of attribute type information, as defined in 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.

8.6.4 Maintenance of Attribute Syntaxes

Suggested methods for the interpretation of selected Attribute Syntaxes is defined in Appendix B.

8.7 Classification of Support for Attribute Types

This section classifies directory support for selected attribute type specified in the Directory documents.

Classification of support for selected attribute types is either mandatory or optional.

8.7.1 Mandatory Support

The Directory must be able to support these Attribute Types:

Aliased Object Name	Postal Address
Business Category	Postal Code
Common Name	Preferred Delivery Mode
Country Name	Presentation Address
Locality Name	Role Occupant
ISDN Address	See Also
Member	Serial Number
Object Class	State or Province Name
Organization Name	Street Address
Organization Unit Name	Supported Application Context
Owner	Teletex Number
Post Office Box	Telephone Number
	Title
	User Password
	X.121 Address

Note: Support of these Attribute Types implies full support of the relevant Attribute Syntaxes.

8.7.2 Optional Support

Directory support of these attribute types is considered optional:

Authority Revocation List	Registered Address
CA Certificate	Search Guide
Certificate Revocation List	
Description address	T.73 Presentation Capabilities
Destination IndicatorTeltex	Non-Basic Parameter
Facsimile Telephone Number	Teletex Terminal ID
G3 Non-Basic Parameter	Telex Number
Knowledge Information	User Number
Physical Delivery Office Name	

Note: DSAs should consider initial support of the Attribute Syntax relevant to any Attribute Type for which future support is planned, in addition to those required for mandatory Attribute Types.

8.8 Introduction to Pragmatic Constraints

The following sections of this document define the pragmatic constraints to which a conformant implementation must adhere in addition to those specified in the Directory documents. The pragmatic constraints are divided into two areas. The first includes those

aspects of pragmatic constraints which apply to the scope of service. The second includes those aspects of pragmatic constraints which are specific to particular attribute types.

8.9 General Constraints

8.9.1 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:

- o T.61
- o PrintableString
- o NumericString

8.9.2 APDU Size Considerations

In the process of chaining requests it is possible that a chaining DSA may receive invoke or return APDUs that exceed its capacity:

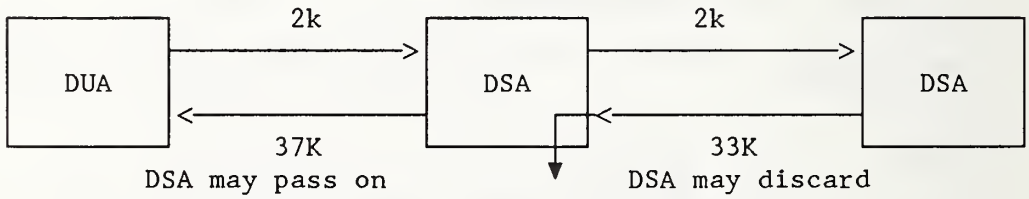


Figure 8.5 APDU Exchange

It is a minimum requirement that invoke and return result APDUs must be accepted unless they exceed 32767 octets in size; in this case they may be discarded as illustrated in the right side of figure 8.5, and an "unwillingToPerform" error reporting service shall be used.

8.9.3 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.

8.9.4 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 10.6 may be considered as an example by DSAs that implement this Service Control.

- The DSA maintains three logical queues corresponding to the three priority levels.
- The DSA Scheduler is separate and distinct from any scheduling function provided by the underlying operating system or control program services.
- The DSA Scheduler presents jobs to the Underlying Operating Services for execution and always present jobs of a higher priority before those of a lower priority.
- The DSA Scheduler will not preempt a request once it has been passed to the underlying operating system service.

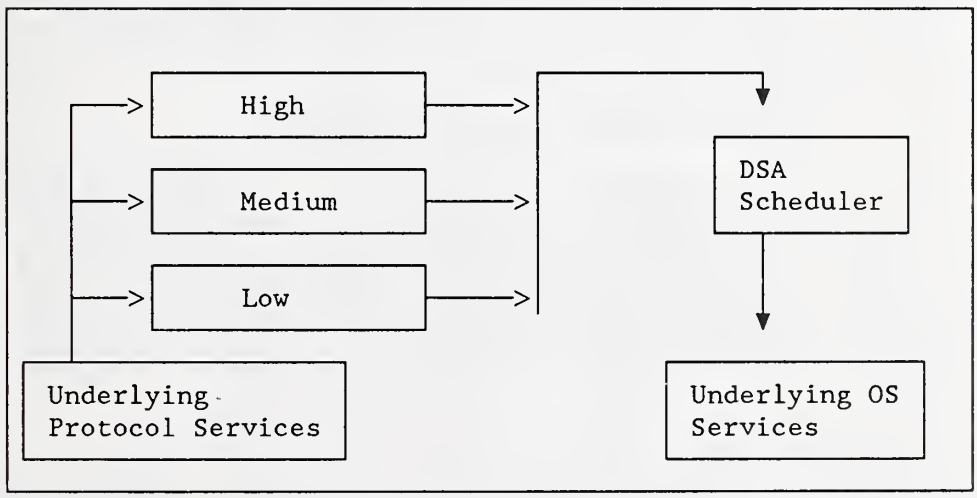


Figure 8.6 Logical DSA Application Environment

8.10 Constraints on Operations

There are no overall constraints upon service arguments or results except those implied in section 10.9.2 of this document.

8.10.1 Filters

It is required that DSAs, at a minimum, support 8 nested "Filter" parameters, and a total limit of 32 Filter Items. If these limits are exceeded, the recipient of that SearchArgument may return the ServiceProblem "unwillingToPerform".

8.10.2 Errors

There are no constraints upon any Error service except the APDU size limit as defined in section 8.9.2.

8.11 Constraints on Attribute Types

This section defines the pragmatic constraints specific to particular attribute types.

8.11.1 Attribute Values

This section describes the pragmatic constraints for attribute values. Each constraint 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 must not exceed and which a receiving entity must accept and process. A sending entity need not be capable of sending attribute values as large as the Length Constraints.

Use of Pragmatic Constraints for Strings

The Length Constraint for strings is expressed as the number of allowable characters and the number of allowable octets. When using the Printable String ASN.1 data type, the number of octets equals the number of characters. When using the T.61 ASN.1 data type, the number of octets is twice the number of characters. This is because some T.61 characters occupy two octets per character.

Attribute Types

Tables 8.1 and 8.2 specify the pragmatic constraints for selected attribute type specified in the Directory documents; many of these constraints also appear and are the same on the CCITT version of the Directory documents.

Alphabets

T.61 Strings used as attribute values shall only encode graphic characters and spaces. (They must not contain formatting characters (such as subscript) or other control characters.

Table 8.1 Pragmatic Constraints for Selected Attributes.

Attribute Type	Content	Number of Printable Characters	Number of Allowable Octets	Status	Notes
Object Class	Object Identifier		256		
Common Name	T.61 or Printable String	64	128		X.500
Serial Number	Printable String	64	64		X.500
Country Name	Printable String				ISO 3166 String only
Locality	T.61 or Printable String	128	256		X.500
Postal Address	T.61 or Printable String	30 x 6	60 x 6		Note 4 X.500 & UPU
Postal Code	T.61 or Printable String	40	80		UPU X.500
Organization Name	T.61 or Printable String	64	128		X.500
Organizational Unit Name	T.61 or Printable String	64	128		X.500

Table 8.1 Pragmatic Constraints for Selected Attributes. (continued)

Attribute Type	Content	Number of Printable Characters	Number of Allowable Octets	Status	Notes
Title	T.61 or Printable String	64	128		X.500
Description	T.61 or Printable String	1024	2048		X.500 About 1 Screen full
Search Guide					ffs
Business Category	T.61 or Printable String	64	128		
Facsimile Telephone Number	Facsimile Telephone Number		ffs		CCITT E.163 X.500 Includes G3 Non Basic Parameters
ISDN Address	Numeric String	n/a	16+40		CCITT E. 164 X.500
Presentation Address	Presentation Address	n/a	224		Note 2,ISO 7498 X.500, & X.200
Telephone Number	Printable String	32	32		CCITT E.163 X.500
Teletex Terminal Identifier	Teletex Terminal Identifier		ffs		CCITT F.200 X.500 Includes Teletex Non Basic Parameters

Table 8.1 Pragmatic Constraints For Selected Attributes. (continued)

Attribute Type	Content	Number of Printable Characters	Number of Allowable Octets	Status	Notes
Telex Number	Telex Number		ffs		CCITT X.500 X.121 Contains telex-number, country code, answer back
X.121 Address	Numeric	15	15		CCITT X.121 X.500
Member	Distinguished Name				Note 3
Owner	Distinguished Name				Note 3
Role Occupant	Distinguished Name				Note 3
See Also	Distinguished Name				Note 3
User Password	Octet String	n/a	128		X.500 Allow long passwords. Machine Generated
Aliased Object Name	Distinguished Name				Note 3
Knowledge Information	T.61 or Printable String	1024	2048		About 1 screen full. ffs
Street Address	T.61 or Reference				UPU. Component of Postal Address

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 implementor's agreements. With 8 NSAP addresses this value is 224.

3. Pragmatic constraints are only applied to the individual components of DistinguishedName 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.

8.12 Conformance

The following sections will describe various aspects of Directory conformance. It should be noted that conformance to the various ASEs and conformance to the Authentication Framework are viewed as separate issues and are presented in that context.

8.12.1 DUA Conformance

Conformance requirements for DUAs are adequately specified in the Directory documents, Part 5, section 9.1. It should be noted that DUA conformance can only be demonstrated by exercising the interface that it provides to its user.

It is recognized that DUAs will be widely differing in nature:

- Some are intended to support human users, some application users
- Particular DUAs may not support particular operations because the application that they support has no requirement; others will be general purpose, and will support all operations.
- Some DUAs will have a fixed view of the Directory content and structure, reflecting the usage of the Directory by a particular application; others will have a more flexible view which can be adapted to new usages.

- Some DUAs will provide automatic referral services with automatic establishment and release of associations; others will place the burden on the user.

- Some DUAs will provide a variety of authentication means; others will support simple authentication only

- Some DUAs will handle operations synchronously; others will have the capability of maintaining several identifiable dialogues with the Directory at one time.

No general implementation agreements are spelled out in respect of these possibilities.

8.12.2 DSA Conformance

Basic conformance requirements for a DSA are defined in the Directory documents, Part 5, section 9.2.

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 must be fully "protocol" conformant to the DAP.

Cooperating

In a distributed directory, responsibility for various portions of the DIT may be "distributed" among multiple DSAs. On a per operation basis we define a DSA to be "holding" when it is responsible for the fragment of the DIB in which a given entry will appear if it exists; we define a DSA to be "propagating" when it is unable to complete the name resolution process. All DSAs must be capable of acting as a "holder" and a"propagator."

8.12.3 Directory Systems Conformance Classes

A DSA implementation shall satisfy the conformance requirements as defined in the Directory documents, part 5, section 9.2, and shall support the "Versions" argument of "Bind".

Additionally, an implementation conformant to these agreements shall state which of the following conformance classes it implements:

Conformance Class 0: Centralized Directory

A DSA implementing conformant to this class only supports the DirectoryAccessAC. It must implement all operations in the Read and Modify ASEs. Optionally, it may implement the operations in the Search ASE.

A DSA that does not support the Search and List operations must reply to such operations with an unwillingToPerform service error.

Conformance Class 1: Distributed Directories

A DSA implementing conformant to this class must implement all the operations in the ASEs that are part of the Application context for which it claims conformance. It must support the DirectoryAccessAC and it may optionally support the DirectorySystemAC.

8.12.4 Authentication Conformance

A Directory System may choose to implement various levels of authentication (Directory documents part 8). We define the following four (4) levels of authentication in the DS:

- No authentication at all; (None)
- Identification of the remote peer entity only, without verification; (Weak)
- Simple authentication: verified identification with or without a password. Intended to make masquerading difficult; and
- 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; we have listed several practical uses of the various levels:

NONE — No authentication may be required for associations with a DSA containing public information: DSAs operating on a private network in a controlled environment may implicitly trust all connections and have no requirement for authentication.

WEAK — 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 authentication may be necessary in situations where strong authentication is not practical, (i.e.: international connections, no knowledge of algorithms in use, etc).

STRONG authentication will be required for secure environments.

8.12.5 Authentication Conformance Classes

We define the following two (2) conformance classes for the support of authentication:

- None, Weak,
- None, Weak, and Simple (unprotected)

Consideration of protected simple and strong authentication is not part of this agreement.

8.13 Distributed Operations

The following requirements apply to DSAs supporting distributed operations:

1. DSAs must support the generation of referrals.

2. DSAs may additionally support chaining. DSAs that only support chaining (i.e. no referrals) are not allowed.

3. DSAs supporting authentication (e.g. simple authentication by name and password) must be able to invoke DSP operations to carry out authentication by reference to other DSAs. Thus all such DSAs must support the DSP protocol.

8.13.1 Referrals and Chaining

It is recommended that a DSA which has chained a request act upon any referrals it receives rather that returning them to the requestor if the "PreferChaining" service control is present.

8.14 Underlying Services

8.14.1 ROSE

It should be noted that support of "abandon" implies support of operation class 2.

8.14.2 Session

A DSA implementation is required to support version 2 of the session protocol, and must also be capable of supporting version 1.

A DUA implementation my use version 1, version 2, or both.

Note: DUAs supporting user authentication may require session version 2, because of the potentially large size of the Bind credentials.

8.15 Access Control

Guidelines relation to access control can be found in annex F of the Directory documents, part 2.

8.16 Authentication

Simple Authentication, as defined in Part 8 of the Directory documents, encompasses both Weak and Simple Authentication, as called for by section 8.12. (This shall be taken to imply unprotected simple authentication.)

A DSA that implements Simple Authentication as called for in section 8.12, will check a user password, (if one is supplied) by means of a compare operation on the user's entry or otherwise; if no user password is supplied, the DSA will validate the presence of an entry for the user, by a read operation or otherwise; the user's entry does not exist. DSP must be used when the user's entry is not a local entry; this form of authentication is therefore not normally appropriate for DAP only DSAs.

A DSA that implements Weak Authentication will accept simple credentials without validating them.

8.17 Test Considerations

This section outlines some items that implementors may wish to consider in terms of testing expectations; additionally, future conformance testers may wish to consider these items when developing tests.

8.17.1 Major elements of Architecture

One important aspect of testing is to confirm the correct behavior of DSAs and DUAs in respect of major elements of the directory architecture.

Such major elements include:

- o Conformance Statement
- o Distinguished names (e.g., name resolution, equivalence of various forms)
- o Entries and Attributes (e.g., accessibility by operations, compliance with rules)
- o Handling of distributed operations (e.g.,naming contexts and knowledge)
- o Schemas
  - Structure rules (e.g., storage and maintenance of structure and of naming)
  - Object classes and sub-classes (e.g., storage and extension of rules for object attributes)
  - Attribute types (e.g., storage and maintenance of syntax classes and rules for multi or single valued attributes)
  - Attribute syntax (e.g. maintenance and support for attribute value testing and matching, to specification for a defined set of attribute types)
- o Operations **
  - all operations
  - correct function
  - correct result
  - correct responses
- o Aliases (e.g., correct resolution, error responses)
- o Authentication and Access Control (e.g., limitation of modify access)
- o ROSE (e.g., correct handling of invokes, results, rejects, and invoke ids)
- o ACSE (e.g., association establishment / refusal for invalid application contexts, etc.)

8.17.2 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.

8.18 Errors

The contents of this section are in the form of a strawman and should be completed prior to the final copy of this document going to press.

- o busy Error - Temporary
- o unwillingToPerform error - Perm.
- o We need an unSupported error that is Perm.
- o Should provide a table indicating perm / temp nature of various errors

This section provides clarification of the semantics of various operation errors and implementation guidelines on their usage.

8.18.1 Permanent vs. Temporary Service Errors

The usage of the Service Errors busy, unavailable and unwillingToPerform requires some clarification:

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 requested would result in the generation of an APDU in excess of 32767 octets.

8.19 APPENDIX A Definitions

Any definitions not found in this appendix can be found in the Directory Documents.

- Holding
- Propagating
- Stand-alone
- Cooperating

8.20 APPENDIX B Attributes and Object Classes

The Contents of this section are incomplete. Additional work describing algorithms related to schemas and special attribute types should be placed here.

The attribute types defined in the Directory documents, Part 6, and listed in tables 8.1 and 8.2 have requirements for underlying algorithms which relate:

1. To the checking of attribute values from the viewpoint of syntactical correctness and compliance with pragmatic constraints.

2. To the comparison of attribute values for the purpose of comparison (for equality, for matching substrings, and as a preliminary for determining relative ordering)

Sections B.2 and B.3 give brief characteristics of the checking and comparison algorithms, respectively. These characteristics are not currently defined explicitly in the Directory documents. Section B.4 summarizes their applicability to the attribute syntaxes defined by the Directory documents.

It should be noted that determining relative ordering requires the application of a further algorithm appropriate to the type of value which requires ordering.

B.1 Checking Algorithms

Note. A particular attribute type in some cases defines more than one alternative attribute syntax. In this case, an attribute value is satisfied if it satisfies at least one checking algorithm, as listed below, maximum and minimum sizes may be specified for the attribute-syntax for the attribute type.

B.1.1 T.61 String Check

Checks that the value has the type code for T.61 string, that it encodes a sequence of valid T.61 characters, and that it complies with a maximum character length, if this is specified for this syntax. Only graphic characters (including spaces) are permitted.

B.1.2 Printable String Check

Checks that the value has the type code for Printable String, and that it encodes as a sequence of valid printable string if this is specified.

B.1.3 Numeric String Check

Checks that the value has the type code for numeric string, that it encodes a sequence of valid numeric string characters, and that it complies with a maximum character length, if this is specified.

B.1.4 Distinguished Name Check

Checks that the value is a valid ASN.1 encoding for Distinguished name, and that for each known attribute type (i.e. that is registered in the DSA) each attribute value satisfies the appropriate checking algorithm.

B.1.5 Object Identifier Check

Checks that the type is correct. It is further study whether the value itself can be validated further.

B.1.6 Criteria Check

Checks that the value is a valid ASN.1 encoding for Criteria. It is for further study whether the value itself can be validated further.

B.1.7 Presentation Address Check

Checks that the value is a valid ASN.1 encoding for Presentation Address, and that each field is within the size limits appropriate to the field.

B.1.8 Telephone Number Check

For further study.

B.1.9 Telex Number Check

For further study.

B.1.10 Facsimile Telephone Number

Checks that the value of the first element is a valid telephone number (B.1.8), and that the second element is a valid ASN.1 encoding for Facsimile G3 Non-Basic Parameters.

B.1.11 Teletex Terminal Identifier

Checks that the value of the first element is a valid Printable String (B.1.2), and that the second element is a valid ASN.1 encoding for Teletex Non-Basic Parameters.

B.1.12 Integer Check

Checks that the value has integer type, and lies between defined or default integer values.

B.1.13 Case Ignore

Checks that each component is compliant using T.61 String Check or Printable String Check.

B.1.14 Preferred Delivery Method

For further study.

B.1.15 Octet String Check

No checking.

B.1.16 Country Check

Checks that the value is a Z-character string complaint with IS 3166 codes only.

B.2 Matching Algorithms

Note: Matching algorithms are conveniently defined in terms of a two step process.

1. Take the checked reference value and the value to be matched, and reduce them to a canonical (i.e. standard) form (normalization), if necessary.

2. Carry out the comparison in the specified way (e.g. equality, substrings or ordering)

Important Note

The brief descriptions below outline the first step. The algorithms may be replaced, in a particular implementation, by equivalent procedures.

B.2.1 String Match

All the specific algorithms below carry out the following basic normalization: remove leading and trailing spaces; intermediate multiple spaces become single spaces. For example "[sp]Time[sp][sp]Flies[sp][sp]" becomes "Time[sp]Flies".

B.2.2 Case Ignore String Match

The basic normalization just described is carried out; in addition each lower case letter is converted to its upper case form. The ASN.1 value is compared octet by octet, disregarding type. It is possible in this way to compare, for example, T.61 strings with Printable Strings.

B.2.3 Case Exact String Match

As above, but without the lower case to upper case conversion.

B.2.4 ASN.1 Match

The ASN.1 is converted to a standardized encoding form: Such as that given in Part 8 of the Directory documents, section 8.6.

Two ASN.1 values may then be compared octet by octet, including the initial ASN.1 type. No matching for substrings or order is possible.

B.2.5 Distinguished Name Match

As for ASN.1 match, except that for each AVA within the distinguished name, normalization takes place (possibly recursively) in accordance with the nature of the attribute type.

B.2.6 Case Ignore String List Match

Each component must match as for Case Ignore String Match.

B.2.7 Criteria Match

For further study.

B.2.8 Telephone Number Match

For further study.

B.2.9 Telex Number Match

For further study.

B.3 Mapping of Attribute Syntaxes

The following table maps the attribute syntaxes of Part 6 to the algorithms described in B.2 and B.3:

Syntax	Check	Match
undefined	None	None
caseExactStringSyntax	T.61 String Check or Printable String Check	Case Exact String Match
caseIgnoreStringSyntax	T.61 String Check or Printable String Check	Case Ignore String Match
caseIgnoreStringListSyntax	String List Check	Case Ignore String List
numericStringSyntax	Numeric String Check	Case Exact String Match
printableStringSyntax	Printable String Check	Case Exact String Match
distinguishedNameSyntax	Distinguished Name Check	Distinguished Name Match
objectIdentifierSyntax	Object Identifier Check	No normalization
criteriaSyntax	Criteria Check	Criteria Match
presentationAddressSyntax	Presentation Address	Presentation Address
telephoneNumberSyntax	Telephone Number Check	Telephone Number Match
telexNumberSyntax	Telex Number Check	Telex Number Match
countrySyntax	Country Check	Case Ignore String Match
integerSyntax	Integer Check	No normalization
octetStringSyntax	Octet String Check	No normalization

8.21 APPENDIX C The Use of ROSE

The use of ROSE by the BIND and UNBIND macros as described in the Directory documents, part 5, clause 8.2 will be used with no additional agreements.

8.22 APPENDIX D Guidelines

The following guidelines were used to provide a general mechanism for arriving at pragmatic constraints within the Directory. These are included for the reader's information.

1. Align with other activities

2. Catch outlandish behavior

    - Infinite Loops
    - Runaway Process

3. Conserve Resources and Promote Efficiency

    - Memory
    - CPU
    - Bandwidth
    - Effort

4. Compromise Arbitrary Opinions

5. Police Actions / Catch Violators

    - Protect the Network

6. Facilitate Interworking

7. Syntax Interpretation

    - Errors

8. Set practical limits for the purpose of testing

8.23 APPENDIX E 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
B-RM	Basic Reference Model
CA	Certification Authority
CCITT	Consultative Committee on International Telegraph & Telephone
CEN	Committee for European Normalization
CENELEC	Committee for European Normalization Electronique
CEPT	Committee European Postal & Telephonique
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	International Organization for Standardization
KT	Knowledge Tree
LL	Lower layers of OSI model (layers 1-4)
MAP	Manufacturing Automation Protocol
MHS	Message Handling Systems
NBS	National Bureau of Standards
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
SIG	Special Interest Group
SPAG	Standards Promotion & Application Group
TOP	Technical and Office Protocols
UL	Upper layers of OSI model (layers 5-7)
UPU	Universal Postal Union

9. SECURITY

The Security Architecture specified in ISO 7498/Part 2 - Security Architecture (as presented in ISO/TC 97/SC 21/N1528) shall be used as a basis for further work in the Special Interest Group on Security.

The security services that are to be implemented first shall include confidentiality, integrity, authentication and access control. Non-repudiation of the source shall also be included for consideration for implementation. These services are defined and discussed in more detail in ISO 7498/Part 2 - Security Architecture.

9.1 Definitions

The following definitions, based on the definitions in ISO 7498/Part2, are to be used when interpreting Chapter 9.

Access Control:	The provision of a security system that establishes and enforces which users or processes can get access to what data or processing facilities.
Authentication Information:	Information used to establish the validity of a claimed identity.
Authorization:	The granting of access rights.
Confidentiality:	A security service that protects data from unauthorized disclosure.
Connection:	A state of communication that exists between two communicating entities by establishing an association between them, providing one or more data paths between them allowing sequential transfers of data, and then terminating the association.
Connectionless:	A state of communication that provides transfer of data from one entity to another without a preestablished association.
Data Integrity:	The property that data has not been altered or destroyed in an unauthorized manner.

Data Origin Authentication:	The corroboration that the source of data received is as claimed.
Digital Signature:	Data that allows a recipient of information to verify the source and integrity of the information.
Peer-entity Authentication:	The corroboration that peer entities in an association are as claimed.
Repudiation:	Denial by one or both of the entities of an association of having participated in all or part of the association or communication of the association.
Selective Field Protection:	The protection of specified fields of data in a communication.
Traffic Analysis:	The inference of information from observation of traffic flow in communications (presence, absence, amount, direction and frequency).
Traffic Flow Confidentiality:	A confidentiality service to protect against traffic analysis.

9.2 Matrix of Security Services and OSI Layers

The following matrix shows the layers of the OSI architecture at which certain security services are considered to be desirable. The entries in the matrix are "H" for high level of desirability, "M" for medium desirability, and "L" for low level of desirability. No entry in the matrix means that the service is not considered desirable. This matrix was produced from a similar matrix in ISO 7498/Part 2 which showed the layers of the architecture that could be used to provide the security service. The level of desirability was established by the members of the Special Interest Group in Security of the OSI Implementors Workshop.

Note: The Matrix is a consensus of the opinions of the members as to where selected security services should be placed. It should not be considered restrictive and interpreted as meaning that the security services cannot be placed elsewhere in the OSI architecture or have other implementation priorities. This will depend upon the differing

security needs of specific applications. Also, it should not be considered complete in that other security services may exist that should be incorporated in the architecture.

Table 9.1 OSI Layers Desirable for Placing Security

SERVICE	1	2	3	4	5	6	7
1.(a) Peer entity authentication			L	H			H
(b) Data origin authentication			L	L			H
2. Access Control Service			Authorization Model				
3.(a) Connection confidentiality	L	L	L	H		H	H
(b) Connectionless confidentiality		L	H	L		H	H
(c) Selective field confidentiality						H	H
(d) Traffic flow confidentiality	M		L				L
4.(a) Connection integrity with recovery				H			L
(b) Connection integrity without recovery			No Plan				
(c) Selective field connection integrity			No Plan				
(d) Connectionless integrity			H	L			L
(e) Selective field connectionless integrity							H
5.(a) Non-repudiation: originator							L
(b) Non-repudiation: receiver							L

Implementation priority: H (high)
M (medium)
L (Low)

Table 1 ISO 7498/Part 2: Security Addendum -- NBS OSI Workshop Summary Of SIG-SEC Discussions of Security Service Placement, May, 1987

Notes: The following notes are for explanation of the above matrix and comments.

A security system should be considered to be an integrated set of security services that are placed at selected OSI layers. The services should be selected based on a risk analysis for the computer system being protected. Security mechanisms must be then chosen that will provide the security services and incorporated in the software and hardware of the computer system and controlled by the OSI software and hardware at the selected layer(s).

For example, authentication, access control, confidentiality and integrity are selected as the major security goals for an OSI system. A connection oriented transport protocol is being implemented. An example of the use of the Matrix could be in an electronic mail system, to illustrate this the following specific services and layers were chosen:

Peer entity authentication: Layer 4
Data origin authentication: Layer 7

Access Control: Layer 7

Connection confidentiality: Layer 4
Selective field confidentiality: Layer 7

Connection integrity with recovery: Layer 4
Connectionless integrity: Layer 7

The layer 7 services were chosen to support the mail system that would protect the selective paragraphs of an electronic message as directed by the user. A mail system is considered connectionless. Access control is a function of only layer 7.

The layer 4 services were chosen to provide a reliable transport service from the sender to the intended receive of the electronic message. A full connection integrity and confidentiality service with peer entity authentication will assure that all information gets to the receiver correctly and confidentially.

Note: The security protocols and mechanisms that provide these services are beyond the scope of this Chapter at this time. The mechanisms and standards for their interoperability are presently being defined and will be added to this Chapter as they become available.

10. OBJECT IDENTIFIER: STRUCTURE AND ALLOCATION

In order to complete a stable version of the NBS OSI Implementation Agreements, the following objects need to be administered by an ad hoc registration authority:

Application Context Name
Abstract Syntax Name
Transfer Syntax Name
Document Type Name
Constraint Set Name
File Model

Since all objects to be administered by the NBS Workshop SIGs are identified by the ASN.1 type OBJECT IDENTIFIER, the following structure shall be used:

Using the NameAndNumberForm (::= identifier (NumberForm)) for an ObjIdComponent we have:

```
ObjectIdentifierValue ::=     { identifier1 (NumberForm1)
       identifier2 (NumberForm2)
       identifier3 (NumberForm3)
       identifier4 (NumberForm4)
       identifier5 (NumberForm5)
       identifier6 (NumberForm6) }
```

The assignment of identifiers and NumberForms is as follows:

identifier1	NumberForm1
iso	1
identifier2	NumberForm2
identified-organization	3
identifier3	NumberForm3
issuing-organization	9999
identifier4	NumberForm4
organization-code	1

identifier5	NumberForm5
application-context	1
abstract-syntax	2
file model	3
constraint-set	4
document-type	5
transfer-syntax	6
ftam-nil-ap-title	7

Note 1: The value of NumberForm3 is selected for use by implementors of these agreements: it has not been assigned by ISO or by any official Registration Authority. It does correspond to an "ad hoc" issuing organization with an ICD of 9999, as specified by ISO 6523. We intend to use the procedure designated in D.7 of the Specification of ASN.1, ISO 8824 once the appropriate Registration Authority has been established. This mechanism is subject to change dependent upon ISO standards.

Note 2: Specific values for identifier6 and NumberForm6 are chosen as needed by the . A table of the currently allocated values is given later.

Note 3: The NBS UL SIG will assign values for identifier5 and NumberForm5 as required by other SIGs.

Note 4: Companies wishing to interoperate may designate themselves with an organization code other than 1 under { iso (1) identified-organization (3) - issuing-organization (9999) } for the purpose of defining private OBJECT IDENTIFIERs.

Table 10.1 TABLE OF ALLOCATED OBJECT IDENTIFIERS

The values of the first 4 NumberForms are constant, so the following value is defined for use in the table below.

```
nbs-ad-hoc OBJECT IDENTIFIER ::= {1 3 9999 1}
```

Note that the only OBJECT IDENTIFIERS herein defined all begin with "nbs-ad-hoc"; all other OBJECT IDENTIFIERS and their associated ObjeceDescriptor's are reproduced here solely for the convenience of implementors. The standards defining these OBJECT IDENTIFIERSs and ObjectDescriptor's take precedence over the values specified below.

Application Context

"ISO FTAM"
{ iso (1) standard (0) 8571 application-context (1) iso-ftam (1) }

Abstract Syntax

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

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

"FTAM unstructured text abstract syntax"

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

"FTAM unstructured binary abstract syntax"

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

"NBS abstract syntax AS1"

{ nbs-ad-hoc abstract-syntax (2) nbs-as1 (1) }

"NBS file directory entry abstract syntax"

{ nbs-ad-hoc abstract-syntax (2) nbs-as2 (2) }

"ISO 8650-ACSE1"

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

"Directory Services"

File Model

```
"FTAM hierarchical file model"
     { iso (1) standard (0) 8571 file-model (3) hierarchical (1) }
```

Constraint Set

```
"FTAM unstructured constraint set"
     { iso (1) standard (0) 8571 constraint-set (4) unstructured (1) }

"FTAM sequential flat constraint set"
     { iso (1) standard (0) 8571 constraint-set (4) sequential-flat
     (2) }
"FTAM ordered flat constraint set"
     { iso (1) standard (0) 8571 constraint-set (4) ordered-flat (3) }

"NBS ordered flat constraint set"
     { nbs-ad-hoc constraint-set (4) nbs_ordered-flat (3) }
```

Document Type

```
"ISO FTAM unstructured text"
     { iso (1) standard (0) 8571 document-type (5) unstructured-text
     (1) }

"ISO FTAM sequential text"
     { iso (1) standard (0) 8571 document-type (5) sequential-text (2)
     }

"ISO FTAM unstructured binary"
     { iso (1) standard (0) 8571 document-type (5) unstructured-binary
     (3) }

"NBS-6 FTAM sequential file"
     { nbs-ad-hoc document-type (5) sequential (6) }

"NBS-7 FTAM random access file"
     { nbs-ad-hoc document-type (5) random-file (7) }

"NBS-8 FTAM indexed file"
     { nbs-ad-hoc document-type (5) indexed-file (8) }

"NBS-9 FTAM file directory file"
     { nbs-ad-hoc document-type (5) file-directory (9) }
```

Transfer Syntax

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

Miscellaneous

"nil AP Title"
{ nbs-ad-hoc ftam-nil-ap-title (7) }

10.1 Specific ASE Requirements for ACSE Presentation and Session

The following list for each ASE the corresponding NBS SIGs requirements of and restrictions on ACSE, Presentation, and Session.

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

All OBJECT IDENTIFIERS are specified in terms of their associated ObjectDescriptor's. See the chapter on OBJECT IDENTIFIERs for the values of the associated OBJECT IDENTIFIERs.

10.1.1 FTAM

10.1.1.1 Phase 2

ACSE Requirements:
all

Application Contexts:
- "ISO FTAM" - implies the use of the ACSE and the FTAM ASE.

Abstract Syntaxes:
- "ISO 8650-ACSE1"

Associated Transfer Syntax:
- "Basic Encoding of a single ASN.1 type"

Presentation Requirements:

Presentation Functional Units:
- kernel

Presentation Contexts:
- At least 4 Presentation Contexts must be supported.

Abstract Syntaxes:

- "FTAM-PCI"

  Associated Transfer Syntax:
  - "Basic Encoding of a single ASN.1 type:

- "FTAM-FADU"

  Associated Transfer Syntax:
  - "Basic Encoding of a single ASN.1 type"

- "FTAM unstructured text abstract syntax"

  Associated Transfer Syntax:
  - "Basic Encoding of a single ASN.1 type"

- "FTAM unstructured binary abstract syntax"

  Associated Transfer Syntax:
  - "Basic Encoding of a single Asn.1 type"

- "NBS abstract syntax AS1"

  Associated Transfer Syntax:
  - "Basic Encoding of a single ASN.1 type"

- "NBS file directory entry abstract syntax"

  Associated Transfer Syntax:
  - "Basic Encoding of a single Asn.1 type"

- "NBS file directory entry abstract syntax"

  Associated Transfer Syntax:
  - "Basic Encoding of a single Asn.1 type"

Session Requirements:

Session Functional Units:

- o kernel
- o duplex

Version Number: 2

Maximum size of User Data parameter field: 10,240

Session Options:

Session Functional Units:

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

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

10.1.2 MHS

10.1.2.1 Phase 1

Session Requirements:

Session Functional Units:

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

Version Number: 1

Maximum size of User Data parameter field: 512

Session Notes:

- Restricted use is made by the RTS of the session services implied by the functional units selected. Specifically,
  - No use is made of S-TOKEN-GIVE, and
  - S-PLEASE-TOKENS only asks for the data token.
- In the S-CONNECT SPDU, the Initial Serial Number should not be present.
- The format of the Connection Identifier in the S-CONNECT SPDU is described in Version 5 of the X.400-Series Implementors' Guide.

10.1.3 DS

10.1.3.1 Phase 1

ACSE Requirements:

all

Application Context:

Abstract Syntaxes:

- "ISO 8650-ACSE1"
  - Associated Transfer Syntax:
    - "Basic Encoding of a single ASN.1 type"
- "Directory Services"
  - Associated Transfer Syntax:
    - "Basic Encoding of a single ASN.1 type"

Presentation Requirements:

Presentation Functional Units:

- kernel

Presentation Contexts:

- At least 2 Presentation Contexts must be supported.

Session Requirements:

Session Functional Units

- kernel
- duplex

Version Number:2

Maximum size of User Data parameter field: 10,240

11. REFERENCES

Selected references are grouped by organization publishing the documents and by Reference Model layer to aid relating standards to the OSI Basic Reference Model and to aid relating equivalent standards published by different standards organizations.

11.1 ISO

Status of ISO work can be determined by the reference number; working drafts are referenced by committee and number; e.g., TC 97/SC 6 Nxxxx. Standards are cited by either ISO xxxx or IS xxxx; DIS and DPs are cited in similar form.

Information Processing Systems - Open Systems Interconnection - Basic Reference Model. ISO/IS 7498. First Edition - Oct. 15, 1984. Ref. No. ISO 7498-1984(E).

OSI Basic Reference Model - Part 2: Security Architecture. ISO/DIS 7498-2 TC 97/SC 21/N1895. Project 97.21.18. May 1987.

OSI Basic Reference Model - Part 3: Naming and Addressing. ISO/DIS 7498-3, ISO/TC 97/ SC 21 N2141. May, 1987.

Data Interchange - Structure for the identification of organizations. ISO 6523. 1984-02-01.

Physical Layer

Information Processing Systems - Local Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Physical Layer Specification ISO/DIS 8802/3

Information Processing Systems - Local Area Networks - Part 4: Token-Passing Bus Access Method and Physical Layer Specification, ISO/DIS 8802/4

Information Processing Systems - Local Area Networks - Part 5: Token-Ring Access Method and Physical Layer Specification ISO/DIS 8802/5

Data Link Layer

Information Processing Systems - Local Area Networks - Part 2: Logical Link Control ISO/DIS 8802/2 1985

Information Processing Systems - Data Communications - High-Level Data Link Control Procedures - Description of the X.25 LAPB-compatible DTE Data Link Procedures, IS 7776.

Network Layer

Information Processing Systems - Open Systems Interconnection - Network Service Definition, IS 8348.

Information Processing Systems - Open Systems Interconnection - Addendum to the Network Service Definition Covering Connectionless Data Transmission, IS 8348/AD1.

Information Processing Systems - Open Systems Interconnection - Addendum to the Network Service Definition Covering Network Layer Addressing, IS 8348/AD2.

Information Processing Systems - Open Systems Interconnection - Internal Organization of the Network Layer, DIS 8648.

Information Processing Systems - Open Systems Interconnection - Protocol for Providing the Connectionless Network Service, IS 8473.

Information Processing Systems - Open Systems Interconnection -Addendumto IS 8473 - Provision of the Underlying Service Assumed by ISO 8473, ISO TC 97/SC 6 N 3453.

Information Processing Systems - Open Systems Interconnection, Working Draft, End System to Intermediate System Routing Exchange Protocol for use in Conjunction with ISO 8473 ISO/DIS 9542 TC 97/SC 6 N 4053.

Information Processing Systems - Open Systems Interconnection - Data Communications - X.25 Packet Level Protocol for Data Terminal Equipment, IS 8208.

Information Processing Systems - Open Systems Interconnection - Data Communications - Use of X.25 to provide the OSI Connection-mode Network Service DIS 8878

Transport Layer

Information Processing Systems - Open Systems Interconnection - Transport Service Definition, IS 8072.

Information Processing Systems - Open Systems Interconnection - Transport Protocol Specification, IS 8072, 1984.

Session Layer

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

Information Processing Systems - Open Systems Interconnection - Basic Connection Oriented Session Protocol Specification, ISO 8326: 1987 (E).

Information Processing Systems - OSI - Basic Oriented Session Service Definition - DAD 2 to ISO 8326 to Incorporate Unlimited User Data, ISO/IS 8326. Aug. 27, 1987.

Note: This Document, SC21 1987, may also be obtained from:

Standards Resource Librarian
AT&T Bell Labs
IM304
Crawfords Corner Rd.
Holmdel, NJ
07733

Information Processing Systems - OSI - Basic Oriented Session Protocol Specification - DAD 2 to ISO 8327 to Incorporate Unlimited User Data, ISO/IS 8327. Aug. 27, 1987.

Presentation Layer

Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Service Definition, ISO 8822: 1987 (E), (ISO/TC97/SC21 N 2335).

Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Protocol Specification, ISO 8823: 1987 (E), (ISO/TC97/SC21 N 2336).

Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), ISO 8824: 1987 (E).

Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation (ASN.1), ISO 8825: 1987 (E).

7-bit Coded Character Set for Information Processing Interchange, ISO 646.

Information Interchange - Representation of Local Time Differentials, ISO 3307.

Application Layer

Information Processing Systems - Open Systems Interconnection - Application Layer Structure, ISO/DP 9545, ISO/TC97/SC21/N1743. July 24, 1987. Revised November 1987.

Application Layer -- **FTAM**

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part I: General Introduction, ISO 8571-1: 1987 (E) (ISO/TC97/SC21 N 2331).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 2: Virtual Filestore Definition, ISO 8571-2: 1987 (E) (ISO/TC97/SC21 N 2332).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 3: The File Service Definition, ISO 8571-3: 1987 (E) (ISO/TC97/SC21 N 2333).

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part 4: File Protocol Specification, ISO 8571-4: 1987 (E) (ISO/TC97/SC21 N 2334).

Application Layer -- **ASE/ACSE**

Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element, ISO 8649: 1987 (E) (ISO/TC97/SC21 N)

Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element, ISO 8650: 1987 (E) (ISO/TC97/SC21 N)

Application Layer -- **VTP**

Information Processing Systems - Open Systems Interconnection - Virtual Terminal Service - Basic Class, IS 9040.

Information Processing Systems - Open Systems Interconnection - Virtual Terminal Protocol - Basic Class, IS 9041.

Application Process -- Office Document Interchange -- **ODA/ODIF**

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 1; General Information, DIS 8613/1.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 2; Document Structures, DIS 8613/2.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 3; Document Processing Reference Model, DIS 8613/3.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 4; Document Profile, DIS 8613/4.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 5; Office Document Interchange Format, DIS 8613/5.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 6; Character Content Architecture, DIS 8613/6.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 7; Raster Graphics Content Architecture, DP 8316/7.

Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 8; Geometric Graphics Content Architecture, DP 8613/8.

Information Processing Systems - Text and Office Systems - Standard Generalized Markup Language (SGML), IS 8879.

Application Process -- Computer Graphics -- CGM/GKS

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 1; Functional Specification, IS 8632/1

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 2; Character Encoding, IS 8632/2

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 3; Binary Encoding, IS 8632/3.

Information Processing Systems - Computer Graphics - Metafile (CGM) for the Storage and Transfer of Picture Description Information, Part 4; Clear Text Encoding, IS 8632/4.

Information Processing Systems - Font and Character Information Interchange, IS 9541.

Information Processing Systems - 8-Bit Single Byte Coded Graphic Character Sets, Part 1; Latin Alphabet Part 1, IS 8859/1.

Information Processing Systems - Computer Graphics Functional Specification of the Graphical Kernel System (GKS), IS 7942.

Information Processing Systems - Computer Graphics - Graphical Kernel System for Three Dimensions (GKS-3D), Functional Description, DIS 8805.

Information Processing Systems - Computer Graphics - Programmers Hierarchical Interactive Graphics System (PHIGS), DP 9592.

Information Processing Systems - Computer Graphics - Interfacing Techniques for Dialogues with Graphical Devices (CGI),ISO TC 97/SC 21 N 1179.

ISO documents may be obtained from:

Frances E. Schrotter
ANSI
ISO TC 97/SC 6 Secretariat
1430 Broadway
New York, NY. 10018

11.2 CCITT

Network Layer

CCITT Recommendation X.25 - 1980, Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks.

CCITT Recommendation X.25 - 1984, Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks.

Transport Layer

CCITT Recommendation X.214, (Red Book, 1984), Transport Service Definition for Open Systems Interconnection for CCITT Applications.

CCITT Recommendation X.224, (Red Book, 1984), Transport Protocol Profile for Open Systems Interconnection for CCITT Applications.

Session Layer

CCITT Recommendation X.215, (Red Book, 1984), Session Service Definition for Open Systems Interconnection for CCITT Applications.

CCITT Recommendation X.225, (Red Book, 1984), Session Protocol Profile for Open Systems Interconnection for CCITT Applications.

Presentation

CCITT Recommendation T.50, (Red Book, 1984), International Alphabet No. 5.

Application Layer -- MHS

CCITT Recommendation X.400, (Red Book, 1984), Message Handling Systems: System Model-Service Elements.

CCITT Recommendation X.401, (Red Book, 1984), Message Handling Systems: Basic Service Elements and Optional User Facilities.

CCITT Recommendation X.408, (Red Book, 1984), Message Handling Systems: Encoded Information Type Conversion Rules.

CCITT Recommendation X.409, (Red Book, 1984), Message Handling Systems: Presentation Transfer Syntax and Notation.

CCITT Recommendation X.410, (Red Book, 1984), Message Handling Systems: Remote Operations and Reliable Transfer Server.

CCITT Recommendation X.411, (Red Book, 1984), Message Handling Systems: Message Transfer Layer.

CCITT Recommendation X.420, (Red Book, 1984), Message Handling Systems: Interpersonal Messaging User Agent Layer.

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

Application Layer -- Directory Services

The Directory--Overview of Concepts, Models, and Services (CCITT Recommendation X.500, ISO 9594)

The Directory--Information Framework (CCITT Recommendation X.501, ISO 9594)

The Directory--Access and System Services Definition (CCITT Recommendation X.511, ISO 9594)

The Directory--Procedures For Distributed Operation (CCITT Recommendation X.518, ISO 9594)

The Directory--Access and System Protocols Specification (CCITT Recommendation X.519, ISO 9594)

The Directory--Selected Attribute Types (CCITT Recommendation X.520, ISO 9594)

The Directory--Selected Object Classes (CCITT Recommendation X.521, ISO 9594)

The Directory--Authentication Framework (CCITT Recommendation X.509, ISO 9594)

Remote Operations-Part 1: Model, Notation and Service Definition (CCITT Recommendation X.219, ISO 9072 Version 5)

Remote Operations-Part 2: Protocol Specification (CCITT Recommendation X.229, ISO 9072 Version 5)

Association Control-Service Definition (CCITT Recommendation X217, ISO 8649)

Association Control-Protocol Definition (CCITT Recommendation X.217, ISO 8650)

CCITT documents may be obtained from:

International Telecommunications Union
Place des Nations, CH 1211,
Geneva 20 SWITZERLAND

11.3 IEEE

Physical Layer

IEEE Standard for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Physical Layer Specification, ANSI/IEEE Standard 802.3 1985, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

IEEE Standard for Local Area Networks: Token-Passing Bus Access Method and Physical Layer Specification, ANSI/IEEE Standard 802.4 - 1985, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

IEEE Standard for Local Area Networks: Token-Ring Access Method, ANSI/IEEE Standard 802.5-1985, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

Data Link Layer

IEEE Standard for Local Area Networks: Logical Link Control, ANSI/IEEE Standard 802.2 - 1985, Institute of Electrical and Electronics Engineers, 345 East 47th St., New York, NY. 10017, 1985.

11.4 NBS

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

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

Implementation Agreements for Open Systems Interconnection Protocols: NBS Workshop for Implementors of Open Systems Interconnection, National Bureau of Standards, NBSIR 86-3385-6, Robert Rosenthal, Editor, Revised July 1987.

Implementation Agreements Among Participants of OSINET, National Bureau of Standards, Institute for Computer Sciences and Technology, NBSIR 86-3478, 1987.

U. S. Government Open Systems Interconnection Profile (GOSIP), National Bureau of Standards, Institute for Computer Sciences and Technology, 1987.

NBS documents may be obtained from:

NTIS
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA. 22161.
or
National Bureau of Standards
Institute for Computer Sciences and Technology
Gaithersburg, MD. 20899

11.5 MAP

Manufacturing Automation Protocol, General Motors Corporation,Manufacturing Engineering and Development, Advanced Product and Manufacturing Engineering Staff (APMES), APMES A/MD-39, GM Technical Center, Warren, MI. 48090-9040.

11.6 TOP

Technical and Office Protocols Specification Version 3.0, MAP/TOP Users Group, Attention TOP 3.0 Document. One SME Drive, P.O. Box 930, Dearborn Mi. 48121.

11.7 CEN/CENELEC

FTAM Draft Functional Standard A/111 (prENV 41 204) Simple File Transfer-Unstructrued, 1987-12-01.

ENV 41201 "Private Message Handling Systems"

ENV 41202 "Public Message Handling Systems"

11.8 SPAG

FTAM Draft Profile A/112 Positional File Transfer - Flat, 1987-120-7.

FTAM Draft Profile A/122 Positional File Access - Flat, 1987-12-07.

FTAM Draft Profile A/13 File Management, 1987-12-07.

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** *(See instructions)*	1. PUBLICATION OR REPORT NO. NBS/SP-500/150	2. Performing Organ. Report No.	3. Publication Date January 1988

4. TITLE AND SUBTITLE

COMPUTER SCIENCE AND TECHNOLOGY:
Stable Implementation Agreements for Open Systems Interconnection Protocols

5. AUTHOR(S)

Robert Rosenthal, Editor

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

SAME AS ITEM #6 ABOVE.

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card No. 88-600500

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

This document records current Stable Agreements for Open Systems Interconnection Protocols among the organizations participating in the NBS/OSI Workshop Series for Implementors of OSI Protocols.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

NBS/OSI Workshop; network protocols; open systems interconnection; OSINET; testing protocols

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

257

15. Price

*U.S. GOVERNMENT PRINTING OFFICE:1988-199-781

USCOMM-DC 6043-P80

READER RESPONSE FORM

Please add/retain my name for the next mailing of the NBS Workshop for Implementors of Open Systems Interconnection.

NAME _____

ADDRESS _____

_____

_____

PHONE NO. _____

Mail this page to:

Lawrence Keys
National Bureau of Standards
Bldg. 225/B-217
Gaithersburg, MD 20899

NBS Technical Publications

Periodical

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).
NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.
Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*
Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300