NISTIR 90-4247

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held December 15, 1989
National Institute of Standards and
Technology
Gaithersburg, MD 20899

Tim Boland, Editor

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

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
Lee Mercer, Deputy Under Secretary
for Technology
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

NISTIR 90-4247

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held December 15, 1989
National Institute of Standards and
Technology
Gaithersburg, MD 20899

Tim Boland, Editor

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

February 1990

Issued March 1990

Table of Contents

1. GENERAL INFORMATION 1
 1.1 PURPOSE OF THIS DOCUMENT 1
 1.2 PURPOSE OF THE WORKSHOP 2
 1.3 WORKSHOP ORGANIZATION 2
 1.4 USE AND ENDORSEMENT BY OTHER ENTERPRISES 2
 1.5 RELATIONSHIP OF THE WORKSHOP TO THE NIST LABORATORIES . . . 3
 1.6 STRUCTURE AND OPERATION OF THE WORKSHOP 3
  1.6.1 Plenary 3
  1.6.2 Special Interest Groups 3
 1.7 POINTS OF CONTACT 13

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 1
  2.5.3 IEEE 802.4 Token Bus Access Method 2
  2.5.4 IEEE 802.5 Token Ring Access Method 2
  2.5.5 Fiber Distributed Data Interface (FDDI) 2
   2.5.5.1 Token Ring Media Access Control (MAC, X3.139-1987) 2
   2.5.5.2 Token Ring Physical Level (PHY,X3.148-1988) . 3
   2.5.5.3 Physical Layer Media Dependent (PMD, X3.166-198X) 4
 2.6 X.25 WIDE AREA NETWORKS 4
  2.6.1 Introduction 4
  2.6.2 ISO 7776 4
  2.6.3 ISO 8208 4
 2.7 INTEGRATED SERVICES DIGITAL NETWORKS (ISDN) 4
  2.7.1 Introduction 4
  2.7.2 Implementation Agreements 4
   2.7.2.1 Physical Layer, Basic Access at "U" 5
   2.7.2.2 Physical Layer, Basic Access at S and T . . . 5
   2.7.2.3 Physical Layer, Primary Rate at "U" 5
   2.7.2.4 Data Link Layer, D-Channel 5
   2.7.2.5 Signaling 5
   2.7.2.6 Data Link Layer B-Channel 5
   2.7.2.7 Packet Layer 5
   2.8 APPENDIX A 5
  2.8.1 Data Link Layer, D-Channel 5
  2.8.2 Signaling 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 ISO 8473 1
3.5.2 Provision of CLNS over Local Area Networks 3
3.5.3 Provision of CLNS over X.25 Subnetworks 3
3.5.4 Provision of CLNS over ISDN 3
3.5.4.1 CLNP Utilizing X.25 Services 3
3.5.5 Provision of CLNS over Point-to-Point Links . . . 3
3.6 CONNECTION-MODE NETWORK SERVICE 3
3.6.1 Mandatory Method of Providing CONS 3
3.6.1.1 General 3
3.6.1.2 X.25 WAN 4
3.6.1.3 LANs 4
3.6.1.4 ISDN 4
3.6.1.5 PRIORITY 4
3.6.2 Additional Option: Provision of CONS over X.25 1980 Subnetworks 4
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
3.6.4 Interworking 4
3.7 ADDRESSING 4
3.8 ROUTING 5
3.8.1 End System to Intermediate System Routing 5
3.8.2 Intermediate Systems to Intermediate Systems Routing 5
3.9 PROCEDURES FOR OSI NETWORK SERVICE/PROTOCOL IDENTIFICATION . 5
3.9.1 General 5
3.9.2 Processing of Protocol Identifiers 6
3.9.2.1 Originating NPDUs 6
3.9.2.2 Destination System Processing 6
3.9.2.3 Further Processing in Originating End System . 6
3.9.3 Applicable Protocol Identifiers 6
3.10 MIGRATION CONSIDERATIONS 6
3.10.1 X.25-1980 6
3.11 USE OF PRIORITY 6
3.11.1 Introduction .6
3.11.2 Overview 7
3.12 CONFORMANCE 8
3.13 APPENDIX A 8
3.13.1 Problem Statement 8
3.13.2 Address Notational Considerations 9
3.13.3 Requirement to Use Functional Addressing 10
3.13.4 Proposal to Revise Agreements 11

4. TRANSPORT LAYER 1
4.1 INTRODUCTION 1
4.2 SCOPE AND FIELD OF APPLICATION 1
4.3 STATUS 1
4.4 ERRATA 1
4.4.1 ISO/CCITT Defect Reports 1
4.5 PROVISION OF CONNECTION MODE TRANSPORT SERVICES 1
4.5.1 Transport Class 4 1

4.5.1.1 Transport Class 4 Overview 1
4.5.1.2 Protocol Agreements 1
4.5.1.2.1 General Rules 2
4.5.1.2.2 Transport Class 4 Service Access Points or Selectors 2
4.5.1.2.3 Retransmission Timer 2
4.5.1.2.4 Keep-Alive Function 2
4.5.1.2.5 Congestion Avoidance Policies 2
4.5.1.2.6 Use of Priority 3
4.5.2 Transport Class 0 4
4.5.2.1 Transport Class 0 Overview 5
4.5.2.2 Protocol Agreements 5
4.5.2.2.1 Transport Class 0 Service Access Points 5
4.5.2.3 Rules for Negotiation 5
4.5.3 Transport Class 2 5
4.5.3.1 Transport Class 2 Overview 5
4.5.3.2 Protocol Agreements 5
4.6 PROVISION OF CONNECTIONLESS TRANSPORT SERVICE 5
4.7 TRANSPORT PROTOCOL IDENTIFICATION 5

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 1
5.4.1 ISO Defect Reports 1
5.4.2 Session Defects 1
5.5 ASSOCIATION CONTROL SERVICE ELEMENT 1
5.5.1 Introduction 1
5.5.2 Services 2
5.5.3 Protocol Agreements 2
5.5.3.1 Application Context 2
5.5.3.2 AE Title 2
5.5.4 ASN.1 Encoding Rules 2
5.5.5 Connectionless 2
5.5.6 Result Parameter 2
5.6 ROSE 2
5.7 RTSE 2
5.8 PRESENTATION 2
5.8.1 Introduction 3
5.8.2 Service 3
5.8.3 Protocol Agreements 3
5.8.3.1 Transfer Syntaxes 3
5.8.3.2 Presentation Context Identifier 3
5.8.3.3 Default Context 3
5.8.3.4 P-Selectors 3
5.8.3.5 Provider Abort Parameters 3
5.8.3.6 Provider Aborts and Session Version 3
5.8.3.7 CPC-Type. 3
5.8.3.8 Presentation-context-definition-result-list . 4
5.8.3.9 RS-PPDU 4

5.8.4 Presentation ASN.1 Encoding Rules 4
5.8.4.1 Invalid Encoding. 4
5.8.5 General 4
5.8.5.1 Presentation Data Value (PDV) 4
5.8.6 Connection Oriented 4
5.8.7 Connectionless 4
5.9 SESSION 4
5.9.1 Introduction 4
5.9.2 Services 4
5.9.3 Protocol Agreements 5
5.9.3.1 Concatenation 5
5.9.3.2 Segmenting. 5
5.9.3.3 Reuse of Transport Connection 5
5.9.3.4 Use of Transport Expedited Data 5
5.9.3.5 Use of Session Version Number 5
5.9.3.6 Receipt of Invalid SPDUs. 5
5.9.3.7 Invalid SPM Intersections 5
5.9.3.8 S-Selectors 5
5.9.4 Connectionless 5
5.10 UNIVERSAL ASN.1 ENCODING RULES 5
5.10.1 TAGS 6
5.10.2 Definite Length 6
5.10.3 External 6
5.10.4 Integer 6
5.10.5 String Types 6
5.10.6 Bit String 6
5.11 CHARACTER SETS 7
5.11.0 Introduction 7
5.11.1 Scope. 7
5.11.1.1 Recording Additional Character Sets 7
5.11.1.2 General Applicability of Character Sets . . . 7
5.11.1.3 Minimum Number of Character Sets. 7
5.11.2 References 7
5.11.3 Definitions . . . .
5.11.3.1 character data: 7
5.11.3.2 composite graphic symbol. 8
5.11.4 Abbreviations. 8
5.11.5 Position within the Taxonomy 8
5.11.6 Conformance. 8
5.11.6.1 Processed Character Data. 8
5.11.6.1.1 Non-supported Character Sets. . . . 8
5.11.6.1.2 Reserved Character Codes. 8
5.11.6.1.3 Validation of Character Codes . . . 8
5.11.6.2 Unprocessed Character Data. 8
5.11.6.2.1 Validation of Character Codes. . . . 8
5.11.7 General Agreements 9
5.11.7.1 Encoding. 9
5.11.7.1.1 Overprint, Composite Characters . . 9
5.11.7.1.2 Code Extension Facilities for General String and GraphicString. 9
5.11.7.1.3 Initial Conditions for TeletexString 9
5.11.7.2 Comparisons 9

5.11.7.2.1 Matching Characters 9
5.11.7.2.2 Caseignore Comparisons. 10
5.11.7.2.3 Ordering and Comparing Characters . 10
5.11.7.2.4 Comparing Encoded ASN.1 Character Strings 10
5.11.8 Character Set Profiles 10
5.11.8.1 CSI-1 ISO 646 Graphic Character Set 11
5.11.8.1.1 Base Standard 11
5.11.8.1.2 Subrepertoire or Version. 11
5.11.8.1.3 Standard Options. 11
5.11.8.1.4 Character Set Components and Designated Position 11
5.11.8.1.5 Other Agreements. 11
5.11.8.2 CS1-2 JIS X0208 11
5.11.8.3 CSI-3 CCITT Recommendation. 11
5.11.8.3.1 Base Standard 11
5.11.8.3.2 Subrepertoire or version. 11
5.11.8.3.3 Standard Options Selected 11
5.11.8.3.4 Character Set Components and Designated Position 11
5.11.8.3.5 Other Agreements. 11
5.11.8.4 CSI-4 ISO 8859-1 Latin Alphabet No. 1 11
5.11.8.4.1 Base Standard 11
5.11.8.4.2 Subrepertoire or Version. 11
5.11.8.4.3 Standard Options Selected 11
5.11.8.4.4 Character Set Components and Designated Postion. 11
5.11.8.4.5 Other Agreements. 12
5.11.8.5 CS1-5 ISO 6937-2 Coded Character Sets for Text Communication. 12
5.11.8.5.1 Base Standard. 12
5.11.8.5.2 Subrepertoire or Version . . . 12
5.11.8.5.3 Standard Options Selected. . . 12
5.11.8.5.4 Character Set Components and Designated Postion 12
5.11.8.5.5 Other Agreements 12
5.11.8.5.6 CSI-6 ISO 8859/7 Greek Supplementary Set. 12
5.11.8.5.7 CSI-7 CCITT Recommendation . . 12
5.11.8.7.1 Base Standard 12
5.11.8.7.2 Subrepertoire or Version. 12
5.11.8.7.3 Standard Options Selected 12
5.11.8.7.4 Character Set Components and Designated Position. 12
5.11.8.7.5 Other Agreements. 12
5.11.8.8 CS 1-8 CCITT Recommendation T.61 Graphic Character Sets 12
5.11.8.9 Korean Notional Character Set. 12
5.11.8.10 CS2-1 ISO CO Control Functions 13
5.11.8.10.1 Base Standard. 13
5.11.8.10.2 Subrepertoire or Version . . . 13
5.11.8.10.3 Standard Options Selected. . . 13

5.11.8.10.4 Character Set Components and Designated Position. 13
5.11.8.10.5 Other Agreements 13
5.11.8.11 CS2-2 ISO 6429 Additional Control Functions. 13
5.11.8.11.1 Base Standard. 13
5.11.8.11.2 Subrepertoire or Version . . . 13
5.11.8.11.3 Standard Options Selected. . . 13
5.11.8.11.4 Character Set Components and Designated Position. 13
5.11.8.11.5 Other Agreements 13
5.11.8.12 CS2-3 CCITT. 13
5.11.8.12.1 Base Standard. 13
5.11.8.12.2 Subrepertoire or Version . . . 13
5.11.8.12.3 Standard Options Selected. . . 13
5.11.8.12.4 Character Set Components and Designated Position. . . . 13
5.11.8.12.5 Other Agreements 13
5.11.8.13 CS2-4 CCITT. 13
5.11.8.13.1 Base Standard. 13
5.11.8.13.2 Subrepertoire or Version . . . 13
5.11.8.13.3 Standard Options Selected. . . 13
5.11.8.13.4 Character Set Components and Designated Position. 13
5.11.8.13.5 Other Agreements 14
5.11.9 Annex Character Set Technology. 15
5.11.9.1 Introduction 15
5.11.9.2 Scope. 15
5.11.9.3 Field of Application 15
5.11.9.4 Character Set Standards. 15
5.11.9.5 Introduction to Character Set Standards. 15
5.11.9.6 Definitions. 15
5.11.9.7 ISO 2022 Information Processing - ISO 7 bit and 8-bit coded character sets - Code extension techniques 16
5.11.9.7.1 Structure of a Family of 8-bit codes. 16
5.11.9.7.2 Elements of Code Extension in an 8-bit Environment 18
5.11.9.7.3 Multiple Character Sets. . . . 18
5.11.9.7.4 Announcement of Extension Facilities 21
5.11.9.7.5 Composite Graphic Characters . 21
5.11.9.7.6 International Register of Coded Character Sets to be used with Escape Sequences . . 21
5.11.9.8 Character Sets 22
5.11.9.8.1 ISO 646 7-bit coded character set for information processing interchange and CCITT Recommendation T.50 International Alphabet No. 5 22

5.11.9.8.2 ISO 8859 Information Processing - 8 bit single-byte coded character sets 23
5.11.9.8.3 ISO 6937 Information Processing - Coded Character Sets for Text Communication. 24
5.11.9.8.3.1 ISO 6937/1 Information Processing - Coded Character Sets for Text Communication - Part 1: General . . . . . Introduction. 24
5.11.9.8.3.2 ISO 6937/2 Information Processing - Coded Character Sets for Text Communication - Part 2: Latin Alphabetic and Non-alphabetic Graphic Characters. 25
5.11.9.8.4 CCITT Recommendation T.51 Coded Character Sets for Telematic Services 26
5.11.9.8.5 CCITT Recommendation T.61 Character Repertoire and Coded Character Sets for the International Teletex Service . 26
5.11.9.9 ASN.1 Character String Types 26
5.11.9.9.1 Universal Class Numbers and Registration Numbers 26
5.11.9.9.2 Initial States 26
5.11.9.10 Use of ASN.1 OctetSTring as a Character String 27
5.11.9.11 Escape Sequences for Character Set Designation. 27
5.12 CONFORMANCE. 28
5.12.1 Specific ASE Requirements 28
5.12.1.1 FTAM 28
5.12.1.2 MHS 28
5.12.1.2.1 Phase 1 (1984 X.400) 28
5.12.1.2.2 Phase 2, Protocol P1 (1988 X.400) . 28
5.12.1.2.3 Phase 2, Protocol P7 (1988 X.400) . 28
5.12.1.2.4 Phase 2, Protocol P3 (1988 X.400) . 28
5.12.1.3 DS 28
5.12.1.3.1 Phase 1 28
5.12.1.4 Virtual Terminal 29
5.12.1.4.1 Phase 1a 29
5.12.1.4.2 Phase 1b 29
5.13 APPENDIX A: RECOMMENDED PRACTICES 30
5.14 APPENDIX B: OBJECT IDENTIFIER REGISTER. 31
5.14.1 Register Indes 31
5.14.2 Object Identifier Descriptions 31

6. OBJECT IDENTIFIERS AND OTHER REGISTRATION ISSUES 1

7. STABLE MESSAGE HANDLING SYSTEMS 1

8. MESSAGE HANDLING SYSTEMS 1
8.1 INTRODUCTION 1
8.2 SCOPE 1
8.3 STATUS 1
8.4 ERRATA 1
8.5 MT KERNEL 1
8.5.1 Introduction 1
8.5.2 Elements of Service 1
8.5.3 MTS Transfer Protocol (P1) 1
8.5.4 MTS - APDU Size 1
8.5.5 1988/84 Interworking Considerations 4
8.6 IPM KERNEL 2
8.6.1 Introduction 4
8.7 MESSAGE STORE 4
8.7.1 Introduction 4
8.7.2 Scope 4
8.7.3 Elements of Service 5
8.7.4 Attribute Types 5
8.7.5 Pragmatic Constraints for Attribute Types 5
8.7.6 Implementation of the MS with 1984 Systems 5
8.7.7 MS Access Protocol (P7) 5
8.7.8 MTS Access Protocol (P3) 5
8.8 REMOTE USER AGENT SUPPORT 5
8.8.1 Introduction 5
8.8.2 Scope 5
8.8.3 Elements of Service 6
8.8.4 MTS Access Protocol (P3) 6
8.9 NAMING, ADDRESSING & ROUTING 6
8.9.1 Use of O/R Addresses for Routing 6
8.9.2 Distribution Lists 6
8.9.2.1 Introduction 6
8.9.2.2 Elements of Service 6
8.9.3 MHS Use of Directory 7
8.9.3.1 Introduction 7
8.9.3.2 Functional Configuration 7
8.9.3.3 Functionality 7
8.9.3.4 Naming and Attributes 8
8.9.3.5 Elements of Service 10
8.10 MHS MANAGEMENT 10
8.11 MHS SECURITY 10
8.11.1 Introduction 10
8.11.2 Elements of Service 10
8.12 SPECIALIZED ACCESS 12
8.12.1 Physical Delivery 12
8.12.1.1 Introduction 12
8.12.1.2 Elements of Service 12
8.12.2 Other Access Units 14
8.12.2.1 Facsimile Access Units 14
8.12.2.2 Telex Access Units 15
8.12.2.3 Teletex Access Units 15

8.13 CONVERSION 15
8.13.1 Introduction 15
8.13.2 Elements of Service 15
8.14 USE OF UNDERLYING LAYERS 16
8.14.1 MTS Transfer Protocol (P1) 16
8.14.2 MTS Access Protocol (P3) and MS Access Protocol (P7) 16
8.15 ERROR HANDLING 16
8.15.1 PDU Encoding 16
8.15.2 Contents 16
8.15.3 Envelope 16
8.15.4 Reports 16
8.15.5 Pragmatic Constraints 16
8.16 CONFORMANCE 17
8.17 APPENDIX A: MHS PROTOCOL SPECIFICATIONS 17
8.17.1 MTS Transfer Protocol (P1) 17
8.17.2 Interpersonal Massaging Protocol (P2) 17
8.17.3 MTS Access Protocol (P3) 17
8.17.4 MS Access Protocol (P7) 17
8.17.5 Message Store General Attribute Support 17
8.17.6 Message Store IPM Attribute Support 17
8.18 APPENDIX B: INTERPRETATION OF ELEMENTS OF SERVICE 18
8.19 APPENDIX C: RECOMMENDED PRACTICES 19
8.19.1 Printable String 19
8.19.2 Rendition of IA5Text 19
8.19.3 EDI 19
8.19.3.1 Introduction and Scope 19
8.19.3.2 Model 19
8.19.3.3 Protocol Elements Supported for EDI 19
8.19.3.4 Addressing and Routing 19
8.20 APPENDIX D: LIST OF ASN.1 OBJECT IDENTIFIERS 20
8.20.1 Content Types 20
8.20.2 Body Part Types 20

9. STABLE FTAM PHASE 2 1

10. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 3 1

11. DIRECTORIES 1
11.1 INTRODUCTION 1
11.2 SCOPE AND FIELD OF APPLICATION 1
11.3 STATUS 1
11.4 USE OF DIRECTORIES 1
11.5 DIRECTORY ASES AND APPLICATION CONTEXTS. 1
11.6 SCHEMA 1
11.6.1 Support of Structure and Naming Rules 1
11.6.2 Support of Object Classes and Subclasses 1
11.6.3 Support of Attribute types 2
11.6.4 Naming Contexts. 2
11.6.5 Common Profiles 2
11.6.6 Naming Contexts 2

11.6.6.1 OIW Directory Common Application Directory Profile 2
11.6.6.1.1 Standard Application Specific Attributes and Attribute Sets . . . 2
11.6.6.1.2 Standard Application Specific Object Classes. 2

11.6.6.2 OIW Directory Strong Authentication Directory Profile 2
11.6.6.2.1 Other Profiles Supported. 2
11.6.6.2.2 Standard Application Specific Object Classes. 2
11.6.7 Restrictions on Object Class Definitions 3
11.7 PRAGMATIC CONSTRAINTS. 3
11.7.1 General Constraints. 3
11.7.1.1 Character Sets. 3
11.7.1.2 APDU Size Considerations. 3
11.7.1.3 Service Control (SC) Considerations 3
11.7.1.4 Priority Service Control. 3
11.7.2 Constraints on Operations. 3
11.7.2.1 Filters 3
11.7.2.2 Errors. 3
11.7.2.3 Error Reporting - Detection of Search Loop. . 4
11.7.3 Constraints Relevant to Specific Attribute Types . 4
11.8 GENERAL CONSTRAINTS 4
11.8.1 DUA Conformance. 4
11.8.2 DSA Conformance. 4
11.8.3 DSA Conformance Classes. 4
11.8.4 Authentication Conformance 4
11.8.5 Directory Service Conformance. 4
11.8.6 The Directory Access Profile 4
11.8.7 The Directory System Profile 4
11.8.8 Digital Signature Protocol Conformance Profile . . 5
11.8.9 Strong Authentication Protocol Conformance Profile 5
11.9 DISTRIBUTED OPERATIONS 5
11.9.1 Referrals and Chaining 5
11.9.2 Trace Information. 5
11.10 UNDERLYING SERVICES 5
11.10.1 Rose 5
11.10.2 Session. 5
11.10.3 ACSE 5
11.11 ACCESS CONTROL. 5
11.12 TEST CONSIDERATIONS 6
11.12.1 Major Elements of Architecture 6
11.12.2 Search Operations 6
11.13 ERRORS 6
11.13.1 Permanent vs. Temporary Service Errors 6
11.13.2 Guidelines for Error Handling. 6
11.13.2.1 Introduction. 6
11.13.2.2 Symptoms. 6
11.13.2.3 Situations. 6
11.13.2.4 Error Actions 6

11.13.2.5 Reporting 7
11.14 SPECIFIC AUTHENTICATION SCHEMES 7
11.14.1 Specific Strong Authentication Schemes 7
11.14.1.1 ElGamal 7
11.14.1.1.1 References 7
11.14.1.1.2 Background. 7
11.14.1.1.3 Digital Signature 7
11.14.1.1.4 Verification. 7
11.14.1.1.5 Known Constraints on Parameters . . 7
11.14.1.1.6 Note on subjectPublicKey. 7
11.14.1.2 One-Way Hash Functions. 8
11.14.1.2.1 SQUARE-MOD-N Algorithm. 8
11.14.1.2.2 MD2 Algorithm 8
11.14.1.2.3 Study of Other One-Way Hash Functions 8
11.14.1.2.4 Use of One-Way Hash Functions in Forming Signatures. 8
11.14.1.3 ASN.1 for Strong Authentication Algorithms. . 8
11.14.2 Protected Simple Auothentication 8
11.14.3 Simple Authentication. 8
11.15 APPENDIX A: MAINTENANCE OF ATTRIBUTE SYNTAXES 9
11.15.1 Introduction 9
11.15.2 General Rules. 9
11.15.3 Checking Algorithms. 9
11.15.3.1 distinguishedNameSyntax 9
11.15.3.2 integerSyntax 9
11.15.3.3 telephoneNumberSyntax 9
11.15.3.4 countryName 9
11.15.3.5 preferredDeliveryMethod 9
11.15.3.6 presentationAddress 9
11.15.4 Matching Algorithms. 10
11.15.4.1 UTCTimeSyntax 10
11.15.4.2 distinguishedNameSyntax 10
11.15.4.3 caseIgnoreListSyntax. 10
11.16 APPENDIX B: GLOSSARY. 10
11.17 APPENDIX C: REQUIREMENTS FOR DISTRIBUTED OPERATIONS. . 10
11.17.1 General Requirements 10
11.17.2 Protocol Support 10
11.17.2.1 Usage of Chaining Arguments 10
11.17.2.2 Usage for ChainingResults. 10
11.18 APPENDIX D: GUIDELINE FOR APPLICATIONS USING THE DIRECTORY 11
11.18.1 Tutorial 11
11.18.1.1 Overview. 11
11.18.1.2 Use of the Directory Schema 11
11.18.1.2.1 Use of Existing Object Classes. . . 11
11.18.1.2.2 Kinds of Object Classes 11
11.18.1.2.3 Use of Unregistered Object Classes. 11
11.18.1.2.4 Side Effects of Creating Unregistered Object Classes 11
11.18.2 Creation of New Object Classes. 11
11.18.2.1 Creation of New Subclasses. 11

11.18.2.2 Creation of New Attributes. 12
11.18.3 DIT Structure Rules. 12
11.19 APPENDIX E: TEMPLATE FOR AN APPLICATION SPECIFIC PROFILE FOR USE OF THE DIRECTORY. 12

12. STABLE SECURITY AGREEMENTS 1

13. SECURITY 1
13.1 INTRODUCTION 1
13.1.1 References 1
13.1.2 Assumptions 1
13.1.3 Definitions 1
13.1.4 Motivation 1
13.1.5 Security Chapter Structure 1
13.2 SCOPE AND FIELD OF APPLICATION 1
13.3 STATUS 1
13.4 ERRATA 1
13.5 GENERAL OSI SECURITY MODEL 1
13.5.1 General Matrix from 7498-2 1
13.5.2 Selected Matrix of Services/Layers 1
13.5.3 Security Domain Model 1
13.6 OSI MANAGEMENT SECURITY AND SECURITY MANAGEMENT 1
13.7 PHYSICAL LAYER 1
13.7.1 Introduction 1
13.7.1.1 References 1
13.7.1.2 Definitions 1
13.7.1.3 Assumptions 1
13.7.1.4 Motivation 1
13.7.2 Scope and Field of Application 1
13.7.3 Specific Security Model 1
13.7.4 Services Offered 1
13.7.5 Services Required 2
13.7.6 Protocols 2
13.7.7 Management Elements Required/Impacted 2
13.7.8 Conformance Class Definitions 2
13.7.9 Conformance Class Specifications 2
13.7.10 Registration Issues Requirements 2
13.8 DATA-LINK LAYER 2
13.8.1 Introduction 2
13.8.1.1 References 2
13.8.1.2 Definitions 2
13.8.1.3 Assumptions 2
13.8.1.4 Motivation 2
13.8.2 Scope and Field of Application 2
13.8.3 Specific Security Model 2
13.8.4 Services Offered 2
13.8.5 Services Required 2
13.8.6 Protocols 2
13.8.7 Management Elements Required/Impacted 2
13.8.8 Conformance Class Definitions 2
13.8.9 Conformance Class Specifications 2
13.8.10 Registration Issues Requirements 2

13.9 NETWORK LAYER 2
13.9.1 Introduction 2
13.9.1.1 References 3
13.9.1.2 Definitions 3
13.9.1.3 Assumptions 3
13.9.1.4 Motivation 3
13.9.2 Scope and Field of Application 3
13.9.3 Specific Security Model 3
13.9.4 Services Offered 3
13.9.5 Services Required 3
13.9.6 Protocols 3
13.9.7 Management Elements Required/Impacted 3
13.9.8 Conformance Class Definitions 3
13.9.9 Conformance Class Specifications 3
13.10 TRANSPORT LAYER 3
13.10.1 Introduction 3
13.10.1.1 References 3
13.10.1.2 Definitions 3
13.10.1.3 Assumptions 3
13.10.1.4 Motivation 3
13.10.2 Scope and Field of Application 3
13.10.3 Specific Security Model 3
13.10.4 Services Offered 3
13.10.5 Services Required 3
13.10.6 Protocols 4
13.10.7 Management Elements Required/Impacted 4
13.10.8 Conformance Class Definitions 4
13.10.9 Conformance Class Specifications 4
13.10.10 Registration Issues Requirements 4
13.11 SESSION LAYER 4
13.11.1 Introduction 4
13.11.1.1 References 4
13.11.1.2 Definitions 4
13.11.1.3 Assumptions 4
13.11.1.4 Motivation 4
13.11.2 Scope and Field of Application 4
13.11.3 Specific Security Model 4
13.11.4 Services Offered 4
13.11.5 Services Required 4
13.11.6 Protocols 4
13.11.7 Management Elements Required/Impacted 4
13.11.8 Conformance Class Definitions 4
13.11.9 Conformance Class Specifications 4
13.11.10 Registration Issues Requirements 4
13.12 PRESENTATION LAYER 4
13.12.1 Introduction 4
13.12.1.1 References 4
13.12.1.2 Definitions 5
13.12.1.3 Assumptions 5
13.12.1.4 Motivation 5
13.12.2 Scope and Field of Application 5
13.12.3 Specific Security Model 5

13.12.4 Services Offered 5
13.12.5 Services Required 5
13.12.6 Protocols 5
13.12.7 Management Elements Required/Impacted 5
13.12.8 Conformance Class Definitions 5
13.12.9 Conformance Class Specifications 5
13.12.10 Registration Issues Requirements 5
13.13 APPLICATION LAYER 5
13.13.1 Introduction 5
13.13.1.1 References 5
13.13.1.2 Definitions 5
13.13.1.3 Assumptions 5
13.13.1.4 Motivation 5
13.13.2 Scope and Field of Application 5
13.13.3 Specific Security Model 5
13.13.4 Services Offered 5
13.13.4.1 ACSE 5
13.13.4.2 ROSE 5
13.13.4.3 TRSE 6
13.13.4.4 CCR 6
13.13.5 Services Required 6
13.13.6 Protocols 6
13.13.7 Management Elements Required/Impacted 6
13.13.8 Conformance Class Definitions 6
13.13.9 Conformance Class Specifications 6
13.13.10 Registration Issues Requirements 6
13.14 FTAM 6
13.14.1 Introduction 6
13.14.1.1 References 6
13.14.1.2 Definitions 6
13.14.1.3 Assumptions 6
13.14.1.4 Motivation 6
13.14.2 Scope and Field of Application 6
13.14.3 Specific Security Model 6
13.14.4 Services Offered 6
13.14.5 Services Required 6
13.14.6 Protocols 6
13.14.7 Management Elements Required/Impacted 6
13.14.8 Conformance Class Definitions 6
13.14.9 Conformance Class Specifications 6
13.14.10 Registration Issues Requirements 6
13.15 Message Handling System Security 7
13.15.1 Definitions of Elements of Security Service . . . 9
13.16 DIRECTORY 11
13.16.1 Introduction 11
13.16.1.1 References 11
13.16.1.2 Definitions 11
13.16.1.3 Assumptions 11
13.16.1.4 Motivation 11
13.16.2 Scope and Field of Application 11
13.16.3 Specific Security Model 11
13.16.4 Services Offered 11

13.16.5 Services Required 12
13.16.6 Protocols 12
13.16.7 Management Elements Required/Impacted 12
13.16.8 Conformance Class Definitions 12
13.16.9 Conformance Class Specifications 12
13.16.10 Registration Issues Requirements 12
13.17 VTP 12
13.17.1 Introduction 12
13.17.1.1 References 12
13.17.1.2 Definitions 12
13.17.1.3 Assumptions 12
13.17.1.4 Motivation 12
13.17.2 Scope and Field of Application 12
13.17.3 Specific Security Model 12
13.17.4 Services Offered 12
13.17.5 Services Required 12
13.17.6 Protocols 12
13.17.7 Management Elements Required/Impacted 12
13.17.8 Conformance Class Definitions 12
13.17.9 Conformance Class Specifications 12
13.17.10 Registration Issues Requirements 12

14. ISO VIRTUAL TERMINAL PROTOCOL 1
14.1 INTRODUCTION 1
14.2 SCOPE AND FIELD OF APPLICATION 1
14.2.1 Phase Ia Agreements 1
14.2.2 Phase Ib Agreements 1
14.2.3 Phase II Agreements 1
14.3 STATUS 1
14.3.1 Status of Phase Ia 1
14.3.2 Status of Phase Ib 2
14.3.3 Status of Phase II 2
14.4 ERRATA 2
14.5 CONFORMANCE 2
14.6 PROTOCOL 2
14.7 NIST REGISTERED CONTROL OBJECTS 2
14.7.1 Sequenced Application (SA) 2
14.7.2 Unsequenced Application (UA) 2
14.7.3 Sequenced Terminal (ST) 3
14.7.4 Unsequenced Terminal (UT) 3
14.7.5 Termination Conditions CO (TC) 3
14.7.5.1 Entry Number 3
14.7.5.2 Name of Sponsoring Body 3
14.7.5.3 Date 3
14.7.5.4 Identifier 3
14.7.5.5 Descriptor Value 3
14.7.5.6 CO VTE-parameters 4
14.7.5.7 CO Values, Semantic and Update Syntax 4
14.7.5.8 Additional Information 5
14.7.5.9 Usage 5
14.8 OIW DEFINED VTE-PROFILES 5
14.8.1 Telnet Profile 5

14.8.2 Transparent Profile 5
14.8.3 Forms Profile 5
14.8.4 X3 Profile 6
14.8.5 Scroll Profile 6
14.8.5.1 Introduction 6
14.8.5.2 Association Requirements 6
14.8.5.2.1 Functional Units 6
14.8.5.2.2 Mode 6
14.8.5.3 Profile Body 6
14.8.5.4 Profile Argument Definitions: 10
14.8.5.5 Profile Dependent CO Information. 12
14.8.5.6 Profile Notes 12
14.8.5.6.1 Definitive Notes. 12
14.8.5.6.2 Informative Notes 12
14.8.5.7 Specific Conformance Requirements 13
14.9 APPENDIX A: NO TITLE. 14
14.10 APPENDIX B: CLARIFICATIONS 15
14.10.1 Defaults 15
14.11 APPENDIX C: OBJECT IDENTIFIERS 16

15. TRANSACTION PROCESSING 1
15.1 INTRODUCTION 1
15.2 SCOPE. 1
15.3 SPECIFICATION OF FUNCTIONAL UNITS. 2
15.3.1 Functinal Units. 2
15.3.2 Combinations of Functional Units 2
15.4 TP USEOF OSI SERVICES. 2
15.4.1 ACSE - Association Management. 2
15.4.2 CCR - Provided Acid Properties 2
15.4.3 Presentation Services. 2
15.4.4 Directory Services 2
15.5 IMPLEMENTATIONS ISSUES NOT SPECIFIED IN ISO 10026. 3
15.5.1 Application Context. 3
15.5.2 Security 3
15.5.3 Recommended Practices. 3
15.6 CONFORMANCE STATEMENT. 3
15.7 OSI TRANSACTION PROCESSING PROTOCOL AGREEMENTS 3

16. OFFICE DOCUMENT ARCHITECTURE 1

17 ODA INTRODUCTION 1
17.1 SCOPE AND FIELD OF APPLICATION 1
17.2 REFERENCES 1
17.3 DEFINITIONS AND ABBREVIATIONS 4
17.4 POSITION OF THIS DAP IN THE TAXONOMY OF RELATED DAPS 4
17.4.1 AOW ODA SIG 5
17.4.2 CCITT SG VIII, Q26 5
17.4.3 EWOS ODA EG 5
17.4.4 NIST ODA SIG 5
17.4.5 PAGODA 5
17.5 CONFORMANCE 5
17.5.1 Data stream conformance 5

17.5.2 Implementation conformance 6
17.6 CHARACTERISTICS SUPPORTED BY THIS DAP 6
17.7 SPECIFICATION OF CONSTITUENT CONSTRAINTS 7
17.7.1 Document profile constraints 7
17.7.1.1 Macro definitions 7
17.7.1.2 Constituent constraints 9
17.7.1.2.1 Presence of document constituents . 9
17.7.1.2.2 Document characteristics 10
17.7.1.2.3 Document management attributes . . 12
17.7.2 Logical constituent constraints 12
17.7.2.1 Diagrams of relationships of logical constituents 12
17.7.2.2 Macro definitions 16
17.7.2.3 Factor constraints 19
17.7.2.4 Constituent constraints 20
17.7.2.4.1 LogDoc 20
17.7.2.4.2 Passage 20
17.7.2.4.3 NumberedSegment 20
17.7.2.4.4 Number 21
17.7.2.4.5 Paragraph 21
17.7.2.4.6 FNote 21
17.7.2.4.7 FNBody 21
17.7.2.4.8 BodyText 22
17.7.2.4.9 BodyRaster 22
17.7.2.4.10 BodyGeometric 22
17.7.2.4.11 CommonContent 23
17.7.2.4.12 PageNumber 23
17.7.3 Layout constituent constraints 23
17.7.3.1 Diagrams of relationships of layout constituents 23
17.7.3.2 Macro definitions 26
17.7.3.3 Factor constraints 27
17.7.3.4 Constituent constraints 29
17.7.3.4.1 LayDoc 29
17.7.3.4.2 PageSet 29
17.7.3.4.3 Page 29
17.7.3.4.4 RPage 30
17.7.3.4.5 VPage 30
17.7.3.4.6 CompositeHeaderFooter 30
17.7.3.4.7 CompositeBodyFixed 30
17.7.3.4.8 ColumnFixed 31
17.7.3.4.9 ColumnVariable 31
17.7.3.4.10 SnakingColumns 32
17.7.3.4.11 SynchronisedColumns 32
17.7.3.4.12 FootNote 33
17.7.3.4.13 ArrangedContentFixed 34
17.7.3.4.14 ArrangedContentVariable 34
17.7.3.4.15 SourcedContentFixed 35
17.7.3.4.16 SourcedContentVariable 35
17.7.3.4.17 BasicHeaderFooter 35
17.7.3.4.18 BasicBody 36
17.7.3.4.19 Block 36

17.7.4 Layout style constraints 37
17.7.4.1 Factors constraints 37
17.7.4.2 Constituent constraints 37
17.7.4.2.1 LStyle1 37
17.7.4.2.2 LStyle2 37
17.7.4.2.3 LStyle3 37
17.7.4.2.4 LStyle4 37
17.7.4.2.5 LStyle5 38
17.7.4.2.6 LStyle6 38
17.7.5 Presentation style constraints 38
17.7.5.1 Macro definitions 38
17.7.5.2 Factor constraints 40
17.7.5.3 Constituent constraints 40
17.7.5.3.1 PStyle1 40
17.7.5.3.2 PStyle2 41
17.7.5.3.3 PStyle3 41
17.7.5.3.4 PStyle4 41
17.7.6 Content portion constraints 41
17.7.6.1 Character content portion 41
17.7.6.2 Raster graphics content portion 42
17.7.6.3 Geometric graphics content portion 42
17.7.7 Additional usage constraints 42
10.8 INTERCHANGE FORMAT 43
17.8.1 ASN.1 generation constraints 43
17.8.2 Encoding of application comments 43
17.8.3 Encoding of raster content information 43
17.9 GSS/RSS proforma 43
17.9.1 Generator support statement proforma 43
17.9.2 Receiver support statement proforma 43
17.10 Informative Recommendations 44
17.10.1 ISO 8632 (CGM) constraints for this DAP 44
17.10.1.1 Delimiter elements 44
17.10.1.2 Metafile description elements 44
17.10.1.3 Picture descriptor elements 45
17.10.1.4 Control elements 45
17.10.1.5 Graphical primitive elements 45
17.10.1.6 Attribute elements 45
17.10.1.7 External Elements 46
17.10.2 Registration of entities 47
17.10.3 Conveyance of ODA over CCITT X.400-1984 47
17.10.3.1 P2 protocol encoding 48
17.10.3.2 P1 protocol encoding 48
17.10.4 Interoperability with SGML applications 48

18. NETWORK MANAGEMENT 1
18.1 INTRODUCTION 1
18.1.1 References 2
18.2 SCOPE AND FIELD OF APPLICATION 5
18.2.1 Use of Evolving Standards 9
18.2.2 Management Architecture 11
18.2.2.1 Systems Management Overview 11
18.2.2.2 Constraints/Assumptions for Phase 1 14

18.2.2.3 Migration to Future Phases 15
18.2.2.4 Relationship to Other Management Specifications 15
18.2.3 Management Scenarios 15
18.3 STATUS 16
18.4 ERRATA 16
18.5 MANAGEMENT FUNCTIONS AND SERVICES 16
18.5.1 Object Management Function Agreements 19
18.5.1.1 Object Creation Operation Agreements: 20
18.5.1.2 Object Deletion Operation Agreements: 23
18.5.1.3 Object Renaming Operation Agreements: 25
18.5.1.4 Attribute Reading Operation Agreements: . . . 26
18.5.1.5 Attribute Changing Operation Agreements: . . 27
18.5.1.6 Object Listing Operation Agreements: 29
18.5.1.7 Object Management Services Agreements 30
18.5.1.7.1 Enrol Object Service Agreements . . 31
18.5.1.7.2 Deenrol Object Service Agreements: 32
18.5.1.7.3 Reenrol Object Service Agreements: 32
18.5.1.7.4 Attribute Change Event Report Service 33
18.5.1.7.5 Add Value Event Report Service Agreements: 34
18.5.1.7.6 Remove Value Event Report Service Agreements: 34
18.5.2 State Management Function Agreements 35
18.5.2.1 State Reading Operation Agreements: 36
18.5.2.2 State Changing Operation Agreements: 37
18.5.2.3 State Change Reporting Service Agreements: . 38
18.5.3 Relationship Management Function 39
18.5.3.1 Relationship Creation Service Agreements . . 42
18.5.3.2 Relationship Deletion Service Agreements . . 43
18.5.3.3 Relationship Change Service Agreements . . . 45
18.5.3.4 Relationship Listing Service Agreements . . . 46
18.5.3.5 Related Object Listing Service Agreements: . 47
18.5.3.6 Relationship Creation Report Service Agreements 48
18.5.3.7 Relationship Deletion Report Service Agreements 48
18.5.3.8 Relationship Change Report Service Agreements 49
18.5.3.9 The usage of compound Relationship attributes 'Group' 49
18.5.3.10 The usage of the combined Add/Change/Delete Services 50
18.5.4 Error Reporting and Information Retrieval . . 50
18.5.4.1 Error Reporting Service Agreements: 50
18.5.4.1.1 Error Reporting Model Agreements: . 52
18.5.4.1.2 Support Managed Object Agreements: 52
18.5.4.1.3 Error Reporting Service Agreements: 52
18.5.4.2 Information Retrieval Function Agreements: . 55
18.5.4.2.1 Information Retrieval Service Agreements: 55
18.5.5 Management Service Control Functions Agreements: . 55

18.5.5.1 Event Reporting Control Function Agreements: 55
18.5.5.1.1 Event Reporting Control Model . . . 56
18.5.5.1.2 Support Managed Object - Event . . 57
18.5.5.1.3 Initiate Event Reporting Service . 59
18.5.5.1.4 Terminate Event Reporting Service 61
18.5.5.1.5 Suspend Event Reporting Service . . 63
18.5.5.1.6 Resume Event Reporting Service Agreements: 64
18.5.5.1.7 Modify Event Forwarding Discriminator 65
18.5.5.1.8 Retrieve Event Forwarding Discriminator 67
18.5.5.2 Service Access Control Function Agreements: . 68
18.5.6 Event Logging Control Function Agreements: 68
18.5.6.1 Event Logging Model Agreements: 68
18.5.6.2 Support Managed Object Agreements: 68
18.5.6.2.1 Log Discriminator Agreements: . . . 68
18.5.6.2.2 LOG Agreements: 69
18.5.6.3 Log Control Services Agreements: 69
18.5.6.3.1 Initiate Event Logging Service Agreements: 69
18.5.6.3.2 Terminate Event Logging Service Agreements: 69
18.5.6.3.3 Suspend Event Logging Service Agreements: 69
18.5.6.3.4 Resume Event Logging Service Agreements: 69
18.5.6.3.5 Modify Event Logging Parameters Service 69
18.5.6.3.6 Event Log Parameters Retrieval Service 69
18.6 MANAGEMENT COMMUNICATIONS 69
18.6.1 Association Policies 69
18.6.1.1 Types of Association 70
18.6.1.2 Functional Units 70
18.6.1.3 Functional Unit Negotiation 70
18.6.1.4 Span of an Association 70
18.6.1.5 Other Aspects of Associations 71
18.6.2 Agreements on CMIS 71
18.6.2.1 Object Naming 71
18.6.2.2 Multiple Object Selection 71
18.6.2.2.1 Scoping 72
18.6.2.2.2 Filtering 74
18.6.2.2.3 Synchronization 77
18.6.2.2.4 Linked Replies 77
18.6.2.3 Time 79
18.6.2.4 Access Control 80
18.6.2.5 Error Handling 80
18.6.3 Agreements on CMIP 80
18.6.3.1 General PDU Agreements 81
18.6.3.1.1 Invoke Ids 81

18.6.3.1.2 Access Control 81
18.6.3.1.3 Time 81
18.6.3.2 Specific PDU Agreements 82
18.6.3.2.1 M-Initialize 82
18.6.3.2.2 M-Terminate 82
18.6.3.2.3 M-Abort 83
18.6.3.2.4 M-Event-Report 83
18.6.3.2.5 M-Get 84
18.6.3.2.6 M-Set 86
18.6.3.2.7 M-Action 88
18.6.3.2.8 M-Create 90
18.6.3.2.9 M-Delete 92
18.6.4 Services Required by CMIP 93
18.7 MANAGEMENT INFORMATION 94
18.7.1 The Information Model 94
18.7.1.1 Basic Concepts 96
18.7.1.2 Management Operations Supported 97
18.7.1.3 Filter 97
18.7.1.4 Inheritance 97
18.7.1.5 Polymorphism 98
18.7.2 Principles of Naming 98
18.7.2.1 Containment Hierarchy 98
18.7.2.2 Name Structure 99
18.7.2.2.1 Object Class Identification 99
18.7.2.2.2 Object Instance Identification . . 99
18.7.2.2.3 Selection Of Distinguishing Attributes 100
18.7.2.2.4 Attribute Identification 101
18.7.3 Guidelines for the Definition of Management Information 101
18.7.3.1 Syntactical Definitions of Management Information 101
18.7.3.1.1 Managed Object Class Template . . . 101
18.7.3.1.2 Name Binding Template 102
18.7.3.1.3 Attribute Template 102
18.7.3.1.4 Group Attribute Template 102
18.7.3.1.5 Action TEmplate 102
18.7.3.1.6 Notification Template 102
18.7.3.2 Semantic Definitions of Management Information 102
18.7.3.3 Other Guidelines 103

19. REMOTE DATABASE ACCESS (RDA) 1

20. MANUFACTURING MESSAGE SPECIFICATION (MMS) 1
20.1 INTRODUCTION 1
20.1.1 References 1
20.2 SCOPE AND FIELD OF APPLICATION 1
20.2.1 Phase I Agreements 2
20.3 STATUS 2
20.3.1 Status of Phase 1 Agreements 2
20.4 ERRATA 2

20.5 SPECIFIC SERVICE AGREEMENTS 2
20.5.1 Initiate 2
20.5.1.1 Max Serv Outstanding 2
20.5.1.2 Version Number 3
20.5.1.3 Minimum Supported PDU Size 3
20.5.1.4 Max Supported PDU Size 3
20.5.1.5 Negotiation of MMS Abstract Syntaxes. 5
20.5.2 Scattered Access 5
20.6 INTEROPERABILITY AGREEMENTS. 5
20.6.1 Calling MMS-user Interoperability Agreements . . . 6
20.6.2 Called MMS-user Interoperability Agreements. . . . 6
20.6.3 General Interoperability Agreements. 6
20.6.3.1 VMD Logical Status. 6
20.6.3.2 No Title. 7
20.7 APPENDIX A: DIS 9506 MODIFICATIONS REQUIRED FOR INTEROPERABILITY 8
20.7.1 References 8
20.7.2 Version 0. 8
20.7.2.1 General 8
20.7.2.1.1 Implementation Base 8
20.7.2.1.2 Rules of Extensibility. 8
20.7.2.2 Modifications to the Protocol definitions . . 8
20.7.2.2.1 Page39,Section 7.5.2 of DIS 9506-2 8
20.7.2.2.2 Page49,Section 7.6.4 of DIS 9506-2 9
20.7.2.2.3 Page95,Section 12.2.1 of DIS 9506-2 9
20.7.2.2.4 Page96,Section 12.3.1 of DIS 9506-2 9
20.7.2.2.5 Page98,Section 12.4.2 of DIS 9506-2 9
20.7.2.2.6 Page138,Section 15.14 of DIS 9506-2 10
20.7.2.2.7 Page166,Section 17.10 of DIS 9506-2 10
20.7.2.3 Behavior Requirements 10
20.7.2.3.1 Filenames 10
20.7.2.3.2 Identify Service. 10
20.7.2.3.3 Initiate Service. 10
20.7.2.3.3.1 Segment Size 11
20.7.2.3.3.1.1 Minimum Segment Size . . . 11
20.7.2.3.3.1.2 Maximum Segment Size . . . 11
20.7.2.3.3.2 Abstract Syntax Name 12
20.7.2.3.3.3 Application Context Name . . . 12
20.7.2.3.4 Minor Version Number. 12
20.7.2.3.5 Parameter CBB Subset. 12
20.7.3 Service Subset 12

21. REFERENCES 1

List of Figures

Figure 2.1 FDDI Frame Format 3
Figure 6.1. Structure of Object Identifier for OIW 4
Figure 6.2. Structure of an Object Identifier for a bogus object for the Registration Authority SIG of OIW 4
Figure 8.9 Example of Unregistered Object Class Definition 9
Figure 17.4: Structure for logdoc and passage 13
Figure 17.5: Structure for paragraph 14
Figure 17.6: Structure for fnote 14
Figure 17.7: Structure for numbered segment 15
Figure 17.8: Structure for common content 15
Figure 17.9: Structure for layout document root and page set 24
Figure 17.10: Structure for Page, VPage and RPage 25
Figure 17.11: Structure for CompositeHeaderFooter 25
Figure 17.12 Structure for CompositeBody 26

List of Tables

Table 2.1 - ANSI-CCITT Cross-References 6
Table 8.13 MHS Security : IPM Elements of Service 7
Table 8.14 Use of Directory: MT Elements of Service 10
Table 8.15 Use of Directory: IPM Elements of Service 10
Table 8.16 MHS Security: MT Elements of Service 11
Table 8.17 MHS Security: IPM Elements of Service 11
Table 8-18 Physical Delivery: MT Èlements of Service 12
Table 8-19 Physical Delivery: IPM Elements of Service. 13
Table 8-20 Physical Delivery O/R Address Attributes 14
Table 8-21 Character String Support 14
Table 8-22 Coversion: MT Elements of Service 16
Table 13.1 X.400 Relationship between Elements of Security Service and MHS Components 8
Table 18.1 Relevant Standards Documents and the Current Schedules for Progressing these Documents to IS Status 10

1. GENERAL INFORMATION

1.1 PURPOSE OF THIS DOCUMENT

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

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

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

o The companion document, "Stable Implementation Agreements for Open Systems Interconnection Protocols, Version 3, Edition 1, December 1989" records mature agreements considered advanced enough for use in product development or procurement reference.

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

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

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

An implementor should look at the aligned section in the Stable Document to get the true current status of stable material. In this Working Document, all references to the Stable Document are to V3, E1 (December 1989). Where appropriate, statements related to backward compatibility, interworking considerations, or agreement maintenance are given in this document.

1.2 PURPOSE OF THE WORKSHOP

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

1.3 WORKSHOP ORGANIZATION

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

1.4 USE AND ENDORSEMENT BY OTHER ENTERPRISES

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

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

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

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

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

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

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

1.5 RELATIONSHIP OF THE WORKSHOP TO THE NIST LABORATORIES

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

The NIST laboratories bear no other relationship to the Workshop.

1.6 STRUCTURE AND OPERATION OF THE WORKSHOP

1.6.1 Plenary

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

1.6.2 Special Interest Groups

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

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

Following are procedures for cooperative work among Special Interest Groups.

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

Following are the charters of the Special Interest Groups.

FTAM SIG

Scope

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

  and

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

High priority work items:

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

Low priority work items:

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

(MESSAGE HANDLING SYSTEMS) SIG

Develop and maintain product level specifications for Message Handling Systems using the CCITT X.400 recommendations (and corresponding documents). Review Abstract Tests for X.400 and provide feedback to the appropriate bodies.

LOWER LAYER SIG

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

The objectives of the Lower Layer SIG are:

- o Study OSI layers 1-4 as directed by the plenary - such study is to include management objects, security, ISDN user-network interfaces for use in conjunction with OSI network services, routing exchange protocols, etc.
- o Produce and maintain recommendations for implementation of these layers,
- o Where necessary, provide input to the relevant standards bodies concerning layers 1-4, in the proper manner, and
- o Review base standard abstract test suites with the goal of identifying the test cases required for the layer 1-4 Implementation Agreements. Develop test cases for Implementation

Agreement functionality not present in the base standard (if any).

OSI SECURITY ARCHITECTURE SIG

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

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

OBJECTIVES:

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

DIRECTORY SERVICES SIG

Produce functional implementation agreements based on ISO/CCITT specifications for Directory Services in accordance with the objectives and goals of the plenary.

- Provide a subset for NIST publication which is functional and forward compatible to further work by this Special Interest Group.
- Define stable core functionality which can be implemented in the near term.

VIRTUAL TERMINAL SIG

This Special Interest Group's charter is based upon the implementation of International Standards 9040 and 9041 in providing Basic Virtual Terminal Service.

This group will develop agreements for the implementation and testing of the following VTE-profiles.

- X.29 PAD
- TELNET
- Basic Scrolling
- Basic Paging
- Basic Forms

UPPER LAYERS SIG

The charter of the Upper Layers SIG is as follows.

- Develop product level specifications for the implementation of:
    - Session service and protocol
    - Presentation service and protocol
    - ACSE service and protocol
    - Remote Operations Service Element (ROSE)
    - Reliable Transfer Service Element (RTSE)

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

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

The objectives of the Upper Layers SIG are to:

- Study OSI Session, Presentation, ACSE, ROSE, and RTSE

- Incorporate implementor's agreements in the 1988 NBS standing document,

- Produce and maintain recommendations for implementations of these layers,

- Where necessary provide input to the relevant standards bodies concerning Session, Presentation, ACSE, ROSE, and RTSE

- React in a timely manner (i.e., to develop corresponding implementor's agreements) to technical changes in ISO documents.

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

- Align implementation agreements with other organizations such as ANSI and ISO,
- Develop implementor's agreements that promote the efficiency of protocols,
- Develop implementor's agreements that promote ease in the verification of interoperability,
- Develop necessary conformance statements.

NETWORK MANAGEMENT SIG

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

The SIG will:

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

As necessary, the SIG will:

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

OFFICE DOCUMENT ARCHITECTURE

The SIG will:

- develop one or more product level specifications for implementations of ISO/DIS 8613, i.e., the SIG will define one or more Document Application Profiles (DAPs)

- develop requirements for conformance testing of products purporting conformance to the (se) DAP (s)

- specify and describe requirements for services that manage the generation and interpretation of the ODA document representation

- determine preferred relationships between ODA and other document interchange formats

- promote the SIG's agreements (e.g., presentations, product demonstrations, press releases)

As necessary, the SIG will:

- establish liaison with required SIGs (e.g., , FTAM, and Upper Layers SIGs) to seek efficient transfer capability for document interchange based on the ODA SIG agreements

- provide feedback and liaison to groups working on ISO/DIS 8613 related activities

REGISTRATION SIG

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

A. Registration of NIST OSI Workshop-Specified Objects.

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

1. Define policies and procedures for the registration of objects defined by the NIST OSI Workshop,
2. Take account of currently existing OSI Workshop registration work,
3. Establish policies for the publication and promulgation of registered objects;
4. Liaise with other OSI Workshop SIGs, appropriate standards bodies (e.g., ANSI) and other appropriate organizations.

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

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

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

TRANSACTION PROCESSING SIG

The SIG will be the focal point for all work on Transaction Processing within the Workshop. In particular:

1. Define DP/DIS/IS 10026 (TP) Implementation Agreements.

2. Liaise with Upper Layers SIG to define DIS/IS 9805 (CCR) Implementation Agreements to satisfy TP requirements.

3. Liaise with other internal and external organizations as required.

MANUFACTURING MESSAGE SPECIFICATION (MMS) SIG

Scope

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

Objectives

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

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

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

- Develop Conformance requirements

- Develop recommendations on MMS testing

As Necessary

- Respond to defect reports as accepted

- Provide feedback on Addendum material

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

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

High Priority Work Items

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

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

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

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

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

Low Priority Work Items

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

- Develop subsequent subsets of MMS

- Produce a set of implementors agreements for the subsequent subsets

- Provide input on Test Cases for the subsequent subsets

- Provide input on errors for the subsequent subsets

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

REMOTE DATABASE ACCESS SIG

Scope:

For all RDA Implementations based on ISO 9579:

- Develop Implementors' agreements;

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

- Coordinate with other organizations on matters relevant to RDA.

Objectives:

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

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

High Priority Work Items

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

   a. review ULA agreements affecting RDA implementations,

   b. specify limits on encodings in RDA pdus,

   c. specify minimum conformance requirements for RDA implementations,

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

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

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

   g. identify RDA implementation requirements for CCR and TP,

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

Low Priority Work Items

1. Future RDA specializations, if any.

1.7 POINTS OF CONTACT

OSI Workshop - Chairman	Tim Boland	NIST	(301) 975-3608
OSI Workshop - Registration	Brenda Gray	NIST	(301) 975-3664
Directory Services SIG	Chris Moore	Touch Comm.	(408) 374-2500
FTAM SIG	Klaus Truoel	GMD/DFN	49-615-1-875-700
Lower Layers SIG	Fred Burg	AT&T	(201) 949-0919
Manufacturing Message Specification (MMS) SIG	Herbert Falk	SISCO	(313) 774-0070
Network Management SIG	Paul Brusil	Mitre	(617) 271-7632
ODA SIG	Frank Dawson	IBM	(214) 556-5052
Remote Database Access SIG	Rich Gerhardt	GM	(313) 947-0572
Security SIG	James Galvin	Trusted Info. Sys.	(301) 854-6889
Technical Liaison Committee	Einar Stefferud	NMA-Northrop	(714) 842-3711
Transaction Processing SIG	Andrew P. Schwartz	IBM Corp.	(415) 855-4766
Upper Layers SIG	David Chappell	Cray Research	(612) 825-7928
Virtual Terminal SIG	Cyndi Jung	3COM	(415) 940-7664
X.400 SIG	Barbara Nelson	Retix	(213) 399-1611
MAP	Gary Workman	GM	(313) 947-0599
TOP	Laurie Bride	BCS	(206) 763-5719
Government OSI Profile	Jerry Mulvenna	NIST	(301) 975-3631

2. SUB NETWORKS

Editor's Note: All references to Stable Agreements in this Section are to Version 3, Edition 1, dated December 1989.

2.1 INTRODUCTION

(Refer to Stable Implementation Agreements Document)

2.2 SCOPE AND FIELD OF APPLICATION

(Refer to Stable Implementation Agreements Document)

2.3 STATUS

This material is current as of December 15, 1989.

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

2.4 ERRATA

Errata are reflected in pages of Version 3, Edition 1, Stable Document, dated December 1989.

2.5 LOCAL AREA NETWORKS

(Refer to Stable Implementation Agreements Document)

2.5.1 IEEE 802.2 Logical Link Control

(Refer to Stable Implementation Agreements Document)

2.5.2 IEEE 802.3 CSMA/CD Access Method

- For a data packet of LLC data length of n octets, the length of the pad field shall be

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

2.5.3 IEEE 802.4 Token Bus Access Method

(Refer to Stable Implementation Agreements Document)

2.5.4 IEEE 802.5 Token Ring Access Method

(Refer to Stable Implementation Agreements Document)

2.5.5 Fiber Distributed Data Interface (FDDI)

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

The following are implementation agreements with respect to FDDI MAC.

1 The address length shall be 48 bits.

2 The term "default" is defined to be the value of a parameter in an FDDI station or concentrator as originally supplied by the vendor. Stations need not be reset to the default values by a power off condition, but there shall be some manual or programmatic means of resetting stations and concentrators to the specified default values.

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

4 The value of T_Req shall be equal to T_Max unless set otherwise by the Network Manager or by a concentrator initializing a slave tree to achieve "graceful insertion".

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

P	SD	FC	DA	SA	Info	FCS	ED	FS

Figure 2.1 FDDI FRAME FORMAT

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

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

6 Stations shall not use restricted token service.

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

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

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

- one microsecond in a station, and

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

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

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

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

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

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

2.6 X.25 WIDE AREA NETWORKS

2.6.1 Introduction

(Refer to the Stable Implementation Agreements Document).

2.6.2 ISO 7776

(Refer to the Stable Implementation Agreements Document).

2.6.3 ISO 8208

(Refer to the Stable Implementation Agreements Document).

2.7 INTEGRATED SERVICES DIGITAL NETWORKS (ISDN)

2.7.1 Introduction

(Refer to the Stable Implementation Agreements Document).

2.7.2 Implementation Agreements

(Refer to the Stable Implementation Agreements Document).

2.7.2.1 Physical Layer, Basic Access at "U"

(Refer to the Stable Implementation Agreements Document).

2.7.2.2 Physical Layer, Basic Access at S and T

(Refer to the Stable Implementation Agreements Document).

2.7.2.3 Physical Layer, Primary Rate at "U"

(Refer to the Stable Implementation Agreements Document).

2.7.2.4 Data Link Layer, D-Channel

(Refer to the Stable Implementation Agreements Document).

2.7.2.5 Signaling

(Refer to the Stable Implementation Agreements Document).

2.7.2.6 Data Link Layer B-Channel

(Refer to the Stable Implementation Agreements Document).

2.7.2.7 Packet Layer

(Refer to the Stable Implementation Agreements Document).

2.8 APPENDIX A

(Refer to the Stable Implementation Agreements Document.)

2.8.1 Data Link Layer, D-Channel

(Refer to the Stable Implementation Agreements Document.)

2.8.2 Signaling

(Refer to the Stable Implementation Agreements Document.)

3. NETWORK LAYER

Editor's Note: All references to Stable Agreements in this Section are to Version 3, Edition 1, dated December 1989.

3.1 INTRODUCTION

(Refer to the Stable Agreements Document)

3.2 SCOPE AND FIELD OF APPLICATION

(Refer to the Stable Agreements Document)

3.3 STATUS

This material is current as of December 15, 1989.

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

3.4 ERRATA

Errata are reflected in pages of Version 3, Edition 1 Stable Document, dated December 1989.

3.5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS)

3.5.1 ISO 8473

1. Subsets of the protocol:

(Refer to the Stable Implementation Agreements Document).

2. Mandatory Functions:

(Refer to the Stable Implementation Agreements Document).

3. Optional Functions:

- o (Refer to the Stable Implementations Agreements document).

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

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

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

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

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

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

     - "high" priority will be encoded with value 10 (0000 0000 0000 1010),
     - "normal" priority will be encoded with value 5 (0000 0000 0000 0101), and

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

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

  3 Network entities supporting priority shall process PDUs in which the priority parameter is absent as either "low", "normal", or "high" according to a locally configurable parameter. This is to ensure

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

that NPDUs not containing the priority parameter can be processed by intermediate systems in a defined manner with respect to those which do contain the priority parameter.

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

3.5.2 Provision of CLNS over Local Area Networks

(Refer to the Stable Agreements Document)

3.5.3 Provision of CLNS over X.25 Subnetworks

(Refer to the Stable Agreements Document)

3.5.4 Provision of CLNS over ISDN

(Refer to the Stable Implementation Agreements document).

3.5.4.1 CLNP Utilizing X.25 Services

(Refer to the Stable Implementations Agreements document).

3.5.5 Provision of CLNS over Point-to-Point Links

(To be based on ISO 8880)

3.6 CONNECTION-MODE NETWORK SERVICE

3.6.1 Mandatory Method of Providing CONS

3.6.1.1 General

(Refer to the Stable Implementation Agreements document).

3.6.1.2 X.25 WAN

(Refer to the Stable Implementation Agreements document).

3.6.1.3 LANs

(Refer to the Stable Implementation Agreements document).

3.6.1.4 ISDN

(Refer to the Stable Implementation Agreements document).

3.6.1.5 PRIORITY

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

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

(Refer to the Stable Implementation Agreements Document)

3.6.3 Agreements on Protocols

(Refer to the Stable Implementation Agreements Document)

3.6.3.1 ISO 8878

(Refer to the Stable Implementation Agreements Document.)

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

(Refer to the Stable Implementation Agreements Document)

3.6.4 Interworking

(Refer to the Stable Implementation Agreements Document.)

3.7 ADDRESSING

- Refer to the Stable Implementations Agreements Document

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

2 octets	6 octets	1 octet
AREA	ID	N-Selector

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

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

The above recommendation may be applicable in other routing environments.

3.8 ROUTING

3.8.1 End System to Intermediate System Routing

(Refer to the Stable Implementation Agreements Document.)

3.8.2 Intermediate Systems to Intermediate Systems Routing

(Refer to the Stable Implementation Agreements)

3.9 PROCEDURES FOR OSI NETWORK SERVICE/PROTOCOL IDENTIFICATION

3.9.1 General

(Refer to the Stable Implementation Agreements document).

3.9.2 Processing of Protocol Identifiers

(Refer to the Stable Implementation Agreements document).

3.9.2.1 Originating NPDUs

(Refer to the Stable Implementation Agreements document).

3.9.2.2 Destination System Processing

(Refer to the Stable Implementation Agreements document).

3.9.2.3 Further Processing in Originating End System

(Refer to the Stable Implementation Agreements document).

3.9.3 Applicable Protocol Identifiers

(Refer to the Stable Implementation Agreements document.)

3.10 MIGRATION CONSIDERATIONS

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

3.10.1 X.25-1980

(Refer to the Stable Agreements Document)

3.11 USE OF PRIORITY[2]

3.11.1 Introduction

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

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

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

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

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

3.11.2 Overview

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

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

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

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

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

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

3.12 CONFORMANCE

(Agreements to be added at a later date)

3.13 APPENDIX A

This appendix discusses a problem concerning the operation of the ES-IS routing protocol of ISO 9542 in an IEEE 802.5 LAN. The proposal requires further technical review before considering it as having support as an implementation agreement.

Editor's Note: This Appendix represents a discussion paper introduced by one or a small number of LLSIG participants, and is reprinted here solely for future consideration of the SIG. THIS IS NOT AN IMPLEMENTATION AGREEMENT, AND MAY BE REMOVED IN THE FUTURE.

3.13.1 Problem Statement

o From NIST Stable Implementors' Agreements of March, 1989, Section 3.8.1 defines the following subnet point of attachment multicast addresses to support ES-IS:

- ALL_ESN = 0900 2B00 0004

- ALL_ISN = 0900 2B00 0005

o Claim is that these addresses work fine in IEEE802.3 and IEEE802.4 subnet environments, but will not work in practical real-world token ring IEEE802.5 networks.

o A "practical, real-world" token ring network is one in which the token ring LAN adapter is either a certain token ring adapter or one compatible to this kind of token ring adapter.

- Proof of this is that a certain vendor may have a large share of the IEEE802.5 token ring market. Most other vendors providing token ring adapters probably need to be compatible to adapters produced by this vendor.

- There are 2 problems:

  - NOTATIONAL - i.e., describing the ES-IS multicast addresses in the agreements for token ring in an unambiguous fashion
  - SUBSTANTIVE - Certain adapters do not allow the full range of possible IEEE802.5 multicast addresses. Concepts of "group" and "functional" multicast addresses are defined and these are the only types allowed. Anything else will be rejected by such adapters. The current agreed upon ES-IS multicast addresses do not fit the form accepted by these adapters.

3.13.2 Address Notational Considerations

- When an octet of an address string is written down in HEX notation, it represents 8 bits with the following convention:

  - The least significant bit (LSB) of the octet is on the right side and the most significant bit is on the left side. This is the opposite to the conventions used in the IEEE802 MAC level standards.

- So for the first octet of the ES-IS multicasts given in implementors agreements:

  -
    ```
    0X09 = 0   0   0   0   1   0   0   1
          MSB                         LSB
                                   |   |
                                   |   |
                                  U/L I/G
                                  2ND 1ST
                                  XMT XMT
                                  BIT BIT
    ```

  - I/G = Individual/Group (I.E. Multicast) BIT
    U/L = Universal/Locally Assigned BIT

  - In all IEEE802 MAC Standards, I/G always transmitted first and U/L always transmitted next.

- In IEEE802.3 and IEEE802.4 in each octet the LSB is transmitted first

- In IEEE802.5 in each octet the MSBof the information field is transmitted first. The address field Bits are transmitted in the sequence of 48 bits starting with I/G. Notationally to describe the address fields like the information fields, keeping the convention of MSB Bit transmitted first, the first octet of the address field is written as follows:

```
-    0X90 =   1    0    0    1    0    0    0    0
             MSB                                 LSB
              |    |
              |    |
              |    |
             I/G  U/L
             1ST  2ND
             XMT  XMT
             BIT  BIT
```

- Note in IEEE802.5, the bits of the first octet go out with I/G first and U/L second as for IEEE802.3 and IEEE802.4. However, the conventional computer science notation to represent the octets is reversed since in this notation LSB is always written to the right.

- Therefore, minimally we need to reverse the notation used in the implementor' agreements to represent the ES-IS multicast addresses for IEEE802.5.

3.13.3 Requirement to Use Functional Addressing

- Certain adapters do not support arbitrary multicast IEEE802 addresses (with first xmitted bit I/G set to 1).

- 2 classes of valid multicasts:

  - Group addresses (what standard calls conventional group mode) - only 1 such address can be registered with the adapter and therefore cannot be used for ES-IS

  - Functional address (what standard calls bit-significant mode) - Some are reserved; however, 12 of these user defined. Has format:

    -- 11000000 00000000 Followed by
    0XXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX

    -- 1 X Set to 1 with remaining X's set to 0.

- Anything else rejected by adapter or will not be properly filtered.

- Using conventional computer science notation:

  First 2 functional address octets = 0XC0 0X00

3.13.4 Proposal to Revise Agreements

- In Section 3.8.1, delete Item #9 and replace with a new #9 and #10 as follows:

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

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

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

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

      - either two addresses from the user-defined functional address space, such as:

        ALL_ESN = C000 0008 0000
        ALL_ISN = C000 0010 0000

      - or two addresses from the reserved space.

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

  - Renumber the current Items 10 and 11 of this Section to 11 and 12, respectively.

  - Note that 2 vendor allowed "user" functional addresses have been specified arbitrarily. It is recommended that the particular final choice of functional address selected by the SIG be verified with a prominent vendor. Perhaps this vendor will reserve a couple ("non-user") functional addresses for this purpose.

4. TRANSPORT LAYER

Editor's Note: All references to Stable Agreements in this Section are to Version 3, Edition 1, dated December 1989.

4.1 INTRODUCTION

(Refer to Stable Implementation Agreements Document)

4.2 SCOPE AND FIELD OF APPLICATION

(Refer to the Stable Implementation Agreements document).

4.3 STATUS

This material is current as of December 15, 1989.

> **Editor's Note:** The priority material (Section 4.5.1.2.6) in particular has been identified as a candidate for stability in March 1990.

4.4 ERRATA

Errata are reflected in pages of Version 3, Edition 1, Stable Document, dated December 1989.

4.4.1 ISO/CCITT Defect Reports

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

4.5 PROVISION OF CONNECTION MODE TRANSPORT SERVICES

(Refer to the Stable Implementation Agreements document).

4.5.1 Transport Class 4

4.5.1.1 Transport Class 4 Overview

(Refer to the Stable Implementation Agreements document).

4.5.1.2 Protocol Agreements

4.5.1.2.1 General Rules

(Refer to the Stable Implementation Agreements Document.)

4.5.1.2.2 Transport Class 4 Service Access Points or Selectors

(Refer to the Stable Implementation Agreements Document.)

4.5.1.2.3 Retransmission Timer

(Refer to the Stable Implementation Agreements Document.)

4.5.1.2.4 Keep-Alive Function

(Refer to the Stable Implementation Agreements Document.)

4.5.1.2.5 Congestion Avoidance Policies

(Refer to the Stable Implementation Agreements Document).

4.5.1.2.6 Use of Priority[3]

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

Additional requirements of systems implementing priority are defined below.

1 When Transport is implemented over a CLNS Network entity, each data TPDU and corresponding NSDU shall be assigned a priority level derived from the Transport

[3] Refer to Section 3.11 for an overview on the use of priority.

connection priority level, except as excluded in item 5b and 5d below[4].

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

3 The four abstract values corresponding to the four levels defined in 3.11 shall be encoded as follows:[5]

- "high reserved" priority will be encoded with value "zero" (0000 0000 0000 0000), and

- "high" priority will be encoded with value 5 (0000 0000 0000 0101),
- "normal" priority will be encoded with value 10 (0000 0000 0000 1010),

- "low" priority will be encoded with value 14 (0000 0000 0000 1110)

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

5 The exchange of priority parameters by Transport entities is performed as described below[6].

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

b A receiving Transport entity supporting priority management shall either accept the priority level proposed in the CR TPDU or select a lower level.

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

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

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

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

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

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

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

4.5.2 Transport Class 0

(Refer to Stable Implementation Agreements Document)

4.5.2.1 Transport Class 0 Overview

(Refer to Stable Implementation Agreements Document)

4.5.2.2 Protocol Agreements

4.5.2.2.1 Transport Class 0 Service Access Points

(Refer to Stable Implementation Agreements Document)

4.5.2.3 Rules for Negotiation

(Refer to Stable Implementation Agreements Document.)

4.5.3 Transport Class 2

(Refer to Stable Implementation Agreements Document.)

4.5.3.1 Transport Class 2 Overview

(Refer to Stable Implementation Agreements Document.)

4.5.3.2 Protocol Agreements

(Refer to Stable Implementation Agreements Document.)

4.6 PROVISION OF CONNECTIONLESS TRANSPORT SERVICE

(Refer to Stable Implementation Agreements Document.)

4.7 TRANSPORT PROTOCOL IDENTIFICATION

(Refer to the Stable Implementation Agreements document.)

5. UPPER LAYERS

Editor's Note: All references to Stable Agreements in this section are to Version 3, Edition 1, December 1989.

5.1 INTRODUCTION

(Refer to Stable Agreements Document)

5.1.1 References

(Refer to Stable Agreements Document)

5.2 SCOPE AND FIELD OF APPLICATION

(Refer to Stable Agreements Document)

5.3 STATUS

This version of the upper layer agreements is under development.

5.4 ERRATA

5.4.1 ISO Defect Solutions

(Refer to Stable Agreements Document)

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

(Refer to Stable Agreements Document)

5.5 ASSOCIATION CONTROL SERVICE ELEMENT

5.5.1 Introduction

(Refer to Stable Agreements Document)

5.5.2 Services

(Refer to Stable Agreements Document)

5.5.3 Protocol Agreements

5.5.3.1 Application Context

(Refer to Stable Agreements Document)

5.5.3.2 AE Title

(Refer to Stable Agreements Document)

5.5.4 ASN.1 Encoding Rules

(Refer to Stable Agreements Document)

5.5.5 Connectionless

(Refer to Stable Agreements Document)

5.5.6 Result Parameter

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

5.6 ROSE

(Refer to Stable Agreements Document)

5.7 RTSE

(Refer to Stable Agreements Document)

5.8 PRESENTATION

5.8.1 Introduction

(Refer to Stable Agreements Document)

5.8.2 Service

(Refer to Stable Agreements Document)

5.8.3 Protocol Agreements

5.8.3.1 Transfer Syntaxes

(Refer to Stable Agreements Document)

5.8.3.2 Presentation Context Identifier

(Refer to Stable Agreements Document)

5.8.3.3 Default Context

(Refer to Stable Agreements Document)

5.8.3.4 P-Selectors

(Refer to Stable Agreements Document)

5.8.3.5 Provider Abort Parameters

(Refer to Stable Agreements Document)

5.8.3.6 Provider Aborts and Session Version

(Refer to Stable Agreements Document)

5.8.3.7 CPC-Type

(Refer to Stable Agreements Document)

5.8.3.8 Presentation-context-definition-result-list

(Refer to Stable Agreements Document)

5.8.3.9 RS-PPDU

(Refer to Stable Agreements Document)

5.8.4 Presentation ASN.1 Encoding Rules

5.8.4.1 Invalid Encoding

(Refer to Stable Agreements Document)

5.8.5 General

5.8.5.1 Presentation Data Value (PDV)

(Refer to Stable Agreements Document)

5.8.6 Connection Oriented

(Refer to Stable Agreements Document)

5.8.7 Connectionless

(Refer to Stable Agreements Document)

5.9 SESSION

5.9.1 Introduction

(Refer to Stable Agreements Document)

5.9.2 Services

(Refer to Stable Agreements Document)

5.9.3 Protocol Agreements

5.9.3.1 Concatenation

(Refer to Stable Agreements Document)

5.9.3.2 Segmenting

(Refer to Stable Agreements Document)

5.9.3.3 Reuse of Transport Connection

(Refer to Stable Agreements Document)

5.9.3.4 Use of Transport Expedited Data

(Refer to Stable Agreements Document)

5.9.3.5 Use of Session Version Number

(Refer to Stable Agreements Document)

5.9.3.6 Receipt of Invalid SPDUs

(Refer to Stable Agreements Document)

5.9.3.7 Invalid SPM Intersections

(Refer to Stable Agreements Document)

5.9.3.8 S-Selectors

(Refer to Stable Agreements Document)

5.9.4 Connectionless

(Refer to Stable Agreements Document)

5.10 UNIVERSAL ASN.1 ENCODING RULES

5.10.1 TAGS

(Refer to Stable Agreements Document)

5.10.2 Definite Length

(Refer to Stable Agreements Document)

5.10.3 EXTERNAL

(Refer to Stable Agreements Document)

5.10.4 Integer

(Refer to Stable Agreements Document)

5.10.5 String Types

(Refer to Stable Agreements Document)

5.10.6 Bit String

(Refer to Stable Agreements Document)

5.11 CHARACTER SETS

5.11.0 Introduction

This International Standardized Profile is defined within the context of Functional Standardization, in accordance with the principles specified by ISO TR 10000, "Taxonomy Framework and Directory of Profiles". The context of Functional Standardization is one part of the overall field of Information Technology (IT) standardization activities, covering base standards, profiles, and registration mechanisms. A profile defines a combination of base standards that collectively perform a specific well-defined IT function. Profiles standardize the use of options and other variations in the base standards, and provide a basis for the development of uniform, internationally recognized system tests.

This International Standardized Profile was developed in close cooperation between the three International OSI Workshops: the NIST OSI Implementors Workshop (NIST OIW), the European Workshop for Open Systems (EWOS), and the AsiaOceania Workshop (AOW). The text is harmonized between these three Workshops and was ratified by the Workshops' plenary assemblies.

This International Standardized Profile contains an informative Annex A - Character Set Technology.

5.11.1 Scope

This International Standardized Profile describes Information Processing Character Set agreements covering character set usage in referencing Association Service Elements and OSI Applications. These agreements are based upon ISO Character Set International Standards and CCITT Character Set Recommendations. The informative Annex A summarizes the character set practices within referencing Association Service Elements and OSI Applications including all relevant encoding information drawn from the appropriate ISO Registers, ISO Standards, and CCITT Recommendations.

5.11.1.1 Recording Additional Character Sets

This International Standardized Profile does not prevent Application Service Elements from adding new graphic character sets or control function sets. When new character sets are to be added, however, they shall be recorded in this International Standardized Profile.

5.11.1.2 General Applicability of Character Sets

For the purpose of this International Standardized Profile when new character sets are to be added, efforts shall be made to obtain agreement on their uses among Application Service Elements so that they are generally applicable.

5.11.1.3 Minimum Number of Character Sets

The number of character sets supported will be kept to the minimum possible so as to maximize interoperability.

5.11.2 References

The following ISO Standards are referenced in this chapter:

[CCITT-T.61-1984, 1985 #22]

[DIS8859-7-1987, 1987 #30; ISO2022-1986, 1986 #1; ISO6429-1983, 1983 #9; ISO646-1983, 1983 #10; ISO6937/1-1983, 1983 #29; ISO6937/2-1983, 1983 #8; ISO8824-1987, 1987 #26; ISO8825-1987, 1987 #27; ISO8859-1:1987, 1987 #6; ISOREG, 1989 #2]

5.11.3 Definitions

5.11.3.1 character data:

Character data is defined to be graphic characters and control functions as defined by ISO 2022 and the appropriate International Standards.

5.11.3.2 composite graphic symbol

A composite graphic symbol is defined as a non-spacing diacritical in combination with an alphabetic as in ISO 6937.

5.11.4 Abbreviations

5.11.5 Position within the Taxonomy

<<The formal position of this International Standardized Profile within the taxonomy is currently unknown.>>

It may be referenced from the ISP for any application service element or OSI application.

5.11.6 Conformance

Implementations claiming conformance to this ISP must designate one or more of the Character Set Profiles defined herein.

Imaging of Graphic Characters is not required by this ISP. Imaging conformance may be defined in the specific Upper Layers Requirements section of the referencing ISP. If no imaging requirements are specified, then there are no conformance requirements.

5.11.6.1 Processed Character Data

Processed character data is character data which must be interpreted by the Application Service Element or OSI Application, for example, store and forward character data.

Senders of character data must not produce invalid character codes or invalid designating or invoking escape sequences.

5.11.6.1.1 Non-supported Character Sets

If an implementation receives a designating escape sequence for a character set that it is not able to interpret, then it shall regard that sequence as "invalid data". If possible, it will signal this error in a way that is appropriate to the protocol definition. For applications for which there is no protocol, then no error need be returned. It will not be required to interpret any following characters that are within that data item.

5.11.6.1.2 Reserved Character Codes

If an implementation receives a coded character that is specified in the standard to be "reserved for future standardization", it shall not be considered an error. Am imaging device shall indicate receipt of such a reserved character to the user in any convenient way, e.g. by making available a character that need not be distinguishable from one of the other characters specified in the standard.

If receivers reject or discard invalid character codes, an appropriate error code must be returned.

5.11.6.1.3 Validation of Character Codes

Character codes for which there is no entry in the code table in the defining standard are defined to be invalid character codes. An invalid escape sequence is any designating or invoking escape sequence which is not defined in these agreements.

Implementations must conform to the following statement.

- Originators of data shall not produce invalid character codes or invalid designating or invoking escape sequences.

5.11.6.2 Unprocessed Character Data

Unprocessed character data is character data which is not interpreted by the Application Service Element or OSI Application, for example, character matching.

5.11.6.2.1 Validation of Character Codes

Character codes for which there is no entry in the code table in the defining standard are defined to be invalid character codes. An invalid escape sequence is any designating or invoking escape sequence which is not defined in these agreements.

Implementations must conform to the following statements.

- Receivers need not validate character codes or designating or invoking escape sequences.

- Senders who do not originate data need not validate character codes.

5.11.7 General Agreements

The agreements recorded in this section cover all character set usage except where explicitly noted to the contrary. Additional agreements specific to individual character sets are recorded in the individual character set profiles.

5.11.7.1 Encoding

The following agreements cover various aspects of character encoding.

5.11.7.1.1 Uverprint, Composite Characters

A composite graphic symbol is considered as one character for purposes of comparison and character string length computation.

With the exception of composite graphic symbols, sequences of graphic characters and control functions which would result in the presentation of two or more graphic characters in a single character position shall not be used. So for example, the sequence "a BACKSPACE ˜" must be interpreted as three characters rather than as the single character ã.

5.11.7.1.2 Code Extension Facilities for GeneralString and GraphicString

This section constitutes the prior agreement on code extension required by ISO 2022.

For ASN.1 GeneralString and GraphicString types, the assumed extension facilities are as though the following escape sequences from ISO 2022 have been applied: ESC 2/0 4/3, ESC 2/0 4/9, and ESC 2/0 5/10. These sequences indicate:

- 8-bit environment;
- the G0, and G1 graphic sets shall be used;
- the designating escape sequences also invoke the G0 and G1 sets into columns 02 to 07 and 10 to 15 respectively;
- no locking shift functions shall be used;
- the graphic character sets may comprise 94 and/or 96 characters,
- a G2 set shall be used; and,
- characters from G2 may be accessed by use of the single-shift 2 control function.

Designating ESCAPE sequences in a data stream are permitted. No Announcers of extension facilities may be used within these ASN.1 string types.

5.11.7.1.3 Initial Conditions fur TeletexString

For TeletexString (T61String) the initial condition is described in CCITT T.61 Annex A, Clause A.2.

5.11.7.2 Comparisons

This section contains agreements concerning comparison of characters during processing.

5.11.7.2.1 Matching Characters

A character submitted for matching with another character does not have to be drawn from the same coded character set. However, the match is restricted to characters taken from any pair of coded character sets for which equality or inequality is defined. The identifications of such pairs of coded character sets are shown in the following list. The result of comparing characters from a pair of different coded character sets not in this list is undefined.

(ISO 646,	ISO 6937-2)
(ISO 646,	ISO 8859-1)
(ISO 6937-2,	ISO 8859-1)

Character matching is defined for characters, not their coded representations. The character must take into account any code extension techniques. For example, the character named "SMALL LETTER a WITH DIAERESIS" of ISO 8859 must match the character named "small a with diaeresis or umlaut mark" of ISO 6937 even though the former character is encoded in a single octet and the latter in two octets.

Two characters are said to be equal if, and only if, their names are identical. The names are recorded in the registration of the character sets in the **International Register of Coded Character Sets to be used with Escape Sequences** and not the character set International Standard or Recommendation.

In the case of ISO 6937-2 the names of the composite graphic symbols are specified in the standard itself. However in the present edition there are some systematic differences between the naming conventions used in the standard and those used in the ISO Character Set Register as shown below:

ISO 6937 name:	capital A with acute accent
ISO Register Name:	CAPITAL LETTER A WITH ACUTE ACCENT

In this case, two characters are equal if, and only if, their names differ only by the inclusion of the word LETTER in the ISO Register Name. For those characters whose names do not follow this convention, the following list defines the match:

ISO 6937 Name	ISO Register Name

<<Editor's Note: to be filled in>

If a character set registration does not provide character names then matching will be defined by exact matching on an octet by octet basis.

<<Editor's Note: The problem of matching Oriental language character sets is for further study.>>

In comparing strings all control functions except code designation and invocation extension facilities shall be ignored. SPACE is treated as a blank graphic character in such comparisons.

In comparing strings when a character code is encountered for which there is no assigned graphical representation, matching will be defined by exact matching on an octet by octet basis.

5.11.7.2.2 Caseignore Comparisons

In character comparisons in which case is ignored, the matching rules of clause 7.2.1 are relaxed in that the characters are equal if their names as defined in clause 7.2.1 differ only by one name having SMALL where the other name has CAPITAL.

5.11.7.2.3 Ordering and Comparing Characters

An agreement on comparison, other than equality or inequality, between characters requires a definition of a collating sequence. This document contains no such agreements.

The collating sequence of letters, accented letters and other graphic symbols is not currently defined in any International Standard or Recommendation.

Preferred collating sequences might vary between countries.

5.11.7.2.4 Comparing Encoded ASN.1 Character Strings

In this section a character string is considered to be a sequence of characters some of which may be composed of multiple bytes depending upon the character set encodings which are specified. Comparing two character strings gives the same result independent of each character string's encoding, for example, the comparison is independent of the Basic Encoding Rules for ASN.1:

- as constructed or primitive form, or,
- as definite or indefinite length form.

5.11.8 Character Set Profiles

A Character Set Profile summarizes implementation agreements specific to a particular character set. Character Set Profiles are identified in the following manner:

CSn-m

where:

CS means Character Set

n = 1 designates a profile for a graphic character set

n = 2 designates a profile for a control function set

m is a number uniquely identifying the Character Set Profile.

The values of n and m are defined in this agreement. Names of Character Set Profiles are also defined in this International Standardized Profile.

This section covers agreements about Character Set Standards and Recommendations including:

- subrepertoires supported,
- standardized options selected,
- component character sets and their registrations in the International

Register of Coded Character Sets to be used with Escape Sequences where there is a choice to be made, or the standard does not specify it, and,

- the designation of component character sets within the ISO 2022 Code Extension Model where there is a choice to be made.

The General Agreements of the preceding section apply to each of these Character Set Profiles.

5.11.8.1 CS1-1 ISO 646 Graphic Character Set

5.11.8.1.1 Base Standard

International Standard 646 - 1983, *Information Processing — ISO 7-bit coded character set for information interchange.*

5.11.8.1.2 Subrepertoire or Version

International Reference Version

5.11.8.1.3 Standard Options Selected

Composite graphic symbols are covered by General Agreements.

5.11.8.1.4 Character Set Components and Designated Position

IRV of ISO 646 number 2 in G0

<<Editor's Note: This will change to number 6.>>

Space is in 2/0

5.11.8.1.5 Other Agreements

None.

5.11.8.2 CS1-2 JIS X0208

<<Editor's Note: to be defined.>>

5.11.8.3 CS1-3 CCITT Recommendation T.61 Graphic Character Sets Basic Teletex Profiles

5.11.8.3.1 Base Standard

CCITT Recommendation T.61 - 1985, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

5.11.8.3.2 Subrepertoire or Version

None

5.11.8.3.3 Standard Options Selected

None

5.11.8.3.4 Character Set Components and Designated Position

Teletex Primary Graphic Set 102 in G0

Teletex Supplementary Graphic Set 103 in G2

SPACE in 2/0

5.11.8.3.5 Other Agreements

Support for CCITT Recommendation T.61 as an ASN.1 GeneralString is outside of this International Standardized Profile.

Support of the graphic set components of T.61 as an ASN.1 GraphicString is outside the scope of this International Standardized Profile.

Use of CCITT Recommendation T.61 except where mandated by standards is outside the scope of this International Standardized Profile. Exceptions to this rule for specific Application Service Element protocol elements must be documented by the referencing Application Service Elements or OSI Applications.

5.11.8.4 CS1-4 ISO 8859-1 Latin Alphabet No. 1

5.11.8.4.1 Base Standard

International Standard 8859-1 - 1987, *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*

5.11.8.4.2 Subrepertoire or Version

Not applicable.

5.11.8.4.3 Standard Options Selected

Not applicable.

5.11.8.4.4 Character Set Components and Designated Position

ASCII Graphic Character Set number 6 in G0

Right hand part of Latin Alphabet No. 1 number 100 in G1

5.11.8.4.5 Other Agreements

None.

5.11.8.5 CS1-5 ISO 6937-2 Coded Character Sets for Text Communication

5.11.8.5.1 Base Standard

International Standard 6937/2 - 1983, *Information Processing — Coded character sets for text communication — Part 2: Latin alphabetic and non-alphabetic graphic characters.*

5.11.8.5.2 Subrepertoire or Version

Full number 0

Minimum number 1

Teletex number 3

Western European Data Processing number 9

5.11.8.5.3 Standard Options Selected

Not applicable

5.11.8.5.4 Character Set Components and Designated Position

IRV of ISO 646 number 2 in G0

> *<<Editor's Note: This will change to number 6.>>*

Supplementary set of Latin Text Processing number 142 in G2

5.11.8.5.5 Other Agreements

For subrepertoires 2 and 5, the supplementary set may be omitted at the discretion of the sending application.

5.11.8.6 CS1-6 ISO 8859/7 Greek Supplementary Set

> *<<Editor's Note: to be defined.>>*

5.11.8.7 CS1-7 CCITT Recommendation T.61 Graphic Character Sets Basic Teletex Profiles (1984)

5.11.8.7.1 Base Standard

CCITT Recommendation T.61 - 1981, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

5.11.8.7.2 Subrepertoire or Version

None

5.11.8.7.3 Standard Options Selected

None

5.11.8.7.4 Character Set Components and Designated Position

Teletex Primary Graphic Set 102 in G0

Teletex Supplementary Graphic Set 103 in G2

SPACE in 2/0

5.11.8.7.5 Other Agreements

Support for CCITT Recommendation T.61 as an ASN.1 GeneralString is outside of this International Standardized Profile.

Support of the graphic set components of T.61 as an ASN.1 GraphicString is outside the scope of this International Standardized Profile.

Use of CCITT Recommendation T.61 except where mandated by standards is outside the scope of this International Standardized Profile. Exceptions to this rule for specific Application Service Element protocol elements must be documented in the referencing Application Service Elements or OSI Applications.

This profile is intended for use with the X.400-1984 implementation agreements only.

5.11.8.8 CS 1-8 CCITT Recommendation T.61 Graphic Character Sets

> *<<Editor's Note: to be defined.>>*

5.11.8.9 Korean National Character Set

<<*Editor's Note: to be defined.*>>

5.11.8.10 CS2-1 ISO 646 C0 Control Functions

5.11.8.10.1 Base Standard

International Standard 646 - 1983, *Information Processing — ISO 7-bit coded character set for information interchange.*

5.11.8.10.2 Subrepertoire or Version

None.

5.11.8.10.3 Standard Options Selected

None.

5.11.8.10.4 Character Set Components and Designated Position

ISO 646 C0 Set number 1 in C0

DELETE in 7/15

5.11.8.10.5 Other Agreements

When a single format effector for vertical (or horizontal) movement is optionally permitted to effect a combined vertical and horizontal movement, implementations shall not use this single format effector for effecting the combined vertical and horizontal movement.

5.11.8.11 CS2-2 ISO 6429 Additional Control Functions

5.11.8.11.1 Base Standard

International Standard 6429 - 1983, *Information Processing — ISO 7-bit and 8-bit coded character sets — Additional control functions for character-imaging devices.*

5.11.8.11.2 Subrepertoire or Version

None.

5.11.8.11.3 Standard Options Selected

None.

5.11.8.11.4 Character Set Components and Designated Position

C1 Control Set of ISO 6429-1983 number 77 in C1

5.11.8.11.5 Other Agreements

None.

5.11.8.12 CS2-3 CCITT **Recommendation T.61 Control Sets**

5.11.8.12.1 Base Standard

CCITT Recommendation T.61 - 1985, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

5.11.8.12.2 Subrepertoire or Version

None.

5.11.8.12.3 Standard Options Selected

Teletex optional repertoire of control functions is not selected.

5.11.8.12.4 Character Set Components and Designated Position

Teletex Primary Set of Control Functions number 106 in C0

Teletex Supplementary Set of Control Functions number 107 in C1

5.11.8.12.5 Other Agreements

None.

5.11.8.13 CS2-4 CCITT **Recommendation T.61 Control Sets (1984)**

5.11.8.13.1 Base Standard

CCITT Recommendation T.61 - 1981, *Character Repertoire and Coded Character Sets for the International Teletex Service.*

5.11.8.13.2 Subrepertoire or Version

None.

5.11.8.13.3 Standard Options Selected

Teletex optional repertoire of control functions is not selected.

5.11.8.13.4 Character Set Components and Designated Position

Teletex Primary Set of Control Functions number 106 in C0

Teletex Supplementary Set of Control Functions number 107 in C1

5.11.8.13.5 Other Agreements

None.

5.11.9 ANNEX

Character Set Technology

(This Annex does not form part of these agreements.)

5.11.9.1 Introduction

This Annex presents information from Information Processing Character Set Standards which is relevant to the implementation of OSI Services. The intent is to collect into one place the most relevant information for implementors from character set standards specified in OSI and OSI related standards.

5.11.9.2 Scope

Material in this Annex is drawn from ISO and CCITT Character Set standards and Recommendations. Topics covered include Character Set Extension Techniques and Character Set Encodings. ASN.1 Basic Encoding Rules are reviewed also. Rationale for the implementation agreements in the ISP is provided where appropriate.

5.11.9.3 Field of Application

This annex covers character set information for ASN.1 Basic Encoding Rules as used by OSI services. It also includes information pertaining to OSI Interchange Formats such as Office Document Architecture.

5.11.9.4 Character Set Standards

The following character set standards have some relevance to this material.

[CCITT-T.100-1984, 1985 #23; CCITT-T.50-1984, 1985 #20; CCITT-T.51-1984, 1985 #21; CCITT-T.61-1984, 1985 #22]

[ISO2022-1986, 1986 #1; ISO2375-1985, 1985 #7; ISO4873-1986, 1986 #4; ISO6429-1983, 1983 #9; ISO646-1983, 1983 #10; ISO6937/2-1983, 1983 #8; ISO7350-1984, 1984 #5; ISO8824-1987, 1987 #26; ISO8825-1987, 1987 #27; ISO8859-1:1987, 1987 #6; ISOREG, 1989 #2]

5.11.9.5 Introduction to Character Set Standards

A brief introduction to reading a character set standard is presented here for the uninitiated. Most of the character set standards described in this Annex use the term "bit combinations" to refer to the ordered string of bits which compose a character. Most implementations of these standards allocate an 8-bit byte to a character and consequently tend to intermix the notions of bytes and characters. In the OSI environment, 8-bit bit combinations are normally referred to as "octets".

A character set standard generally presents its character encodings in a table composed of 16 rows and 8 or 16 columns depending on whether a 7-bit or an 8-bit character set is being defined. A given character code is generally referenced by naming its column and then its row. Thus in ISO 646 the capital letter A is referred to as 4/1. Some standards precede single digits with a zero so that in ISO 8859/1 the capital letter A is referred to as 04/01. This positional notation is especially important in the consideration of the code extension techniques. Code extension techniques describe characters in terms of their position only, without regard for any possible previously assigned interpretations.

5.11.9.6 Definitions

The following definitions drawn from relevant character set standards are provided to assist in understanding the material in this annex. These definitions were drawn from International Standards which were current at the time of drafting this document. Any conflict between these definitions and those of the relevant International Standards shall be resolved by using the definition in the International Standard.

bit combination: An ordered set of bits that represents a character or is used as a part of the representation of a character.

byte: A bit string that is operated upon as a unit and the size of which is independent of redundancy or framing techniques.

character: A member of a set of elements used for the organization, control or representation of data.

code extension: The techniques for the encoding of characters that are not included in the character set of a given code.

control character: A control function the coded representation of which consists of a single bit combination.

control function: An action that affects the recording, processing, transmission or interpretation of data and that has a coded representation consisting of one or more bit combinations.

graphic character: A character, other than a control function, that has a visual representation normally handwritten, printed or displayed.

5.11.9.7 ISO 2022 Information Processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques

This International Standard was originally written to establish extension techniques for the 7-bit codes of ISO 646. It has been revised twice so that it now also provides the basic framework for an 8-bit code family which is compatible with the 7-bit codes. The four interrelated clauses cover

- the extension of the 7-bit code remaining in a 7-bit environment;
- the structure of a family of 8-bit codes;
- the extension of an 8-bit code remaining in an 8-bit environment;
- the relationship between the 7-bit code and an 8-bit code.

The middle two clauses are of special relevance to this document although portions of the others should be read and understood in order to set the context for the relevant material.

Some underlying assumptions from the standard are recorded here in order to understand the context of these agreements. Clause 2 notes that code extension techniques are designed to be used for data to be processed serially in a forward direction.

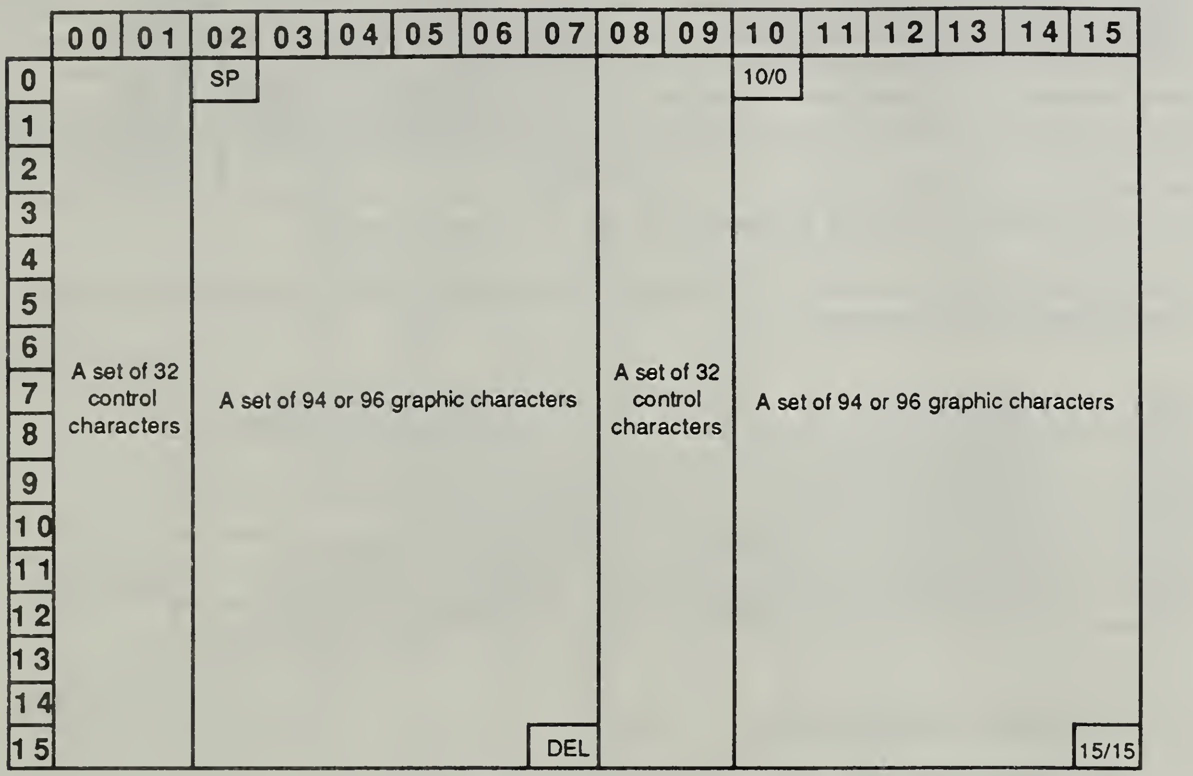
5.11.9.7.1 Structure of a Family of 8-bit codes

Clause 7 of the standard describes a family of 8-bit codes obtained from the 7-bit set. The family of 8-bit codes is obtained by the addition of one bit to each of the bit combinations of the 7-bit code producing a set of 256 8-bit combinations. The characters of the 7-bit code are assigned to the 128 bit combinations for which the eighth bit is set to ZERO. The 128 additional bit combinations for which the eighth bit is set to ONE are available for assignment. The 8-bit code table of clause 7.1 is a 16 by 16 array of columns numbered 00 to 15 and rows numbered 0 to 15.

Columns 08 and 09 are provided for control characters and columns 10 to 15 for graphic characters.

The following figure shows the basic code structure for 8-bit character codes. This structure is followed by the standards described in this annex.

8-bit Code Structure


00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SP
10/0
A set of 32 control characters
A set of 94 or 96 graphic characters
A set of 32 control characters
A set of 94 or 96 graphic characters
DEL
15/15

The family concept is described in clause 7.2 as

a) a set of 32 additional control characters can be selected for columns 08 and 09;

b) a set of 94 or 96 additional graphic characters can be selected for columns 10 to 15. If a set of 94 graphic characters is invoked in columns 10 to 15, positions 10/0 and 15/15 shall not be used.

Three control functions were provided by ISO 646 for purposes of code extension. ISO 2022 uses these three and adds 7 more for use in the 8-bit environment. For reference purposes the corresponding characters from the 7-bit environment are shown also. The following table shows these control functions.

7-bit Name	Abbreviation	8-bit Name	Abbreviation
ESCAPE	ESC	ESCAPE	ESC
SHIFT-OUT	SO	LOCKING-SHIFT ZERO	LS0
SHIFT-IN	SI	LOCKING-SHIFT ONE	LS1
LOCKING-SHIFT TWO	LS2	LOCKING-SHIFT TWO	LS2
LOCKING-SHIFT THREE	LS3	LOCKING-SHIFT THREE	LS3
SINGLE-SHIFT TWO	SS2	SINGLE-SHIFT TWO	SS2
SINGLE-SHIFT THREE	SS3	SINGLE-SHIFT THREE	SS3
		LOCKING-SHIFT ONE RIGHT	LS1R
		LOCKING-SHIFT TWO RIGHT	LS2R
		LOCKING-SHIFT THREE RIGHT	LS3R

5.11.9.7.2 Elements of Code Extension in an 8-bit Environment

The elements of code extension in an 8-bit environment are shown in the following table taken from Clause 8.1 of the standard:

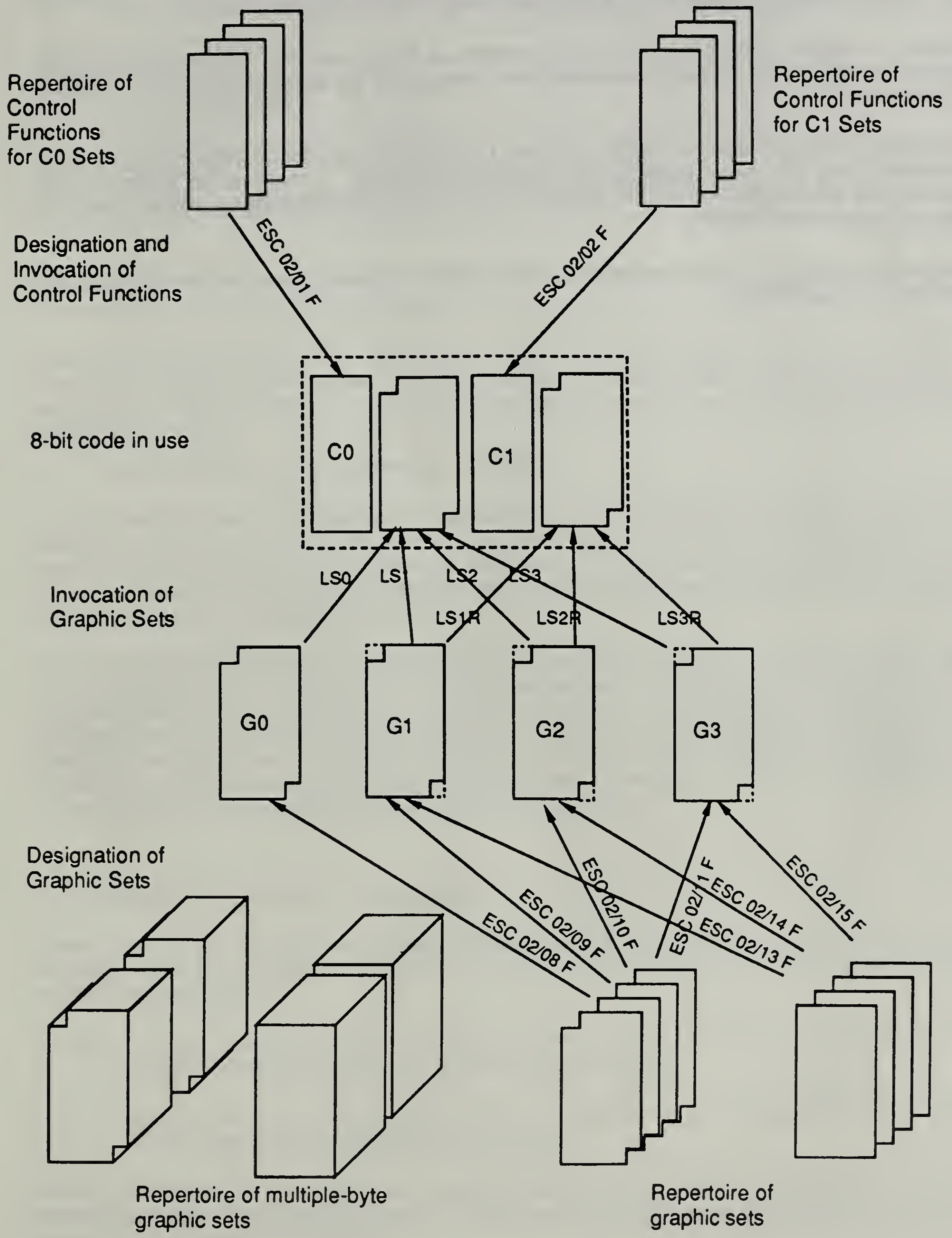
Set	Description	Columns occupied
C0	32 control characters	00 to 01
C1	32 control characters	08 to 09
G0	94 graphic characters	02 to 07
G1	94 or 96 graphic characters	02 to 07 or 10 to 15
G2	94 or 96 graphic characters	02 to 07 or 10 to 15
G3	94 or 96 graphic characters	02 to 07 or 10 to 15

5.11.9.7.3 Multiple Character Sets

<<Describe multi-level designation and invocation here.>>

The standard defines a graphic character set extension strategy in which a designating escape sequence is used to select up to four graphic character sets from the International Character Set Register. An invocation sequence is then used to select up to two graphic sets from the designated sets for concise access to the characters. The following figure shows the technique for the 8-bit environment.

Code Extension in an 8-bit Environment


Repertoire of Control Functions for C0 Sets
Repertoire of Control Functions for C1 Sets
Designation and Invocation of Control Functions
ESC 02/01 F
ESC 02/02 F
8-bit code in use
C0
C1
Invocation of Graphic Sets
LS0
LS1
LS2
LS3
LS1R
LS2R
LS3R
G0
G1
G2
G3
Designation of Graphic Sets
ESC 02/08 F
ESC 02/09 F
ESC 02/10 F
ESC 02/11 F
ESC 02/13 F
ESC 02/14 F
ESC 02/15 F
Repertoire of multiple-byte graphic sets
Repertoire of graphic sets

The standard defines two terms for use in describing code extension practices: to designate and to invoke. They are defined as follows:

to designate: To identify a set of characters that are to be represented, in some cases immediately and in others on the occurrence of a further control function, in a prescribed manner.

to invoke: To cause a designated set of characters to be represented by the prescribed bit combinations whenever those bit combinations occur, until an appropriate code extension function occurs.

Designation of a character set is usually achieved by employing an escape sequence defined by the standard along with values assigned by a registration authority. In many cases, designation of a character set also implies invocation. In other cases a character set must be explicitly invoked usually by using a shift function.

The following table defines the use of the locking shift functions in an 8-bit environment for extension of the graphic set.

Function	Abbreviation	Set Invoked	Columns affected
LOCKING-SHIFT ZERO	LS0	G0	02 to 07
LOCKING-SHIFT ONE	LS1	G1	02 to 07
LOCKING-SHIFT ONE RIGHT	LS1R	G1	10 to 15
LOCKING-SHIFT TWO	LS2	G2	02 to 07
LOCKING-SHIFT TWO RIGHT	LS2R	G2	10 to 15
LOCKING-SHIFT THREE	LS3	G3	02 to 07
LOCKING-SHIFT THREE RIGHT	LS3R	G3	10 to 15

The meanings of control characters in columns 00, 01, 08 and 09 shall not be affected by the occurrence of these locking shift functions.

Clause 6.4 states that at the beginning of any information interchange, except where interchanging parties have agreed otherwise, all designations shall be defined by the use of appropriate escape sequences, and the shift status shall be defined by the use of the appropriate locking shift functions.

5.11.9.7.4 Announcement of Extension Facilities

A code extension facility consists of the elements of code extension employed as well as the means by which these elements are designated and invoked. Thus the control function sets, the graphic character sets, and the character shifting codes must be specified. Specification of control function sets and graphic character sets also specifies the designation and invocation sequences required to use their codes.

Clause 9 of ISO 2022 describes how the various extension facilities are to be made known. If an announcement is to be embedded in the interchanged information, the form is described. The announcement may be omitted by agreement between the interchanging parties. Some restrictions are imposed on the defined announcer sequences. For example the sequence ESC 02/00 04/03 specifies that 1) the G0 and G1 sets shall be used in an 8-bit environment only, 2) the designating escape sequences also invoke the G0 and G1 sets into columns 02 to 07 and 10 to 15, respectively, and 3) no locking shift functions shall be used.

5.11.9.7.5 Composite Graphic Characters

Clause 6.1.8 of the standard addresses methods for the representation of additional graphic characters by the combination of two or more graphic characters in the same position. Two methods are provided for:

a) graphic characters having implicit forward motion (spacing characters) used in conjunction with BACKSPACE or CARRIAGE RETURN;

b) graphic characters having no implicit forward motion (non-spacing characters) used in combination with spacing graphic characters.

Method b allows for the specification of characters with diacritical marks. The technique is known colloquially as the "dead key" approach. A non-spacing accent grave character is immediately followed by the character it modifies.

5.11.9.7.6 International Register of Coded Character Sets to be used with Escape Sequences

ISO 2375 specifies procedures to be used to assign meanings to the final bit combinations of escape sequences defined in ISO 2022. The International Register of Coded Character Sets to be used with escape sequences is the document which records these assignments. The current International Registration Authority for ISO 2375 is the European Computer Manufacturers Association (ECMA).

5.11.9.8 Character Sets

Several character set standards are described here. The standards chosen for description are each used by one or more known OSI applications. The usage of these standards is summarized in tabular form.

5.11.9.8.1 ISO 646 *7-bit coded character set for information processing interchange* and CCITT Recommendation T.50 *International Alphabet No. 5*

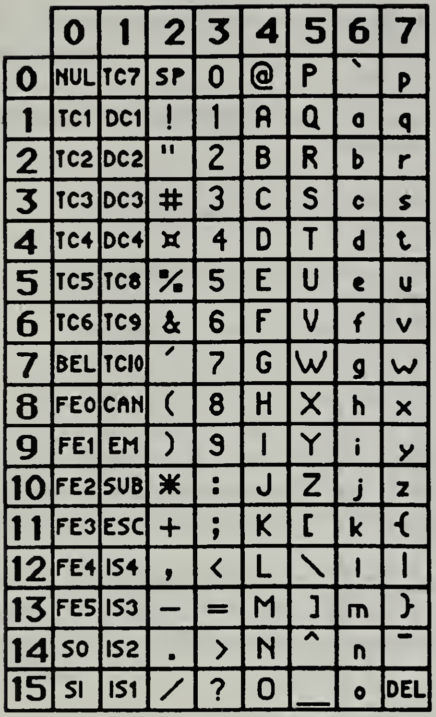
This International Standard specifies a set of 128 characters with their coded representation. The 128 bit combinations of the 7-bit code represent control characters and graphic characters. The allocation of characters to bit combinations is based on the following principles:

- the bit combinations 0/0 to 1/15 represent 32 control characters;
- the bit combination 2/0 represents the character SPACE, which is interpreted as both a control character and a graphic character;
- the bit combinations 2/1 to 7/14 represent up to 94 graphic characters;
- the bit combination 7/15 represents the control character DELETE.

The 7-bit code table consists of 128 positions arranged in 8 columns and 16 rows. The columns are numbered from 0 to 7, and the rows are numbered 0 to 15.

Most of these characters are mandatory and unchangeable, but provision is made for some flexibility to accommodate national and other requirements. The standard provides guidance on how to exercise the options offered in order to define specific national versions and application-oriented versions. It further specifies an International Reference Version in which all options have been exercised.

<<Editor's Note: A revision of ISO 646 which has achieved DP status revises this table.>>



	0	1	2	3	4	5	6	7
0	NUL	TC7	SP	0	@	P	`	p
1	TC1	DC1	!	1	A	Q	a	q
2	TC2	DC2	"	2	B	R	b	r
3	TC3	DC3	#	3	C	S	c	s
4	TC4	DC4	¤	4	D	T	d	t
5	TC5	TC8	%	5	E	U	e	u
6	TC6	TC9	&	6	F	V	f	v
7	BEL	TC10	'	7	G	W	g	w
8	FE0	CAN	(	8	H	X	h	x
9	FE1	EM	)	9	I	Y	i	y
10	FE2	SUB	*	:	J	Z	j	z
11	FE3	ESC	+	;	K	[	k	{
12	FE4	IS4	,	<	L	\	l	|
13	FE5	IS3	–	=	M	]	m	}
14	SO	IS2	.	>	N	^	n	‾
15	SI	IS1	/	?	O	_	o	DEL

ISO 646 International Reference Version

5.11.9.8.2 ISO 8859 *Information Processing — 8-bit single-byte coded character sets*

This International Standard is a multiple part standard. Each part specifies a set of up to 191 graphic characters and the coded representation of each of these characters by means of a single 8-bit byte. The use of control functions for the coded representation of composite characters is prohibited. Each set is intended for a group of languages. Part 1 of ISO 8859 specifies a set of 191 graphic characters identified as Latin alphabet No. 1. This set of graphic characters is suitable for use in a version of an 8-bit code according to ISO 2022.

The standard specifically notes that it is not intended for use with CCITT defined Telematic services. If information coded according to ISO 8859 is to be transferred to such services, it will have to conform at the coding interface to their requirements.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð
1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
4			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
5			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
6			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
8			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
9			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
11			+	;	K	[	k	{			«	»	Ë	Û	ë	û
12			,	<	L	\	l	¦			¬	¼	Ì	Ü	ì	ü
13			–	=	M	]	m	}			SHY	½	Í	Ý	í	ý
14			.	>	N	^	n	˜			®	¾	Î	Þ	î	þ
15			/	?	O	_	o	DEL			¯	¿	Ï	ß	ï	ÿ

ISO 8859/1 - 1987 Latin Alphabet No. 1

5.11.9.8.3 ISO 6937 *Information Processing — Coded Character Sets for Text Communication*

This International Standard specifies repertoires of graphic characters and control functions, and their coded representation for use in text communication. This International Standard consists, at present, of two parts, as follows:

- ISO 6937/1, General Introduction.
- ISO 6937/2, Latin Alphabetic and non-alphabetic graphic characters.

The specifications are based on the 7-bit coded character set specified in ISO 646, the 7-bit and 8-bit code extension techniques of ISO 2022, and the definitions of additional control functions given in ISO 6429.

ISO 6937 was developed in parallel with CCITT Recommendations which in the standard are referred to as S.61 and S.100. These CCITT Recommendations were moved to a new section in 1984 and were renumbered T.61 and T.100. This 1984 designation is being carried forward in the 1988 CCITT Recommendations.

5.11.9.8.3.1 ISO 6937/1 *Information Processing — Coded Character Sets for Text Communication — Part 1: General Introduction*

Annex A of this International Standard describes a method of identification of graphic characters and control functions which is used in other parts of the standard to define the characters of the standard.

5.11.9.8.3.2 ISO 6937/2 *Information Processing — Coded Character Sets for Text Communication — Part 2: Latin Alphabetic and Non-alphabetic Graphic Characters*

This part of the standard

a) defines a repertoire of Latin alphabetic and non-alphabetic characters for the communication of text in European languages;

b) specifies coded representations for the graphic characters;

c) specifies rules for the definition and use of graphic character subrepertoires.

A graphic subrepertoire is a subset of the defined character repertoire. Because the number of characters defined by this standard is so large, this subsetting facility allows for the use of well defined subsets of the characters available. Rules for the definition of subrepertoires are defined in clause 5. The procedure for registration of subrepertoires is given in ISO 7350. Three standard subrepertoires are defined in Annex A of the standard.

Graphic characters which represent accented letters and umlauts are specified using a two byte sequence composed of the diacritical character immediately followed by the character modified. The allowable combinations are carefully defined in the standard and only these combinations are permitted.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			SP	0	@	P	`	p			NBSP	°		―	Ω	ĸ
1			!	1	A	Q	a	q			¡	±	`	¹	Æ	æ
2			"	2	B	R	b	r			¢	²	´	®	Đ	đ
3			#	3	C	S	c	s			£	³	^	©	ª	ð
4			¤	4	D	T	d	t			$	×	~	™	Ħ	ħ
5			%	5	E	U	e	u			¥	µ	¯	♪		ı
6			&	6	F	V	f	v				¶	˘	¬	Ĳ	ĳ
7			'	7	G	W	g	w			§	·	˙	¦	Ŀ	ŀ
8			(	8	H	X	h	x				÷	¨		Ł	ł
9			)	9	I	Y	i	y			‘	’			Ø	ø
10			*	:	J	Z	j	z			“	”	˚		Œ	œ
11			+	;	K	[	k	{			«	»	¸		º	ß
12			,	<	L	\	l	|			←	¼	_	⅛	Þ	þ
13			-	=	M	]	m	}			↑	½	˝	⅜	Ŧ	ŧ
14			.	>	N	^	n	‾			→	¾	˛	⅝	Ŋ	ŋ
15			/	?	O	_	o	DEL			↓	¿	ˇ	⅞	ŉ	SHY

ISO 6937-2 Latin Alphabetic and non-Alphabetic Characters

5.11.9.8.4 CCITT Recommendation T.51 *Coded Character Sets for Telematic Services*

This Recommendation specifies a primary set and a supplementary set of graphic characters which are to be the respective supersets of various primary and supplementary character sets to be used in various telematic services. The Recommendation also describes those code extension mechanisms which are relevant to existing telematic services.

5.11.9.8.5 CCITT Recommendation T.61 *Character Repertoire and Coded Character Sets for the International Teletex Service*

This Recommendation contains detailed definitions of the repertoires of graphic characters and control functions to be used in the basic International Teletex service, and their coded representations for communication.

5.11.9.9 ASN.1 Character String Types

Character String Types are sequences of zero, one or more characters from some specified character set. ISO 8824 defines 8 such types: NumericString, PrintableString, TeletexString (T61String), VideotexString, VisibleString (ISO646String), IA5String, GraphicString, GeneralString.

5.11.9.9.1 Universal Class Numbers and Registration Numbers

The type of each character string is identified by a Universal Class number. Universal Class numbers are assigned by ISO 8824. No other standard or private user may define these numbers. The character sets associated with each type are identified by the ISO Character Set Registration Numbers as shown in the following table:

Name of Character String Type	Universal Class Number	ISO Character Set Registration Numbers
NumericString	18	Not Registered
PrintableString	19	Not Registered
TeletexString (T61String)	20	87, 102, 103, 106, 107 + SPACE + DELETE
VideotexString	21	1, 72, 73, 102, 108, 128, 129 + SPACE + DELETE
VisibleString (ISO646String)	26	2 + SPACE
IA5String	22	1, 2 + SPACE + DELETE
GraphicString	25	All G sets + SPACE
GeneralString	27	All G sets and all C sets + SPACE + DELETE

NumericString and PrintableString do not have Registration Numbers assigned to them since their character sets are defined in table 4 and 5 respectively of ISO 8824.

5.11.9.9.2 Initial States

Some character string types allow multiple character sets through code extension techniques. For these types, at the beginning of each string there are initial default character sets to be designated in G0 and/or C0 and/or C1 and for each character set there is an assumed escape sequence. The following table drawn from ISO 8825 describes these initial states.

Name of Character String Type	Initial G0 (Reg. No.)	Initial C0 (Reg. No.)	Initial C1 (Reg. No.)	Initial ESC Seq and Lock Shift Function	Code Extension
NumericString	2	None	None	ESC 2/8 4/0 LS0	No
PrintableString	2	None	None	ESC 2/8 4/0 LS0	No

TeletexString (T61String)	102	106	107	ESC 2/8 4/0 LS0 ESC 2/1 4/5 ESC 2/2 4/8	Yes
VideotexString	102	1	73	ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1	Yes
VisibleString (ISO646String)	2	None	None	ESC 2/8 4/0 LS0	No
IA5String	2	1	None	ESC 2/8 4/0 LS0 ESC 2/1 4/0	No
GraphicString	2	None	None	ESC 2/8 4/0 LS0	Yes
GeneralString	2	1	None	ESC 2/1 4/0 LS0 ESC 2/1 4/0	Yes

For example, VideotexString initial G0 set is Primary Teletex Graphic Set (ISO Registration Number 102), initial C0 set is ISO 646 C0 set (ISO Registration Number 1), initial C1 set is Attribute Control Set for Videotex (ISO Registration Number 73), initial escape sequence and locking shift function is ESC 2/8 7/5 LS0, and ESC 2/2 4/1, and code extensions are permitted.

5.11.9.10 Use of ASN.1 OctetString as a Character String

<<Editor's Note: Add a description of ODA treatment of character sets.>>

5.11.9.11 Escape Sequences for Character Set Designation

This information is extracted from the ISO Register. In some cases, the defaults supplied by ASN.1 make the use of these escape sequences unnecessary. In some cases, this information is carried by application protocol elements.

Graphic Set Designation

Set No.	G0	G1	G2	Name
2	ESC 2/8 4/0			ISO 646 IRV
6	ESC 2/8 4/2			ISO 646 USA
87	ESC 2/4 4/2	ESC 2/4 2/9 4/2		JIS X0207
100		ESC 2/13 4/1	ESC 2/14 4/1	ISO 8859/1 Right Hand Part
102	ESC 2/8 7/5			CCITT T.61 Primary
103			ESC 2/10 7/6	CCITT T.61 Supp
126		ESC 2/13 4/6		ISO 8859/7 Greek
142			ESC 2/14 4/10	ISO 6937/2 Ad1 Supp

Control Set Designation

Set No.	C0	C1	Name
1	ESC 2/1 4/0		ISO 646 C0
106	ESC 2/1 4/5		CCITT T.61 Primary
107		ESC 2/2 4/8	CCITT T.61 Suppl.

5.12 CONFORMANCE

(Refer to Stable Agreements Document)

5.12.1 Specific ASE Requirements

(Refer to Stable Agreements Document)

5.12.1.1 FTAM

5.12.1.1.1 Phase 2

(Refer to Stable Agreements Document)

5.12.1.2 MHS

5.12.1.2.1 Phase 1 (1984 X.400)

(Refer to Stable Agreements Document)

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

(Refer to Stable Agreements Document)

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

(Refer to Stable Agreements Document)

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

(Refer to Stable Agreements Document)

5.12.1.3 DS

5.12.1.3.1 Phase 1

(Refer to Stable Agreements Document)

5.12.1.4 Virtual Terminal

5.12.1.4.1 Phase 1a

(Refer to Stable Agreements Document)

5.12.1.4.2 Phase 1b

(Refer to Stable Agreements Document)

5.13 APPENDIX A: RECOMMENDED PRACTICES

(Refer to Stable Agreements Document)

5.14 APPENDIX B: OBJECT IDENTIFIER REGISTER

5.14.1 Register Index

(Refer to Stable Agreements Document)

5.14.2 Object Identifier Descriptions

(Refer to Stable Agreements Document)

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

For current Registration Authority information for Workshop--Defined Objects, consult the aligned chapter of Version 3, Edition 1, Stable Implementation Agreements dated December 1989.

7. STABLE MESSAGE HANDLING SYSTEMS

Editor's Note: For current stable MHS agreements, consult the aligned section in the Stable Implementation Agreements document. This section serves as a reference or pointer to Stable Agreements contained in Version 3, Edition 1, December 1989.

8. MESSAGE HANDLING SYSTEMS

8.1 INTRODUCTION

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.2 SCOPE

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.3 STATUS

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.4 ERRATA

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.5 MT KERNEL

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.5.1 Introduction

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.5.2 Elements of Service

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.5.3 MTS Transfer Protocol (P1)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.5.4 MTS - APDU Size

This section is for further study by the NIST X.400 SIG. The following support requirement may be increased in the future.

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.

8.5.5 1988/84 Interworking Considerations

Editor's Note: References to Section 7 are to the Stable Document.

An MTA conforming to this Agreement will downgrade 1988 P1 to 1984 P1 when relaying to 1984-based MTAs, as specified in Annex B of X.419 with the following additional requirements:

- Supplementary Information - will need to be truncated if it exceeds the pragmatic constraint identified in Version 2 of these Agreements (64 octets as opposed to 256 octets in the 1988 MHS standards), and
- Internal Trace Information - If the 1984-based MTA does not support Internal Trace Information per Section 7.7.3.2, the following description is not applicable. When a 1988-based MTA supports interworking with a 1984-based MTA that generates Internal Trace Information as per Section 7.7.3.3, the 1988-based MTA must support reception of the Internal Trace Information by converting the Internal Trace Information from the form in Section 7.7.3.2 to the form specified in 1988 X.411, as per the following description. When the 1988-based MTA sends to a 1984 MTA, the 1988-based MTA must apply the conversion to 1984, as described below. The Stable NBS Implementors Agreements X.400 (1984) implementors' agreements definition for MTA's Internal Trace Information is different from the X.400 (1988) MTA definition. Consequently, a X.400 (1988) MTA operating in an MD with other MTAs of 1984 vintage, must map the Internal Trace Information to and/or from the 1984 format.

  What follows are algorithms for mapping between X.400 (1988) Internal Trace element formats and the NIST IA X.400 (1984) Internal Trace element format.

  To avoid potential looping within a MD composed of 1984 and 1988 vintage MTAs, MD administrators are strongly advised to name all MTAs (1984 and 1988 vintages) using only the Printable String characters. In X.400 (1988) the MTA-Name is defined to be named using IA5 String characters where in the IAs for X.400 (1984) MTAs, NBS restricted the MTA-Name to be formed using the Printable String character subset of IA5. If the 1988-based MTA Name uses

IA5 characters not in the Printable String subset, that Internal Trace Element should be omitted when converting from 1988 to 1984.

1988 to 1984 Mapping

```
For each Internal Trace element in the sequence:
DO
  IF MTA-Name is made up of non-Printable String characters:
    Discard this Internal Trace element;
  ELSE
    {  Discard the GlobalDomainIdentifier;
       Copy the MTAname over;
       Within the MTASuppliedInformation:
         Copy the arrival time over;
         Copy the routing action over;
         IF attempted is present
           {  IF it is a domain:
                Discard it;
              IF it is an MTA:
                Copy it to Previous MTAName;
           }
         IF the additional actions are present:
           {  IF the deferred time is present:
                Copy it over;
              IF the other-actions is present:
                Discard it;
           }
    }
END-DO
```

1984 to 1988 Mapping

```
Find the [APPLICATION 30} entry in the P1 envelope;
FOR each Internal Trace element:
  DO
    Insert the GlobalDomainIdentifier of this MTA;
    Copy the MTAName over;
    Within the MTASuppliedInfo:
      Copy the arrival time;
      IF the deferred time is present:
        copy it to the additional actions field within the
          1988 Internal Trace information;
      IF the routing action is Relayed or Rerouted:
        copy it over;
      IF the routing action is Recipient-reassigned:
        map to Relayed;
      IF the previous MTAName is present:
        copy it to the MTAName in the attempted field;

  END-DO
```

8.6 IPM KERNEL

8.6.1 Introduction

See Stable Implementation Agreements Version 3, Edition 1 dated December 1989.

8.7 MESSAGE STORE

8.7.1 Introduction

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.2 Scope

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.3 Elements of Service

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.4 Attribute Types

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.5 Pragmatic Constraints for Attribute Types

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.6 Implementation of the MS with 1984 Systems

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.7 MS Access Protocol (P7)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.7.8 MTS Access Protocol (P3)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.8 REMOTE USER AGENT SUPPORT

8.8.1 Introduction

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.8.2 Scope

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.8.3 Elements of Service

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.8.4 MTS Access Protocol (P3)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.9 NAMING, ADDRESSING & ROUTING

8.9.1 Use of O/R Addresses for Routing

It is recognized that these Agreements enable a wide variety of naming and addressing attributes. Each domain may adopt particular routing schemes within its domain.

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

Addressing may be secured according to practices outside the scope of these agreements, such as:

- o manual directories
- o on-line directories, such as X.500
- o ORName address translation algorithms.

8.9.2 Distribution Lists

8.9.2.1 Introduction

This section identifies and specifies the Distribution Lists Functional Group, which covers all issues relating to the performance of distribution list (DL) expansion by an MTA. Other aspects concerned with the use of distribution lists are covered in the MT Kernel and IPM Kernel Functional Groups.

8.9.2.2 Elements of Service

This section specifies the requirements for support of Elements of Service for conformance to the Distribution Lists Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in Section 8.5.2.

Support for Elements of Service is specified for the MT Service only, and is only concerned with the performance of DL expansion by an MTA. Such support is in addition to the support requirements specified in Section 8.5 if this Functional Group is supported. Support for IPM Elements of Service for use of distribution lists is as specified in Section 8.6.

Table 8.13 Distribution Lists: MT Elements of Service

Element of Service	Support
DL Expansion History Indication	M
DL Expansion Prohibited	M
Use of Distribution List	M

8.9.3 MHS Use of Directory

8.9.3.1 Introduction

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

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

8.9.3.2 Functional Configuration

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

8.9.3.3 Functionality

Some functional usages of directories have been identified for UAs and the MTAs. These are:

UA Specific Functionality:

o Verify the existence of a Directory Name.

- Given a partial name, return a list of possibilities.

- Ability to scan directory entries.

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

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

- Return the members of a Distribution List.

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

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

MTA Specific Functionality:

- Authentication.

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

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

- Return the members of a Distribution List.

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

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

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

8.9.3.4 Naming and Attributes

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

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

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

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

(iii) the naming for mhs-user is that of organizationalPerson, ResidentialPerson, organizationalRole, organizationalUnit, organization, or Locality.

Note: The mhs-user object class is a generic object class which may be used in conjunction with another standard object class for the purpose of adding MHS information attributes, such as ORAddresses, to a Directory entry. The means to associate attributes of a generic object class to an entry (or to different entries) named by a standard object class(es) is by defining a new (un-)registered object class, whose superclass(es) is that of the naming object class(es), and of the generic object class. E.g., to associate mhs-user attributes in the organizationalPerson entry, the new unregistered object class can be defined as shown in Figure 8.9.

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

Figure 8.9 Example of Unregistered Object Class Definition

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

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

8.9.3.5 Elements of Service

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

The classification scheme for support of Elements of Service is as defined in Section 8.5.2.

Support for Elements of Service is specified both for the MT Service and for the IPM Service.

Table 8.14 Use of Directory: MT Elements of Service

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

Table 8.15 Use of Directory: IPM Elements of Service

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

8.10 MHS MANAGEMENT

For further study.

8.11 MHS SECURITY

8.11.1 Introduction

This section identifies and specifies the MHS Security Functional Group, which is intended to cover all issues relating to provision of secure messaging and secure access management facilities by an MHS implementation.

8.11.2 Elements of Service

This section specifies the requirements for support of Elements of Service for conformance to the MHS Security Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in Section 8.5.2.

Support for Elements of Service is specified both for the MT Service and for the IPM Service (Note: All Elements of Service listed below are 1988).

Table 8.16 MHS Security: MT Elements of Service

Element of Service	Origination	Reception
Content Confidentiality	*	*
Content Integrity	*	*
Message Flow Confidentiality	*	*
Message Origin Authentication	*	*
Message Security Labelling	*	*
Message Sequence Integrity	*	*
Non-repudiation of Delivery	*	*
Non-repudiation of Origin	*	*
Non-repudiation of Submission	*	*
Probe Origin Authentication	*	*
Proof of Delivery	*	*
Proof of Submission	*	*
Report Origin Authentication	*	*
Secure Access Management	*	*

Table 8.17 MHS Security: IPM Elements of Service

Element of Service	Origination	Reception
Content Confidentiality	*	*
Content Integrity	*	*
Message Flow Confidentiality	*	*
Message Origin Authentication	*	*
Message Security Labelling	*	*
Message Sequence Integrity	*	*
Non-repudiation of Delivery	*	*
Non-repudiation of Origin	*	*
Non-repudiation of Submission	*	*
Probe Origin Authentication	*	*
Proof of Delivery	*	*
Proof of Submission	*	*
Report Origin Authentication	*	*
Secure Access Management	*	*

8.12 SPECIALIZED ACCESS

8.12.1 Physical Delivery

8.12.1.1 Introduction

This section identifies and specifies the Physical Delivery Functional Group, which is intended to cover all issues relating to access to physical delivery systems by an MHS implementation.

8.12.1.2 Elements of Service

This section specifies the requirements for support of Elements of Service for conformance to the Physical Delivery Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in Section 8.5.2.

Support for Elements of Service is specified both for the MT Service and for the IPM Service (Note: All Elements of Service listed below are 1988).

Table 8.18 Physical Delivery: MT Elements of Service

Element of Service	Origination	Reception
Additional Physical Rendition	O	O
Basic Physical Rendition	M	M
Counter Collection	M	M
Counter Collection with Advice	O	O
Delivery via Bureaufax Service	O	O
EMS (Express Mail Service)	M	M
Ordinary Mail	M	M
Physical Delivery Notification by MHS	O	O
Physical Delivery Notification by PDS	O	O
Physical Forwarding Allowed	M	M
Physical Forwarding Prohibited	M	M
Registered Mail	O	O
Registered Mail to Addressee in Person	O	O
Request for Forwarding Address	O	O
Special Delivery	M	M
Undeliverable Mail with Return of Physical Message	M	M

Table 8.19 Physical Delivery: IPM Elements of Service

Element of Service	Origination (IPM UA)	Reception (PDAU)
Additional Physical Rendition	O	O
Basic Physical Rendition	O[1]	M
Counter Collection	M	M
Counter Collection with Advice	O	O
Delivery via Bureaufax Service	O	O
EMS (Express Mail Service)	M	M[2]
Ordinary Mail	O[1]	M
Physical Delivery Notification by MHS	O	O
Physical Delivery Notification by PDS	O	M
Physical Forwarding Allowed	O[1]	M
Physical Forwarding Prohibited	M	M
Registered Mail	O	O
Registered Mail to Addressee in Person	O	O
Request for Forwarding Address	O	O
Special Delivery	M	M[2]
Undeliverable Mail with Return of Physical Message	O[1]	M

Notes:

1) Provided by default (when using a physical delivery address).
2) Must support EMS and/or Special Delivery.

Table 8.20 Physical Delivery O/R Address Attributes

O/R Address Attribute Type	UA Orig	PDAU Recep
administration-domain-name	M	M
country-name	M	M
private-domain-name	M	M
physical-delivery-service-name	O	M
physical-delivery-country-name	M	M
postal-code	M	M
extension-postal-O/R-address-components	O	M
extension-physical-delivery-address-components	O	M
local-postal-attributes	O	M
physical-delivery-office-name	O	M
physical-delivery-office-number	O	M
physical-delivery-organization-name	O	M
physical-delivery-personal-name	O	M
post-office-box-address	O	M
poste-restante-address	O	M
street-address	O	M
unformatted-postal-address	M	M
unique-postal-name	O	M

The handling of Printable Strings and Teletex Strings in O/R address components is for further study.

Table 8.21 Character String Support

Character String	Origination (IPM UA)	Reception (PDAU)
Printable	*	M
Teletex	*	M

8.12.2 Other Access Units

8.12.2.1 Facsimile Access Units

The possible development of Agreements in this area is for further study.

8.12.2.2 Telex Access Units

It is not currently intended to develop Agreements in this area.

8.12.2.3 Teletex Access Units

It is not currently intended to develop Agreements in this area.

8.13 CONVERSION

8.13.1 Introduction

This section identifies and specifies the Conversion Functional Group, which is intended to cover all issues relating to support of conversion facilities by an MTA.

8.13.2 Elements of Service

This section specifies the requirements for support of Elements of Service for conformance to the Conversion Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as defined in Section 8.5.2.

Support for Elements of Service is specified for the MT Service only, and is only concerned with the performance of conversion by an MTA. Such support is in addition to the support requirements specified in Section 8.5 if this Functional Group is supported. Support for IPM Elements of Service for access to conversion facilities is as specified in Section 8.6.

Table 8.22 Conversion: MT Elements of Service

Element of Service	Support
Conversion Prohibition in Case of Loss of Information (1988)	*
Explicit Conversion	*
Implicit Conversion	*

8.14 USE OF UNDERLYING LAYERS

8.14.1 MTS Transfer Protocol (P1)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.14.2 MTS Access Protocol (P3) and MS Access Protocol (P7)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.15 ERROR HANDLING

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.15.1 PDU Encoding

8.15.2 Contents

8.15.3 Envelope

8.15.4 Reports

8.15.5 Pragmatic Constraints

If an implementation detects a pragmatic constraint violation, then it may generate an appropriate error indication but is not required to do so.

8.16 CONFORMANCE

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17 APPENDIX A: MHS PROTOCOL SPECIFICATIONS

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.1 MTS Transfer Protocol (P1)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.2 Interpersonal Messaging Protocol (P2)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.3 MTS Access Protocol (P3)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.4 MS Access Protocol (P7)

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.5 Message Store General Attribute Support

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.17.6 Message Store IPM Attribute Support

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.18 APPENDIX B: INTERPRETATION OF ELEMENTS OF SERVICE

The objective of this section is to provide clarification, where required, on the functionality of Elements of Service where the MHS standards are unclear or ambiguous. It is not the intent of this section to define how information should be made available or presented to an MHS user, nor is it intended to define how individual vendors should design their products.

The following MHS Elements of Service require further text to be added to their definitions to represent the proposed implementation of these Elements of Service for conformance to this Agreement. Elements of Service which are not referenced in this section are as defined in the MHS base standards.

Reply Request Indication

The reply-recipients and the reply-time may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request. Note that for an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Forwarded IP-message Indication

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

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

- 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.

8.19 APPENDIX C: RECOMMENDED PRACTICES

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.1 Printable String

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.2 Rendition of IA5Text

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.3 EDI Use of MHS

8.19.3.1 Introduction and Scope

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.3.2 Model

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.3.3 Protocol Elements Supported for EDI

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.19.3.4 Addressing and Routing

See Stable Implementation Agreements, Version 3, Edition 1 dated December 1989.

8.20 APPENDIX D: LIST OF ASN.1 OBJECT IDENTIFIERS

8.20.1 Content Types

8.20.2 Body Part Types

9. STABLE FTAM PHASE 2

Editor's Note: For Stable FTAM Phase 2 Agreements, consult the aligned section in the Stable Implementation Agreements Document. This section serves as a reference or pointer to Stable Agreements contained in Version 3, Edition 1, December 1989.

10. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 3

Editor's Note: For current Stable FTAM Phase 3 Agreements, consult the aligned section in the Stable Implementation Agreements, Version 3, Edition 1, dated December 1989.

11 DIRECTORY SERVICES PROTOCOLS

11.1 Introduction

Refer to Section 11.1 of Stable Agreements Version 3 Edition 1.

11.2 Scope and Field of Application

Refer to Section 11.2 of Stable Agreements Version 3 Edition 1.

11.3 Status

Refer to Section 11.3 of Stable Agreements Version 3 Edition 1.

11.4 Use of the Directory

This section will contain introductory text.

11.4.1 MHS

(TBD)

11.4.2 FTAM

(TBD)

11.5 Directory ASEs and Application Contexts

Refer to Section 11.5 of Stable Agreements Version 3 Edition 1.

11.6 Schema

Refer to Section 11.6 of Stable Agreements Version 3 Edition 1.

11.6.1 Support of Structures and Naming Rules

Refer to Section 11.6.1 of Stable Agreements Version 3 Edition 1.

11.6.2 Support of Object Classes and Subclasses

Refer to Section 11.6.2 of Stable Agreements Version 3 Edition 1.

11.6.3 Support of Attribute Types

Refer to Section 11.6.3 of Stable Agreements Version 3 Edition 1.

11.6.4 Support of Attribute Syntaxes

Refer to Section 11.6.4 of Stable Agreements Version 3 Edition 1.

11.6.5 Naming Contexts

Refer to Section 11.6.5 of Stable Agreements Version 3 Edition 1.

11.6.6 Common Profiles

Refer to Section 11.6.6 of Stable Agreements Version 3 Edition 1.

11.6.6.1 OIW Directory Common Application Directory Profile

Refer to Section 11.6.6.1 of Stable Agreements Version 3 Edition 1.

11.6.6.1.1 Standard Application Specific Attributes and Attribute Sets

Refer to Section 11.6.6.1.1 of Stable Agreements Version 3 Edition 1.

11.6.6.1.2 Standard Application Specific Object Classes

Refer to Section 11.6.6.1.2 of Stable Agreements Version 3 Edition 1.

11.6.6.2 OIW Directory Strong Authentication Directory Profile

Refer to Section 11.6.6.2 of Stable Agreements Version 3 Edition 1.

11.6.6.2.1 Other Profiles Supported

Refer to Section 11.6.6.2.1 of Stable Agreements Version 3 Edition 1.

11.6.6.2.2 Standard Application Specific Object Classes

Refer to Section 11.6.6.2.2 of Stable Agreements Version 3 Edition 1.

11.6.7 Restrictions on Object Class Definitions

Refer to Section 11.6.7 of Stable Agreements Version 3 Edition 1.

11.7 Pragmatic Constraints

Refer to Section 11.7 of Stable Agreements Version 3 Edition 1.

11.7.1 General Constraints

Refer to Section 11.7.1 of Stable Agreements Version 3 Edition 1.

11.7.1.1 Character Sets

Refer to Section 11.7.1.1 of Stable Agreements Version 3 Edition 1.

11.7.1.2 APDU Size Considerations

Refer to Section 11.7.1.2 of Stable Agreements Version 3 Edition 1.

11.7.1.3 Service Control (SC) Considerations

Refer to Section 11.7.1.3 of Stable Agreements Version 3 Edition 1.

11.7.1.4 Priority Service Control

Refer to Section 11.7.1.4 of Stable Agreements Version 3 Edition 1.

11.7.2 Constraints on Operations

Refer to Section 11.7.2 of Stable Agreements Version 3 Edition 1.

11.7.2.1 Filters

Refer to Section 11.7.2.1 of Stable Agreements Version 3 Edition 1.

11.7.2.2 Errors

Refer to Section 11.7.2.2 of Stable Agreements Version 3 Edition 1.

11.7.2.3 Error Reporting – Detection of Search Loop

Refer to Section 11.7.2.3 of Stable Agreements Version 3 Edition 1.

11.7.3 Constraints Relevant to Specific Attribute Types

Refer to Section 11.7.3 of Stable Agreements Version 3 Edition 1.

11.8 Conformance

Refer to Section 11.8 of Stable Agreements Version 3 Edition 1.

11.8.1 DUA Conformance

Refer to Section 11.8.1 of Stable Agreements Version 3 Edition 1.

11.8.2 DSA Conformance

Refer to Section 11.8.2 of Stable Agreements Version 3 Edition 1.

11.8.3 DSA Conformance Classes

Refer to Section 11.8.3 of Stable Agreements Version 3 Edition 1.

11.8.4 Authentication Conformance

Refer to Section 11.8.4 of Stable Agreements Version 3 Edition 1.

11.8.5 Directory Service Conformance

Refer to Section 11.8.5 of Stable Agreements Version 3 Edition 1.

11.8.6 The Directory Access Profile

Refer to Section 11.8.6 of Stable Agreements Version 3 Edition 1.

11.8.7 The Directory System Profile

Refer to Section 11.8.7 of Stable Agreements Version 3 Edition 1.

11.8.8 Digital Signature Protocol Conformance Profile

Refer to Section 11.8.8 of Stable Agreements Version 3 Edition 1.

11.8.9 Strong Authentication Protocol Conformance Profile

Refer to Section 11.8.9 of Stable Agreements Version 3 Edition 1.

11.9 Distributed Operations

Refer to Section 11.9 of Stable Agreements Version 3 Edition 1.

11.9.1 Referrals and Chaining

Refer to Section 11.9.1 of Stable Agreements Version 3 Edition 1.

11.9.2 Trace Information

Refer to Section 11.9.2 of Stable Agreements Version 3 Edition 1.

11.10 Underlying Services

Refer to Section 11.10 of Stable Agreements Version 3 Edition 1.

11.10.1 ROSE

Refer to Section 11.10.1 of Stable Agreements Version 3 Edition 1.

11.10.2 Session

Refer to Section 11.10.2 of Stable Agreements Version 3 Edition 1.

11.10.3 ACSE

Refer to Section 11.10.3 of Stable Agreements Version 3 Edition 1.

11.11 Access Control

Refer to Section 11.11 of Stable Agreements Version 3 Edition 1.

11.12 Test Considerations

Refer to Section 11.12 of Stable Agreements Version 3 Edition 1.

11.12.1 Major Elements of Architecture

Refer to Section 11.12.1 of Stable Agreements Version 3 Edition 1.

11.12.2 Search Operations

Refer to Section 11.12.2 of Stable Agreements Version 3 Edition 1.

11.13 Errors

Refer to Section 11.13 of Stable Agreements Version 3 Edition 1.

11.13.1 Permanent vs. Temporary Service Errors

Refer to Section 11.13.1 of Stable Agreements Version 3 Edition 1.

11.13.2 Guidelines for Error Handling

Refer to Section 11.13.2 of Stable Agreements Version 3 Edition 1.

11.13.2.1 Introduction

Refer to Section 11.13.2.1 of Stable Agreements Version 3 Edition 1.

11.13.2.2 Symptoms

Refer to Section 11.13.2.2 of Stable Agreements Version 3 Edition 1.

11.13.2.3 Situations

Refer to Section 11.13.2.3 of Stable Agreements Version 3 Edition 1.

11.13.2.4 Error Actions

Refer to Section 11.13.2.4 of Stable Agreements Version 3 Edition 1.

11.13.2.5 Reporting

Refer to Section 11.13.2.5 of Stable Agreements Version 3 Edition 1.

11.14 Specific Authentication Schemes

Refer to Section 11.14 of Stable Agreements Version 3 Edition 1.

11.14.1 Specific Strong Authentication Schemes

Refer to Section 11.14.1 of Stable Agreements Version 3 Edition 1.

11.14.1.1 ElGamal

Refer to Section 11.14.1.1 of Stable Agreements Version 3 Edition 1.

11.14.1.1.1 References

Refer to Section 11.14.1.1.1 of Stable Agreements Version 3 Edition 1.

11.14.1.1.2 Background

Refer to Section 11.14.1.1.2 of Stable Agreements Version 3 Edition 1.

11.14.1.1.3 Digital Signature

Refer to Section 11.14.1.1.3 of Stable Agreements Version 3 Edition 1.

11.14.1.1.4 Verification

Refer to Section 11.14.1.1.4 of Stable Agreements Version 3 Edition 1.

11.14.1.1.5 Known Constraints on Parameters

Refer to Section 11.14.1.1.5 of Stable Agreements Version 3 Edition 1.

11.14.1.1.6 Note on subjectPublicKey

Refer to Section 11.14.1.1.6 of Stable Agreements Version 3 Edition 1.

11.14.1.2 One–Way Hash Functions

Refer to Section 11.14.1.2 of Stable Agreements Version 3 Edition 1.

11.14.1.2.1 SQUARE–MOD–N Algorithm

Refer to Section 11.14.1.2.1 of Stable Agreements Version 3 Edition 1.

11.14.1.2.2 MD2 Algorithm

Refer to Section 11.14.1.2.2 of Stable Agreements Version 3 Edition 1.

11.14.1.2.3 Study of Other One–Way Hash Functions

Refer to Section 11.14.1.2.3 of Stable Agreements Version 3 Edition 1.

11.14.1.2.4 Use of One–Way Hash Functions in Forming Signatures

Refer to Section 11.14.1.2.4 of Stable Agreements Version 3 Edition 1.

11.14.1.3 ASN.1 for Strong Authentication Algorithms

Refer to Section 11.14.1.3 of Stable Agreements Version 3 Edition 1.

11.14.2 Protected Simple Authentication

Refer to Section 11.14.2 of Stable Agreements Version 3 Edition 1.

11.14.3 Simple Authentication

There are two major classes of authentication supported by the Directory (i.e., simple and strong authentication). Simple authentication is based on a password being passed between the two associated entities (DUA–DSA or DSA–DSA). In the case of the DUA–DSA interaction, the password is compared in some way with the password attribute in the user's entry in the Directory. In the case of DSA–DSA interaction, this cannot be done since the DSA object class, as defined in the Directory Documents (Part 7, clause 6.14) does not contain a password attribute.

To facilitate simple authentication between DSAs, a DSA shall have local access to a list of one or more known DSAs, with a copy of each known DSA's password. Maintenance of that information is done through the use of bilateral agreements between DSA administrtors.

11.15 Appendix A: Maintenance of Attribute Syntaxes

Refer to Section 11.15 of Stable Agreements Version 3 Edition 1.

11.15.1 Introduction

Refer to Section 11.15.1 of Stable Agreements Version 3 Edition 1.

11.15.2 General Rules

Refer to Section 11.15.2 of Stable Agreements Version 3 Edition 1.

11.15.3 Checking Algorithms

Refer to Section 11.15.3 of Stable Agreements Version 3 Edition 1.

11.15.3.1 distinguishedNameSyntax

Refer to Section 11.15.3.1 of Stable Agreements Version 3 Edition 1.

11.15.3.2 integerSyntax

Refer to Section 11.15.3.2 of Stable Agreements Version 3 Edition 1.

11.15.3.3 telephoneNumberSyntax

Refer to Section 11.15.3.3 of Stable Agreements Version 3 Edition 1.

11.15.3.4 countryName

Refer to Section 11.15.3.4 of Stable Agreements Version 3 Edition 1.

11.15.3.5 preferredDeliveryMethod

Refer to Section 11.15.3.5 of Stable Agreements Version 3 Edition 1.

11.15.3.6 presentationAddress

Refer to Section 11.15.3.6 of Stable Agreements Version 3 Edition 1.

11.15.4 Matching Algorithms

Refer to Section 11.15.4 of Stable Agreements Version 3 Edition 1.

11.15.4.1 UTCTimeSyntax

Refer to Section 11.15.4.1 of Stable Agreements Version 3 Edition 1.

11.15.4.2 distinguishedNameSyntax

Refer to Section 11.15.4.2 of Stable Agreements Version 3 Edition 1.

11.15.4.3 caseIgnoreListSyntax

Refer to Section 11.15.4.3 of Stable Agreements Version 3 Edition 1.

11.16 Appendix B: Glossary

Refer to Section 11.16 of Stable Agreements Version 3 Edition 1.

11.17 Appendix C: Requirements for Distributed Operations

Refer to Section 11.17 of Stable Agreements Version 3 Edition 1.

11.17.1 General Requirements

Refer to Section 11.17.1 of Stable Agreements Version 3 Edition 1.

11.17.2 Protocol Support

Refer to Section 11.17.2 of Stable Agreements Version 3 Edition 1.

11.17.2.1 Usage of ChainingArguments

Refer to Section 11.17.2.1 of Stable Agreements Version 3 Edition 1.

11.17.2.2 Usage of Chainging Results

Refer to Section 11.17.2.2 of Stable Agreements Version 3 Edition 1.

11.18 Appendix D: Guideline for Applications Using the Directory

Refer to Section 11.18 of Stable Agreements Version 3 Edition 1.

11.18.1 Tutorial

Refer to Section 11.18.1 of Stable Agreements Version 3 Edition 1.

11.18.1.1 Overview

Refer to Section 11.18.1.1 of Stable Agreements Version 3 Edition 1.

11.18.1.2 Use of the Directory Schema

Refer to Section 11.18.1.2 of Stable Agreements Version 3 Edition 1.

11.18.1.2.1 Use of Existing Object Classes

Refer to Section 11.18.1.2.1 of Stable Agreements Version 3 Edition 1.

11.18.1.2.2 Kinds of Object Classes

Refer to Section 11.18.1.2.2 of Stable Agreements Version 3 Edition 1.

11.18.1.2.3 Use of Unregistered Object Classes

Refer to Section 11.18.1.2.3 of Stable Agreements Version 3 Edition 1.

11.18.1.2.4 Side Effects of Creating Unregistered Object Classes

Refer to Section 11.18.1.2.4 of Stable Agreements Version 3 Edition 1.

11.18.2 Creation of New Object Classes

Refer to Section 11.18.2 of Stable Agreements Version 3 Edition 1.

11.18.2.1 Creation of New Subclasses

Refer to Section 11.18.2.1 of Stable Agreements Version 3 Edition 1.

11.18.2.2 Creation of New Attributes

Refer to Section 11.18.2.2 of Stable Agreements Version 3 Edition 1.

11.18.3 DIT Structure Rules

Refer to Section 11.18.3 of Stable Agreements Version 3 Edition 1.

11.19 Appendix E: Template for an Application Specific Profile for Use of the Directory

Refer to Section 11.19 of Stable Agreements Version 3 Edition 1.

12. STABLE SECURITY AGREEMENTS

Editor's Note: This section points to Stable Security Agreements which are contained in the aligned section of the Stable Implementation Agreements, Version 3, Edition 1, December 1989.

13. SECURITY

13.1 INTRODUCTION

13.1.1 References

13.1.2 Assumptions

13.1.3 Definitions

13.1.4 Motivation

13.1.5 Security Chapter Structure

13.2 SCOPE AND FIELD OF APPLICATION

13.3 STATUS

13.4 ERRATA

13.5 GENERAL OSI SECURITY MODEL

13.5.1 General Matrix from 7498-2

13.5.2 Selected Matrix of Services/Layers

13.5.3 Security Domain Model

13.6 OSI MANAGEMENT SECURITY AND SECURITY MANAGEMENT

13.7 PHYSICAL LAYER

13.7.1 Introduction

13.7.1.1 References

13.7.1.2 Definitions

13.7.1.3 Assumptions

13.7.1.4 Motivation

13.7.2 Scope and Field of Application

13.7.3 Specific Security Model

13.7.4 Services Offered

13.7.5 Services Required

13.7.6 Protocols

13.7.7 Management Elements Required/Impacted

13.7.8 Conformance Class Definitions

13.7.9 Conformance Class Specifications

13.7.10 Registration Issues Requirements

13.8 DATA-LINK LAYER

13.8.1 Introduction

13.8.1.1 References

13.8.1.2 Definitions

13.8.1.3 Assumptions

13.8.1.4 Motivation

13.8.2 Scope and Field of Application

13.8.3 Specific Security Model

13.8.4 Services Offered

13.8.5 Services Required

13.8.6 Protocols

13.8.7 Management Elements Required/Impacted

13.8.8 Conformance Class Definitions

13.8.9 Conformance Class Specifications

13.8.10 Registration Issues Requirements

13.9 NETWORK LAYER

13.9.1 Introduction

13.9.1.1 References

13.9.1.2 Definitions

13.9.1.3 Assumptions

13.9.1.4 Motivation

13.9.2 Scope and Field of Application

13.9.3 Specific Security Model

13.9.4 Services Offered

13.9.5 Services Required

13.9.6 Protocols

13.9.7 Management Elements Required/Impacted

13.9.8 Conformance Class Definitions

13.9.9 Conformance Class Specifications

13.9.10 Registration Issues Requirements

13.10 TRANSPORT LAYER

13.10.1 Introduction

13.10.1.1 References

13.10.1.2 Definitions

13.10.1.3 Assumptions

13.10.1.4 Motivation

13.10.2 Scope and Field of Application

13.10.3 Specific Security Model

13.10.4 Services Offered

13.10.5 Services Required

13.10.6 Protocols

13.10.7 Management Elements Required/Impacted

13.10.8 Conformance Class Definitions

13.10.9 Conformance Class Specifications

13.10.10 Registration Issues Requirements

13.11 SESSION LAYER

13.11.1 Introduction

13.11.1.1 References

13.11.1.2 Definitions

13.11.1.3 Assumptions

13.11.1.4 Motivation

13.11.2 Scope and Field of Application

13.11.3 Specific Security Model

13.11.4 Services Offered

13.11.5 Services Required

13.11.6 Protocols

13.11.7 Management Elements Required/Impacted

13.11.8 Conformance Class Definitions

13.11.9 Conformance Class Specifications

13.11.10 Registration Issues Requirements

13.12 PRESENTATION LAYER

13.12.1 Introduction

13.12.1.1 References

13.12.1.2 Definitions

13.12.1.3 Assumptions

13.12.1.4 Motivation

13.12.2 Scope and Field of Application

13.12.3 Specific Security Model

13.12.4 Services Offered

13.12.5 Services Required

13.12.6 Protocols

13.12.7 Management Elements Required/Impacted

13.12.8 Conformance Class Definitions

13.12.9 Conformance Class Specifications

13.12.10 Registration Issues Requirements

13.13 APPLICATION LAYER

13.13.1 Introduction

13.13.1.1 References

13.13.1.2 Definitions

13.13.1.3 Assumptions

13.13.1.4 Motivation

13.13.2 Scope and Field of Application

13.13.3 Specific Security Model

13.13.4 Services Offered

13.13.4.1 ACSE

13.13.4.2 ROSE

13.13.4.3 TRSE

13.13.4.4 CCR

13.13.5 Services Required

13.13.6 Protocols

13.13.7 Management Elements Required/Impacted

13.13.8 Conformance Class Definitions

13.13.9 Conformance Class Specifications

13.13.10 Registration Issues Requirements

13.14 FTAM

13.14.1 Introduction

13.14.1.1 References

13.14.1.2 Definitions

13.14.1.3 Assumptions

13.14.1.4 Motivation

13.14.2 Scope and Field of Application

13.14.3 Specific Security Model

13.14.4 Services Offered

13.14.5 Services Required

13.14.6 Protocols

13.14.7 Management Elements Required/Impacted

13.14.8 Conformance Class Definitions

13.14.9 Conformance Class Specifications

13.14.10 Registration Issues Requirements

13.15 Message Handling System Security

The following definitions of the elements of security service are based on the 1988 CCITT Recommendations on the Message Handling System (X.400). The fourteen (14) elements of security service are refinements of the five (5) primary security services as defined in IS 7498 Part 2 (Security Architecture). The Implementor's Workshop prepared Table 13.2 that summarizes where in the MHS the element of security service may be performed (the check marks) as stated in the MHS Recommendations. The Special Interest Group in Security (SIG-SEC) then examined each of the 14 elements of security service and placed a priority rating (1-5) next to one of the checkmarks in each row representing the priority that should be given for consideration of standardization and implementation of that element of service. The SIG-SEC reviewed the User Agent (UA) to User Agent peer entities as the first (perhaps preferred) place to implement security and used the check mark in that column if one was present. The SIG-SEC then reviewed the Message Transfer Agent (MTA) to Message Transfer Agent as the second place to implement security if it has not been implemented in the UA-UA protocol. Finally, the interface between the UA and the MTA was investigated for implementing security.

The Implementor's Workshop will be using this table and the set of definitions as a basis upon which future work in MHS security may be performed. The table is and subject to change during future meetings.

Table 13.1 X.400 Relationship between Elements of Security Service and MHS Components

	UA-MS	MS-MTA	UA-UA	UA-MTA	MTA-MTA	MTA-UA	MS-UA
Message Origin Authentication			√1	√			
Report Origin Authentication					√4	√	
Probe Origin Authentication		√		√5			
Proof of Delivery			√2				√
Proof of Submission						√5	
Peer Entity Authentication	√	√		√	√4	√	√
Content Integrity			√1				
Content Confidentiality			√1				
Message Flow Confidentiality			√4				
Message Sequence Integrity			√2				
Non Repudiation of Origin			√1				
Non Repudiation of Submission						√5	
Non repudiation of Delivery			√3				
Access Control	√	√	√1	√	√	√	√

UA: User Agent
MS: Message Store
MTA: Message Transfer Agent

13.15.1 Definitions of Elements of Security Service

Message Origin Authentication **MT**

This element of service allows the originator of a message to provide to the recipient(s) of the message, and any MTA through which the message is transferred, a means by which the origin of the message can be authenticated (i.e. a signature). Message Origin Authentication can be provided to the recipient(s) of the message, and any MTA through which the message is transferred, on a per-message basis using an asymmetric encryption technique, or can be provided only to the recipient(s) of the message, on a per-recipient basis either a asymmetric or a symmetric encryption technique.

Report Origin Authentication **MT**

This element of service allows the originator of a message (or probe) to authenticate the origin of a report on the delivery or non-delivery of the subject message (or probe), (a signature). report Origin Authentication is on a per-report basis, and uses an asymmetric encryption technique.

Probe Origin Authentication **MT**

This element of service allows the originator of a probe to provide to any MTA through which the probe is transferred a means to authenticate the origin of the probe (i.e. a signature). Probe Origin Authentication is on a per-probe basis, and uses an asymmetric encryption technique.

Proof of Delivery **MT**

This element of service allows the originator of a message to obtain from the recipient(s) of the message the means to authenticate the identity of the recipient(s) and the delivered message and content. Message recipient authentication is provided to the originator of a message on a per-recipient basis using either symmetric or asymmetric encryption techniques.

Proof of Submission **MT**

This element of service allows the originator of a message to obtain from the MTS the means to authenticate that the message was submitted for delivery to the originally intended recipient. Message submission authentication is provided on a per-recipient basis, and can use symmetric or asymmetric encryption techniques.

Peer Entity Authentication **MT**

This element of service provides confirmation of the identity of the Entity (UA, MTA, MS). It provides confidence at the time of usage only that an entity is not attempting to masquerade as an unauthorized entity.

Content Confidentiality **MT**

This element of service allows the originator of a message to protect the content of the message from disclosure to someone other than the intended recipient(s). Content Confidentiality is on a per message basis, and can use either an asymmetric or a symmetric encryption technique.

Content Integrity **MT**

This element of service allows the originator of the message to provide to the recipient of the message a means by which the recipient can verify that the content of the message has not been modified. Content Integrity is on a per-recipient basis, and can use either an asymmetric or a symmetric encryption technique.

Message Flow Confidentiality **MT**

This element of service allows the originator of the message to protect information which might be derived from observation of the message flow.

Message Sequence Integrity **MT**

This element of service allows the originator of the message to provide to a recipient of the message a means by which the recipient can verify that the sequence of messages from the originator to the recipient has been preserved (without message loss, re-ordering, or replay). Message Sequence Integrity is on a per-recipient basis, and can use either an asymmetric or a symmetric encryption technique.

Non Repudiation of Origin **MT**

This element of service allows the originator of a message to provide the recipient(s) of the message irrevocable proof of the origin of the message. This will protect against any attempt by the originator to subsequently revoke the message or its content. Non Repudiation of Origin is provided to the recipient(s) of a message on a per message basis using asymmetric encryption techniques.

Non Repudiation of Submission **MT**

This element of service allows the originator of a message to obtain irrevocable proof that a message was submitted to the MTS for delivery to the originally specified recipient(s). This will protect against any attempt by the MTS to subsequently deny that the message was submitted for delivery to the originally specified recipient(s). Non Repudiation of Submission is provided to the originator of a message on a per message basis, and uses an asymmetric encryption technique.

Non Repudiation of Delivery **MT**

This element of service allows the originator of a message to obtain from the recipient(s) of the message, irrevocable proof that the message was delivered to the recipient(s). This will protect against any attempt by the recipient(s) to subsequently deny receiving the message or its content. Non Repudiation of Delivery is provided to the originator of a message on a per-recipient basis using asymmetric encryption techniques.

Access Control **MT**

This element of service provides protection against unauthorized use of the resources accessed via MHS. Access decisions are directed by a security policy which may be identity and/or role based.

13.16 DIRECTORY

13.16.1 Introduction

13.16.1.1 References

13.16.1.2 Definitions

13.16.1.3 Assumptions

13.16.1.4 Motivation

13.16.2 Scope and Field of Application

13.16.3 Specific Security Model

13.16.4 Services Offered

13.16.5 Services Required

13.16.6 Protocols

13.16.7 Management Elements Required/Impacted

13.16.8 Conformance Class Definitions

13.16.9 Conformance Class Specifications

13.16.10 Registration Issues Requirements

13.17 VTP

13.17.1 Introduction

13.17.1.1 References

13.17.1.2 Definitions

13.17.1.3 Assumptions

13.17.1.4 Motivation

13.17.2 Scope and Field of Application

13.17.3 Specific Security Model

13.17.4 Services Offered

13.17.5 Services Required

13.17.6 Protocols

13.17.7 Management Elements Required/Impacted

13.17.8 Conformance Class Definitions

13.17.9 Conformance Class Specifications

13.17.10 Registration Issues Requirements

14. ISO VIRTUAL TERMINAL PROTOCOL

Editor's Note: References to Stable Agreements in this section refer to Version 3, Edition 1, December 1989.

14.1 INTRODUCTION

See Stable Agreements.

14.2 SCOPE AND FIELD OF APPLICATION

14.2.1 Phase Ia Agreements

See Stable Agreements

14.2.2 Phase Ib Agreements

See Stable Agreements regarding Forms profile.

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

14.2.3 Phase II Agreements

See Stable Agreements regarding X.3 profile.

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

14.3 STATUS

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

14.3.1 Status of Phase Ia

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

14.3.2 Status of Phase Ib

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

14.3.3 Status of Phase II

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

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

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

14.4 ERRATA

14.5 CONFORMANCE

See Stable Agreements.

14.6 PROTOCOL

See Stable Agreements.

14.7 OIW REGISTERED CONTROL OBJECTS

14.7.1 Sequenced Application (SA)

See Stable Agreements.

14.7.2 Unsequenced Application (UA)

See Stable Agreements.

14.7.3 Sequenced Terminal (ST)

See Stable Agreements.

14.7.4 Unsequenced Terminal (UT)

See Stable Agreements.

14.7.5 Termination Conditions CO (TC)

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

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

14.7.5.1 Entry Number

To be supplied by the Registration Authority.

14.7.5.2 Name of Sponsoring Body

NIST/OSI Workshop for Implementors of OSI, VTSIG.

14.7.5.3 Date

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

14.7.5.4 Identifier

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

14.7.5.5 Descriptor Value

"OIW VT CO for Termination Conditions"

14.7.5.6 CO VTE-parameters

```
CO-structure    =  ,    *(not defined in this registration,
                        see note 1 in 14.7.5.8)*
CO-priority = "normal"
    {
```

```
    CO-element-id   = 1,    *(termination length)*
    CO-category = "integer",
    CO-size     = 65535 },
    {
    CO-element-id   = 2, *(time-out mantissa)*
    CO-category = "integer",
    CO-size     = 65535 },
    {
    CO-element-id   = 3, *(time-out exponent)*
    CO-category = "integer",
    CO-size     = 65535 },
*(the following represents possibly multiple invocations of
a generic data element type, according to the value of CO-
structure for the instance of this CO. )*
    FOR N=4 to CO-structure
    {
    CO-element-id   = N,    *(acts as integer identifier for
                            the events in this element)*
    CO-category = "transparent",
    CO-size     =    *(not defined in this registration, see
                     note 2 in 14.7.5.8)* }
```

14.7.5.7 CO Values, Semantic and Update Syntax

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

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

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

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

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

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

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

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

14.7.5.8 Additional Information

Note 1: The value of CO-structure is defined in the profile to be the number of types of termination conditions available for use within the profile.

Note 2: The value of CO-size for each additional data element of this CO must be defined within the profile definition which uses those additional termination conditions.

14.7.5.9 Usage

Defined in profile.

14.8 OIW DEFINED VTE-PROFILES

14.8.1 Telnet Profile

See Stable Agreements.

14.8.2 Transparent Profile

See Stable Agreements.

14.8.3 Forms Profile

See Stable Agreements.

Proposed Definitive Note 7s to be added to 14.8.3.6.1 of the Forms Profile in the Stable Agreements:

The real cursor position associated with the completion of the terminal user's input is, in many cases, used to convey information to the terminal user's application. As a result, when the terminal VT-user relinquishes WAVAR, this cursor position must be reported to the application VT-user.

When the completion of the terminal user's input is associated with a field, this cursor position is reported via a CCO CO-update.

When the completion of the terminal user's input is not associated with a field (only possible when the value for the VTE-parameter is "allowed"), this cursor position is reported via an appropriate implicit or explicit addressing operation.

14.8.4 X3 Profile

See Stable Agreements.

14.8.5 Scroll Profile

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

14.8.5.1 Introduction

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

Scrolling is bi-directional, forward and backward.

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

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

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

14.8.5.2 Association Requirements

14.8.5.2.1 Functional Units

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

14.8.5.2.2 Mode

This profile operates in A-mode.

14.8.5.3 Profile Body

```
Display-objects =
{
    {
    display-object-name = DOA,
    DO-access = profile-argument-r1,
    dimension = "two",
        x-dimension =
        {
            x-bound = profile-argument-r2,
            x-addressing = "no-constraint",
            x-absolute = "no",
            x-window = x-bound
        },
        y-dimension =
        {
            y-bound = "unbounded",
            y-addressing = "no-constraint",
            y-absolute = "no",
            y-window = profile-argument-r10
        },

    erasure-capability = "yes",

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

    repertoire-assignment = profile-argument-r4,

    DO-emphasis = profile-argument-r5,

    foreground-colour-capability =
                     profile-argument-r6,
    foreground-colour-assignment =
                     profile-argument-r7,
    background-colour-capability =
                     profile-argument-r6,
    background-colour-assignment =
                     profile-argument-r8
    },
    {
    display-object-name = DOB,
    DO-access = opposite of profile-argument-r1,
    dimension = "two",
        x-dimension =
        {
            x-bound = profile-argument-r2,
            x-addressing = "no-constraint",
            x-absolute = "no",
            x-window = x-bound
        },
        y-dimension =
        {
            y-bound = "unbounded",
            y-addressing = "higher only",
            y-absolute = "no",
            y-window = 1
        },
    erasure capability = "yes",

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

    repertoire-assignment = profile-argument-r4,

    DO-emphasis = profile-argument-r5,

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

Control-objects =
{
    {
    CO-name         = E,    *(standard Echo CO)*
    CO-type-identifier  = vt-b-sco-echo,
    CO-access           = profile-argument-r1,
    CO-priority     = "normal",
    CO-trigger      = "selected",
    CO-category     = "boolean",
    CO-size         = 1
    },
IF r9 = "TE" THEN
{
CO-name         = TE, *(Termination Event CO)*
CO-type-identifier  = vt-b-sco-tco,
CO-access           = opposite of profile-argument-r1,
CO-priority     = "normal",
CO-trigger      = "selected",
CO-category     = "integer"
},

{
CO-name         = SA, *(NIST Registered CO)*
CO-type-identifier  = nist-vt-co-misc-sa,
CO-access           = profile-argument-r1,
CO-priority     = "normal",
CO-trigger      = "not selected",
CO-category     = "integer",
CO-size         = 65535
},

{
CO-name         = UA, *(NIST Registered CO)*
CO-type-identifier  = nist-vt-co-misc-ua,
CO-access           = profile-argument-r1,
CO-priority     = "urgent",
CO-category     = "integer",
CO-size         = 65535
},

{
CO-name         = ST, *(NIST Registered CO)*
CO-type-identifier  = nist-vt-co-misc-st,
CO-access           = opposite of profile-argument-r1,
CO-priority     = "normal",
CO-category     = "integer",
CO-size         = 65535
},

{
CO-name         = UT, *(NIST Registered CO)*
CO-type-identifier  = nist-vt-co-misc-ut,
CO-access           = opposite of profile-argument-r1,
CO-priority     = "urgent",
CO-category     = "integer",
CO-size         = 65535
},
{
CO-name         = TC, *(Termination conditions CO)*
CO-type-identifier  = nist-vt-co-tcco-tc,
CO-structure        = N, *( defined with TCCO)*
CO-access           = profile-argument-r1,
CO-priority     = "normal",
        {
        CO-element-id   = 1, *(termination length)*
         CO-category    = "integer",
         CO-size        = 65535 },
        {
        CO-element-id   = 2, *(time-out mantissa)*
         CO-category    = "integer",
         CO-size        = 65535 },
        {
        CO-element-id   = 3, *(time-out exponent)*
         CO-category    = "integer",
         CO-size        = 65535 },
        {
        CO-element-id   = 4-N, *(from registered TCCO)*
         CO-category    = ???,
         CO-size        = ??? }

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

Device-objects =
{
    {
    device-name = DVA,  *("output" device object)*
    device-default-CO-access = profile-argument-r1,
    device-default-CO-initial-value = 1."true",
    device-display-object = DOA,
    device-minimum-X-array-length = profile-argument-r2,
    device-minimum-Y-array-length = profile-argument-r3,
    device-control-object = {SA,UA}
    },
    {
    device-name = DVB,  *("input" device object)*
    device-default-CO-access = opposite of
                          profile-argument-r1,
    device-default-CO-initial-value = 1."true",
    device-display-object = DOB,
    device-minimum-X-array-length = profile-argument-r2,
    device-control-object = profile-argument-r9,
    device-control-object = {ST,UT},
    device-control-object = TE
    }
}

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

14.8.5.4 Profile Argument Definitions:

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

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

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

Note: Indicates screen length.

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

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

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

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

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

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

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

14.8.5.5 Profile Dependent CO Information

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

14.8.5.6 Profile Notes

14.8.5.6.1 Definitive Notes

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

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

14.8.5.6.2 Informative Notes

1. This profile models a scrolling device which is capable of scrolling both forwards and backwards. The display pointer may be moved backwards to modify earlier lines. A typical use for this

profile is for applications where type-ahead may be advantageous and control over local echo "on"/"off" is required, e.g. the type of application where a conventional teletypewriter device or'teletype-compatible' video device having 'full duplex'capability is often used. Display object DOA referred to above is typically mapped to the display or printing device and display object DOB is typically mapped to the keyboard.

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

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

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

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

14.8.5.7 Specific Conformance Requirements

None.

14.9 APPENDIX A

See Stable Agreements.

14.10 APPENDIX B - CLARIFICATIONS

14.10.1 Defaults

See Stable Agreements.

14.11 APPENDIX C - OBJECT IDENTIFIERS

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

Profiles defined by OIW VT SIG:

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

Control Objects defined by OIW VT SIG:

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

15. TRANSACTION PROCESSING

15.1 Introduction

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

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

Atomicity: a property of a set of related operations such that the operations are either all performed, or none of them are performed.

Consistency: a property of a set of related operations such that the effect of the operations is performed accurately, correctly, and with validity, with respect to application semantics. Bound data is moved from one consistent state to another consistent state.

Isolation: a property of a set of related operations such that the partial results of the operations are not accessible, except by operations of the set.

Durability: a property of a completed set of related operations such that all the effects of the operation are not altered by any sort of failure.

15.2 Scope

These agreements will address the following areas

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

6. Additionally, the following interoperability issues will be addressed:
   - A. TP usage by other OSI standards
   - B. Application context
   - C. Security

15.3 SPECIFICATION OF FUNCTIONAL UNITS

15.3.1 FUNCTIONAL UNITS

Kernel

Polarized Control

Shared Control

Handshake

Commit

Unchained Transactions

15.3.2 COMBINATIONS OF FUNCTIONAL UNITS

Application Transactions

Unchained Provider-supported Transactions

Chained Provider-supported Transactions

15.4 TP USE OF OSI SERVICES

15.4.1 ACSE - ASSOCIATION MANAGEMENT

15.4.2 CCR - PROVIDER ACID PROPERTIES

15.4.3 PRESENTATION SERVICES

15.4.4 DIRECTORY SERVICES

15.5 IMPLEMENTATIONS ISSUES NOT SPECIFIED IN ISO 10026

15.5.1 APPLICATION CONTEXT

15.5.2 SECURITY

15.5.3 RECOMMENDED PRACTICES

15.6 CONFORMANCE STATEMENT

15.7 OSI TRANSACTION PROCESSING PROTOCOL AGREEMENTS

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

- Optional and Mandatory PDU fields and their ranges
- Optional and Mandatory ASE service primitive parameters and their ranges

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

Unless otherwise noted, the following column descriptions and keys apply to all tables:

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

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

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

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

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

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

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

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

TPPM:	Transaction Processing Protocol Machine
REQ:	Requesting TPSUI

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

TPPM:	TP Protocol machine
IND:	Receiving TPSUI

Notes: Any additional comments applying to the parameter.

TP-BEGIN-DIALOGUE-RI

Sending, to begin a dialogue

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

Sending, to begin a TP channel

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	
Channel-utilization	C	Y	TPPM		

Receiving, to begin a dialogue

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

Receiving, to begin a TP channel

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	
Channel-Utilization	C	Y	TPPM		

Notes

1. May need to determine limits on the amount and type of data passed in this manner.
2. See section "Support of Functional Units" for minimum valid combinations of functional units.

TP-BEGIN-DIALOGUE-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Dialogue/Channel-Identifier	M	Y	TPPM	0..2**31-1	

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Dialogue/Channel Identifier	M	Y	TPPM	0..2**31-1	

TP-REJECT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostic	C	Y			1, 4
User-data	O	O	Req		2, 3
Dialogue/Channel Identifier	M	Y	TPPM	0..2**31-1	

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostic	C	Y			1, 4
User-data	O	O	Req		2, 3
Dialogue/Channel Idebtifier	M	Y	TPPM	0..2**31-1	

Notes:

1. User/Provider division of values is unclear in standard's ASN.1.
2. May need to determine limits on the amount and type of data passed in this manner.
3. Parameter is present on provider rejects.
4. Parameter is present on user rejects.

TP-BID-RI

No parameters

TP-BID-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Result	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Result	M	Y	TPPM		

TP-END-DIALOGUE-RI

No parameters

TP-U-ERROR-RI

No parameters

TP-U-ERROR-RC

No parameters

TP-P-ERROR-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Diagnostic	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Diagnostic	M	Y	Ind		

TP-ABORT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Diagnostics	C	Y	TPPM		1, 4
User-data	C	O	Req		2, 3

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	Ind		
Diagnostics	C	Y	Ind		1, 4
User-data	C	O	Ind		2, 3

Notes:

1. May want to specify meanings for the reason codes, Permanent and Transient failure.
2. May need to determine limits on the amount and type of data passed in this manner. Text says parm is optional, ASN.1 says mandatory.
3. Parameter is present on provider abort.
4. Parameter is present on user abort.

TP-REQUEST-CONTROL-RI

Has no parameters

TP-GRANT-CONTROL-RI

No parameters

TP-HANDSHAKE-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Type	M	Y	TPPM		
Confirmation	C	Y	Req		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Type	M	Y	TPPM		
Confirmation	C	Y	TPPM		1

Note:

1. Parameter is present only on handshake when Shared Control functional unit is active.

TP-HANDSHAKE-RC

No parameters

TP-HANDSHAKE-AND-GRANT-CONTROL-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Confirmation	M	Y	Req		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Confirmation	M	Y	TPPM		

TP-HANDSHAKE-AND-GRANT-CONTROL-RC

No parameters

TP-DEFER-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
End-dialogue	O	Y	TPPM		1
Grant-control	O	Y	TPPM		1
Next-Transaction	O	Y	TPPM		1

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
End-dialogue	O	Y	TPPM		1
Grant-control	O	Y	TPPM		1
Next-Transaction	O	Y	TPPM		1

Notes:

1. The field is mandatory only if required by supported functional units, else it is not used.

TP-PREPARE-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Data-permitted	O		Req		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Data-permitted	O		Ind		

TP-UNCHAIN-RI

No parameters

TP-BEGIN-TRANSACTION-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Chain	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Chain	M	Y	TPPM		

TP-ASSOCIATION-ESTABLISHMENT-RI

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
ProtocolVersion	M	Y	TPPM		
Contention winner assignment	M	Y	TPPM		
Bid-Mandatory	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		
Contention winner assignment	M	Y	TPPM		
Bid-Mandatory	M	Y	TPPM		

TP-ASSOCIATION-ESTABLISHMENT-RC

Sending

FIELD	STND	NIST	SOURCE	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		

Receiving

FIELD	STND	NIST	SINK	NIST RANGE	NOTES
Protocol Version	M	Y	TPPM		

ACSE SERVICE PARAMETERS

This section shows TP's use of ACSE services and parameters.

A-ASSOCIATE

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Mode	M	Y		
Application Context Name	M	Y		
Calling AP Title	M(A)			
Calling AE Qualifier	M(A)			
Calling AP Invocation Identifier	M			
Calling AE Invocation Identifier	M			
Called AP Title	C(A)			
Called AE Qualifier	C(A)			
Called AP Invocation Identifier	C(B)			
Called AE Invocation Identifier	C(B)			
Responding AP Title	M(A)			
Responding AE Qualifier	M(A)			
Responding AP Invocation Identifier	M(A)			
Responding AE Invocation Identifier	M(A)			
User Information	M	Y		
Result	M	Y		

Notes:

(A) Only if CCR is used, else parameter is a user option

(B) Parameter becomes mandatory if the association is being established for purposes (channels)

A-ASSOCIATE sending continued

PARAMETER	STND	NIST	NIST RANGE	NOTES
Diagnostic	O	O		
Calling Presentation Address	M	Y		
Called Presentation Address	M	Y		
Responding Presentation Address	O	O		
Presentation Context Definition List	M	Y		
Presentation Context Definition Result List	O	O		
Default Presentation Context Name	O	NU		
Default Presentation Context Result	O	NU		
Quality of Service	M	Y		
Presentation Requirements	M	Y	Kernel only	
Session Requirements	M	Y	Kernel + Full Duplex + CCR requirements (if used)	
Initial Synchronization point Serial Number	M(A)			
Initial Assignment of Tokens	M(A)			
Session-Connection Identifier	NU	NU		

Notes:

(A) Only if CCR is used, else parameter is a user option

(B) Parameter becomes mandatory if the association is being established for recovery purposes (channels)

A-ASSOCIATE

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Mode	M	Y		
Application Context Name	M	Y		
Calling AP Title	M(A)			
Calling AE Qualifier	M(A)			
Calling AP Invocation Identifier	M			
Calling AE Invocation Identifier	M			
Called AP Title	C(A)			
Called AE Qualifier	C(A)			
Called AP Invocation Identifier	C(B)			
Called AE Invocation Identifier	C(B)			
Responding AP Title	M(A)			
Responding AE Qualifier	M(A)			
Responding AP Invocation Identifier	M(A)			
Responding AE Invocation Identifier	M(A)			
User Information	M	Y		

Note:

(A) Only if CCR is used, else parameter is a user option

(B) Parameter becomes mandatory if the association is being established for recovery purposes (channels)

A-ASSOCIATE receiving continued

PARAMETER	STND	NIST	NIST RANGE	NOTES
Result	M	Y		
Result Source	M	Y		
Diagnostic	O	O		
Calling Presentation Address	M	Y		
Called Presentation Address	M	Y		
Responding Presentation Address	O	O		
Presentation Context Definition List	M	Y		
Presentation Context Definition Result List	O	Y		
Default Presentation Context Name	O	NU		
Default Presentation Context Result	O	NU		
Quality of Service	M	Y		
Presentation Requirements	M	Y	Kernel only	
Session Requirements	M	Y	Kernel + Full Duplex + CCR requirements (if used)	
Initial Synchronization Point Serial Number	M(A)			
Initial Assignment of Tokens	M(A)			
Session-Connection Identifier	NU	NU		

Notes:

(A) Only if CCR is used, else parameter is a user option

(B) Parameter becomes mandatory if the association is being established for recovery purposes (channels)

A-RELEASE

Sending (Request/Response)

Reason	NU	NU		
User information	NU	NU		
Result	M	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Reason	NU	NU		
User information	NU	NU		
Result	M	Y		

A-ABORT

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User Information	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Abort Source	M	Y		
User information	NU	NU		

A-P-ABORT

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Provider Reason	O	O		

PRESENTATION SERVICE PARAMETERS

This section shows TP's use of Presentation services and parameters.

P-TOKEN-PLEASE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens				1
User-data	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens				1
User-data	NU	NU		

Notes:

1. Why is there an inconsistancy in the token paramter of P-Token-Please and P-Token-Give.

P-TOKEN-GIVE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens	M	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Tokens	M			

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	M	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	M	Y		

CCR SERVICE PARAMETERS

This section shows TP's use of CCR services and parameters.

C-BEGIN

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Atomic Action Id.- Master's Name	M	Y		
Atomic Action Id.- Suffix	M	Y		1
Branch Id.-Suffix	M	Y		1
User Data	C	Y		

Receiving (Indication/Confirmation

PARAMETER	STND	NIST	NIST RANGE	NOTES
Atomic Action Id.- Master's Name	M	Y		
Atomic Action Id.- Suffix	M	Y		1
Branch Id.-Superior's Name	M	Y		
Branch Id.-Suffix	M	Y		1
User Data	C	Y		

Notes

1. Must decide which CCR ASN.1 Choice to use

C-PREPARE

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

C-READY

Sending (Request)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	NU	NU		

Receiving (Indication)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	NU	NU		

C-COMMIT

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

C-ROLLBACK

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
User-data	C	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Atomic Action Identifier	M	Y		
Branch Identifier	M	Y		
User-data	C	Y		

C-RECOVER

Sending (Request/Response)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Recovery State	M	Y		
Atomic Action Identifier	M	Y		
Branch Identifier	M	Y		
User-data	C	Y		

Receiving (Indication/Confirmation)

PARAMETER	STND	NIST	NIST RANGE	NOTES
Recovery State	M	Y		
Atomic Action Identifier	M	Y		
Branch Identifier	M	Y		
User-data	C	Y		

16. OFFICE DOCUMENT ARCHITECTURE

Editor's Note: For current Stable ODA Agreements, consult the aligned section of the Stable Implementation Agreements Document, Version 3, Edition 1, December 1989.

There is international alignment work progressing between the OIW, EWOS, and AOW on the Level 3 DAP (based on Chapter 16 in the Stable Document). As these alignment changes are completed, appropriate changes will be included in a revised Chapter 16. The current intention is to rename Chapter 16 to "Office Document Architecture Level 3 DAP."

17 INTRODUCTION

This is the definition of an implementation agreement for ODA that is based on a document application profile (DAP) named NIST ODA Level 2 DAP. These agreements will consist of a document application profile and a generator support statement (GSS)/receiver support statement (RSS) proforma. This document application profile is suitable for interchanging a document in formatted form, processable form or formatted processable form. This implementation agreement has been prepared by the ODA Special Interest Group of the NIST OSI Implementors Workshop (OIW). The document application profile portion of this implementation agreement is defined in accordance with ISO 8613-1 and CCITT T.411 and follows the standardized proforma and notation defined in ISO 8613-1 proposed Draft Addendum. Section 17 through 17.8 define the document application profile included in these agreements. Section 17.9 defines the GSS/RSS proforma included in these agreements. Additional sections contain informative material concerning these agreements.

Note:The document application profile defined by these agreements will be superceded by the equivalent internationally aligned document application profile when it is submitted for processing as an Internationally Aligned Profile (ISP).

17.1 SCOPE AND FIELD OF APPLICATION

This document \application profile specifies interchange formats for the transfer of structured documents between equipment designed for word or document processing. Such documents may contain characters, raster graphics and geometric graphics content.

The documents supported by this profile range from simple documents to structured technical reports, articles and typeset documents such as brochures. This profile provides a comprehensive level of features for the transfer of documents between these systems.

This document application profile describes documents which can be interchanged in the following form, as defined in ISO 8613:

- Formatted form,
- Processable form, and
- Formatted processable form.

The architecture level have matching functionalities so that the interchange formats of a document are convertible from a processable form into any other form.

This document application profile is independent of the processes carried out in an end system to create, edit or reproduce which, for example, may be by means of communication links or storage media.

17.2 REFERENCES

ISO 2022 Information Processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques

ISO 6937-1 Information Processing - Coded character sets for text communication - Part 1: General introduction

ISO 6937-2 Information Processing - Coded character sets for text communication - Part 2: Latin alphabetic and non-alphabetic graphic characters

ISO 8613-1 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 1: Introduction and General Principles

ISO 8613-2 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 2: Document Structures

ISO 8613-4 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 4: Document Profile

ISO 8613-5 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 5: Office Document Interchange Format

ISO 8613-6 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 6: Character Content Architecture

ISO 8613-7 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 7: Raster Graphics Content Architecture

ISO 8613-8 Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format - Part 8: Geometric Graphics Content Architecture

ISO 8613-1 PDAD ... "Document Application Profile Proforma and Notation" (to be published)

ISO 8632-1 Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture description information - Part 1: Functional Specification

ISO 8632-3 Information Processing Systems - Computer Graphics - Metafile for the storage and transfer of picture description information - Part 3: Binary Encoding

ISO 8859-1 Information Processing - 8-bit single byte coded graphic character sets - Part 1: Latin Alphabet No. 1

ISO 8859-7 Information Processing - 8-bit single byte coded graphic character sets - Part 7: Latin/Greek Alphabet

ISO 8824 Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation 1 (ASN.1)

ISO 8825 Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation 1 (ASN.1)

CCITT T.6 - Facsimile coding scheme and coding control functions for Group 4 Facsimile Apparatus, 1984

CCITT T.411 Open Document Architecture (ODA) and Interchange Format - Introduction and general principles, 1988

CCITT T.412 Open Document Architecture (ODA) and Interchange Format - Document structures, 1988

CCITT T.414 Open Document Architecture (ODA) and Interchange Format - Document profile, 1988

CCITT T.415 Open Document Architecture (ODA) and Interchange Format - Open document interchange format, 1988

CCITT T.416 Open Document Architecture (ODA) and Interchange Format - Character content architecture, 1988

CCITT T.417 Open Document Architecture (ODA) and Interchange Format - Raster graphics content architecture, 1988

CCITT T.418 Open Document Architecture (ODA) and Interchange Format - Geometric graphics content architecture, 1988

CCITT T.502 Document Application Profile PM.1 for the interchange of processable form documents

NIST ... Office Document Architecture Level 3 DAP, Stable Implementation Agreements for Open Systems Interconnection Protocols, Version 2 Edition 3, June 1989.

PrENV 41-509 ... Q111 ODA document application profile - processable and formatted documents - basic character content, October 1989.
PrENV 41-510 ... Q112 ODA document application profile - processable and formatted documents - enhanced mixed mode, October 1989.

PrENV ... Q113 ODA document application profile - processable and formattable document - extended mixed mode (to be published).
INTAP ... AE-1126 ODA document application profile ...

PAGODA ... CORE-11 ODA document application profile - processable and formatted documents - basic character content (to be published)

PAGODA ... CORE-26 ODA document application profile - processable and formatted documents - enhanced mixed mode (to be published)

PAGODA ... CORE-36 ODA document application profile - processable and formatted documents - extended mixed mode (to be published)

17.3 DEFINITIONS AND ABBREVIATIONS

The definitions given in ISO 8613-1 are applicable to this document.

The following additional definitions are applicable to this document.

Generating Support Statement (GSS)

A statement which states the range of support of an originating system. An originating system generates ODIF data streams. A GSS defines a subset of all possible data streams supported by an implementation which an origination capability. A GSS is specified by completing the GSSP defined in Annex A of this document.

Generating Support Statement Proforma (GSSP)

A definition of the conformance requirements of a profile in terms of a list of requirements for implementations to originate data streams which conform to the profile. A GSSP defines the format for all GSSs.

Receiving Support Statement (RSS)

A statement which states the range of support of a receiving system. A receiving system interprets ODIF data streams. A RSS defines functions and fall-backs supported by an implementation with a reception capability. A RSS is specified by completing the RSSP defined in Annex A of this document.

Receiving Support Statement Proforma (RSSP)

A definition of the conformance requirements of a profile in terms of a list of requirements, including fall-backs, for implementations to receive data streams which conform to the profile. A RSSP defines the format for all RSSs.

17.4 POSITION OF THIS DAP IN THE TAXONOMY OF RELATED DAPS

There are several regional activities involving the development of ODA document application profiles other than the NIST ODA SIG. These include the following:

- Asia-Oceania Workshop (AOW) ODA SIG

- CCITT Study Group VIII, Question 26

- European Workshop for Open Systems ODA EG

- Profile Alignment Group for ODA (PAGODA)

17.4.1 AOW ODA SIG

This document application profile is a functional subset of the AOW AE-1126 document application profile. This document application profile is a functional superset of the AOW AE-1111 and AE-1116 document application profiles.

17.4.2 CCITT SG VIII, Q26

This document application profile is expected to be a functional subset of the CCITT "pm3" document application profile. This document application profile is expected to be functionally equivalent to the CCITT "pm2" document application profile. This document application profile is a functional superset of the CCITT T.502 Recommendation.

17.4.3 EWOS ODA EG

This document application profile is expected to be a functional subset of the EWOS Q113 document application profile. This document application profile is a functional superset of the EWOS Q111 document application profile. This document application profile is expected to be equivalent to the EWOS Q112 document application profile.

17.4.4 NIST ODA SIG

This document application profile is a subset of the NIST Level 3 DAP.

17.4.5 PAGODA

There are three document application profiles developed by PAGODA for submission as ISPs. These are names Core-11, Core-26 and Core-36. This document application profile is intended to compatable with the Core-26 document application profile. This document application profile is intended to be a superset of the Core-11 document application profile. This document application profile is intended to be a subset of the Core-36 document application profile.

17.5 CONFORMANCE

In order to conform to this document application profile, a data stream representing a document must meet the requirements specified in subclause 1.

Subclause 2 specifies the requirements for implementations that originate and/or receive data streams conforming to this document application profile.

17.5.1 Data stream conformance

The following requirements apply to the encoding of data streams that conform to these agreements.

- The data stream shall be encoded in accordance with the ASN.1 encoding rules defined in ISO 8825,

- The data stream shall be structured in accordance with the interchange format defined in clause 8 of this document application profile,

- The encoded document shall be structured in accordance with one of the document architecture classes specified in clause 7 of this document application profile. In addition, the encoded document shall contain all required constituents specified for that class and contain only constituents permitted or required for that class as specified in clause 7 of this document application profile,

- The encoded constituents shall contain all required attributes as specified in clause 7 of this document application profile,

- The encoded attributes shall have values within the range of permissible values specified in clause 7 of this document application profile,

- The encoded document shall be structured in accordance with the abstract document architecture defined in ISO 8613,

- The encoded document shall be structured in accordance with the characteristics defined in clause 6 of this document application profile.

17.5.2 Implementation conformance

This clause states the requirements for implementations claiming conformance to this document application profile.

An implementation claiming to originate and/or receive data streams conforming to this document application profile must complete a Generator Support Statement (GSS) and/or Receiver Support Statement (RSS) Proforma as defined in section 17.9 of this document application profile.

A conforming receiving implementation must be capable of receiving <u>any</u> data stream conforming to this document application profile. "Receiving" means not rejecting a data stream conforming to this document application profile and usually, but not always, involves recognizing and further processing the data stream elements. The explicit meaning of "receiving" is determined by a RSS defined in accordance with section 17.9 of this document application profile.

17.6 CHARACTERISTICS SUPPORTED BY THIS DAP

Note: Text corresponding to the logical and layout constraint objects defined in section 17.7 will be defined for this section.

17.7 SPECIFICATION OF CONSTITUENT CONSTRAINTS

17.7.1 Document profile constraints

17.7.1.1Macro definitions

```
DEFINE(BASIC-CHAR-SET,"{
        {ESC 02/08 F1, LS0};
--  Designates a version of ISO 646 to G0 and invokes to GL  --
--  F1 should be the final character of a version of ISO 646.  ASCII
(final character 04/02) should be "Rwof" in the GSS/RSS Proforma.  --
")

DEFINE(NON-BASIC-CHAR-SET,"
        ESC 02/09 F2, LS1R |
--  Designates 94 single byte set to G1 and invokes to GR  --
        ESC 02/04 02/09 F3, LS1R |
--  Designates 94 multi-byte set to G1 and invokes to GR  --
        ESC 02/10 F4, LS2R |
--  Designates 94 single byte set to G2 and invokes to GR  --
        ESC 02/04 02/10 F5, LS2R |
--  Designates 94 multi-byte set to G2 and invokes to GR  --
        ESC 02/11 F6, LS3R |
--  Designates 94 single byte set to G3 and invokes to GR  --
        ESC 02/04 02/11 F7, LS3R |
--  Designates 94 multi-byte set to G3 and invokes to GR  --
        ESC 02/13 F8, LS1R |
--  Designates 96 single byte set to G1 and invokes to GR  --
        ESC 02/04 02/13 F9, LS1R |
--  Designates 96 multi-byte set to G1 and invokes to GR  --
        ESC 02/14 F10, LS2R |
--  Designates 96 single byte set to G2 and invokes to GR  --
        ESC 02/04 02/14 F11, LS2R |
--  Designates 96 multi-byte set to G2 and invokes to GR  --
        ESC 02/15 F12, LS3R |
--  Designates 96 single byte set to G3 and invokes to GR  --
        ESC 02/04 02/15 F13, LS3R
--  Designates 96 multi-byte set to G3 and invokes to GR

--  Fn is defined in the "International Register of Coded Character Sets
to be used with Escape Sequences".  The empty sets (final character
07/14) should be designated and invoked in GR if there are no further
requirements on characters other than those designated in G0 set.  --
")

DEFINE(CODE-EXT-ANNOUNCERS,"
        [ESC 02/00 05/00 |
         ESC 02/00 05/03 |
         ESC 02/00 05/05 |
         ESC 02/00 05/07 |
         ESC 02/00 05/10 |
         ESC 02/00 05/11]+
")
```

```
DEFINE(BASIC-SUBREPERTOIRES,"
        2 |
        --  Minimal  --
        5 |
        --  Unique characters allocated in ISO 646  --
        8
        -- ISO 8859-1 subrepertoire --
")

DEFINE(NON-BASIC-SUBREPERTOIRES,"
        1 |
        --  Full  --
        3 |
        --  Teletex  --
        4 |
        --  Videotex  --
        7 },
        --  Western European Typeset  --
")

DEFINE(BASIC-PAG-DIM,"
--  Common Assured Reproduction Areas (CARA)  --
   #horizontal{9240..39732},#vertical{12400..56173}, |
--  CARA of ISO A4 and ANSI A portrait <= ISO A0 portrait  --
   #horizontal{12400..56173},#vertical{9240..39732}, |
--  CARA of ISO A4 and ANSI A landscape <= ISO A0 landscape  --
   #horizontal{9240..39600},#vertical{12400..52200}, |
--  CARA of ISO A4 and ANSI A portrait <= ARA ANSI E portrait  --
   #horizontal{12400..52200},#vertical{9240..39600}
--  CARA of ISO A4 and ANSI A landscape <= ARA ANSI E landscape  --
")

DEFINE(NON-BASIC-PAG-DIM,"
--  Assured Reproduction Areas (ARA)  --

   {#horizontal{13200},#vertical{18480}
--  ARA ISO A3 Portrait (279mm x 391mm)  --
   {#horizontal{18840},#vertical{13200}|
--  ARA ISO A3 Landscape (420mm x 297mm)  --
   {#horizontal{12744},#vertical{19656}|
--  ARA ANSI B Portrait (10.62in x 16.38in)  --
   {#horizontal{19656},#vertical{12744}
--  ARA ASNI B Landscape (16.38in x 10.62in)  -- |

-- Full Page Sizes --

   {#horizontal{14031},#vertical{19843}|
--  ISO A3 Portrait (297mm x 420mm)  --
   {#horizontal{19843},#vertical{14031}|
--  ISO A3 Landscape (420mm x 297mm)  --
   {#horizontal{13200},#vertical{20400}|
--  ANSI B Portrait (11in x 17in)  --
   {#horizontal{20400},#vertical{13200}
--  ASNI B Landscape (17in x 11in)  --
")

DEFINE(NON-BASIC-NOM-PAG-SIZ,"
   {#horizontal{14031},#vertical{19843}|
--  ISO A3 Portrait (297mm x 420mm)  --
   {#horizontal{19843},#vertical{14031}|
--  ISO A3 Landscape (420mm x 297mm)  --
   {#horizontal{13200},#vertical{20400}|
--  ANSI B Portrait (11in x 17in)  --
   {#horizontal{20400},#vertical{13200}
--  ASNI B Landscape (17in x 11in)  --
")

DEFINE(BASIC-CHAR-ORIENTATION,"
   {'0-degrees'}
")

DEFINE(NON-BASIC-CHAR-ORIENTATION,"
   {'90-degrees')

DEFINE(BASIC-CHAR-PATH,"
   {'0-degrees' | '90-degrees'
")
DEFINE(NON-BASIC-CHAR-PATH,"
   {'180-degrees' | '270-degrees'}
")

DEFINE(FDA,"formatted (0)")
DEFINE(PDA,"processable (1)")
DEFINE(FPDA,"formatted-processable (2)")

DEFINE(DAC,"
Document-profile{#Document-characteristics
 {#Document-architecture-class}}  ")

DEFINE(CF,"{2 8 2 6 0}")--  Character formatted  --
DEFINE(CP,"{2 8 2 6 1}")--  Character processable  --
DEFINE(CFP,"{2 8 2 6 2}")--  Character formatted processable --
DEFINE(RFP,"{2 8 2 7 2}")--  Raster formatted processable  --
DEFINE(GFP,"{2 8 2 8 0}")--  Graphics formatted processable  --

DEFINE(FACTOR,"factor-set (2)")
DEFINE(COMPLETE,"complete-generator-set (1)")
DEFINE(PRESENT,"present (1)")
```

17.7.1.2Constituent constraints

17.7.1.2.1Presence of document constituents

```
CASE    {$DAC OF
$FDA:
PERM    Generic-layout-structure($FACTOR);
REQ     Specific-layout-structure($PRESENT);
PERM    Presentation-styles($PRESENT);

$PDA:
PERM    Generic-layout-structure($COMPLETE);
REQ     Generic-logical-structure($COMPLETE);
REQ     Specific-logical-structure($PRESENT);
PERM    Presentation-styles($PRESENT);
PERM    Layout-styles($PRESENT);

$FPDA:
REQ     Generic-layout-structure($COMPLETE);
REQ     Specific-layout-structure($PRESENT);
REQ     Generic-logical-structure   ($COMPLETE);
REQ     Specific-logical-structure($PRESENT);
PERM    Presentation-styles($PRESENT);
PERM    Layout-styles   ($PRESENT);
)

PERM    External-document-class(ANY_VALUE);
PERM    Resource-document(ANY_VALUE);
PERM    Resources   (ANY_VALUE);
```

17.7.1.2.2Document characteristics

```
REQ     Document-application-profile(1 3 14 11 0 1 0);

REQ     Doc-appl-profile-defaults( ( REQ
        #Document-architecture-defaults(

CASE    ($DAC OF

$FDA:
PERM    #Content-architecture-class($FC),

$PDA:
REQ     #Content-architecture-class($PC),

$FPDA:
REQ     #Content-architecture-class($FPC),
)

PERM    #Page-dimensions($BASIC-PAG-DIM | $NON-BASIC-PAG-DIM |
$NON-BASIC-NOM-PAG-SIZE),
PERM    #Medium-type(
   REQ  #Nominal-page-size($BASIC-PAG-DIM | $NON-BASIC-PAG-DIM),
   REQ  #Side-of-sheet(ANY_VALUE)),
PERM    #Character-contents-defaults(
   PERM #Character-path($BASIC-CHAR-PATH | $NON-BASIC-CHAR-PATH),
   PERM #Character-orientation{$BASIC-CHAR-ORIENTATION |
$NON-BASIC-CHAR-ORIENTATION},
   PERM #Character-spacing{ANY_VALUE},
   PERM #Line-spacing{ANY_VALUE},
   PERM #Graphic-rendition{ANY_EXCEPT 26,
        --  Variable spacing  --
        50},
        --  Not variable spacing  --
   PERM #Graphic-char-subrepertoire
        {$BASIC-SUBREPERTOIRES | $NON-BASIC-SUBREPERTOIRES},
   PERM #Widow-size{ANY_VALUE},
   PERM #Orphan-size{ANY_VALUE},
   PERM #Graphic-character-sets{{$BASIC-CHAR-SET |
$NON-BASIC-CHAR-SET}+},
   PERM #Indentation{ANY_VALUE},
   PERM #Kerning-offset{ANY_VALUE},
   PERM #Proportional-line-spacing
        {ANY_VALUE},
   PERM #Pair-wise-kerning{ANY_VALUE},
   PERM #Code-extension-announcers
        {{$CODE-EXT-ANNOUNCERS}+}},
   --  Note:First-line-offset not permitted here.  --
PERM    Raster-gr-contents-defaults{
   PERM #Pel-path{'0-degrees' | '180-degrees'},
   PERM #Line-progression{'90-degrees' |'270-degrees'},
   PERM #Pel-spacing{ANY_VALUE <1200},
   PERM #Compression{ANY_VALUE},
   PERM #Pel-bit-order{'up' | 'down'}},
   --  Note:Inclusion presumes approval by ISO.  --
PERM    Geo-gr-contents-defaults{ANY_VALUE}
};

REQ     Document-architecture-class   {$FDA | $PDA | $FPDA};
REQ     Content-architecture-class   {$CF | $CP | $CFP | $RFP | $GFP};
REQ     Interchange-format-class{if-a (0)};
REQ     ODA-version    {
   REQ  #standard-or-recommendation{"ISO 8613"},
   REQ  #publication-date{"1989-07-04"} };

REQ     Non-basic-doc-characteristics{ {
   REQ  #Profile-character-sets {{$BASIC-CHAR-SET |
$NON-BASIC-CHAR-SET}+},
--   "Profile-character-sets" designate and invoke character sets used in
attributes to which "Profile-character-sets" is applied.  --
   REQ  #Comment-character-sets{{$BASIC-CHAR-SET |
$NON-BASIC-CHAR-SET}+},
--  "Comment-character-sets" specifies the initial designated graphic
character sets and shift status of "User-readable-comments" and
"User-visible-name".  Designation to the same G set overrides the
previous designated graphic character set in "Comment-character-sets".
All the graphic character sets used in "User-readable-comments" and
```

```
"User-visible-name" should be designated and/or invoked in
"Comment-character-sets".  --
   REQ  #Alternative-representation-character-sets
            {{$BASIC-CHAR-SET | $NON-BASIC-CHAR-SET},
--  "Alternate-representation-character-sets" designate and invoke
character sets used in attributes to which
"Alternate-representation-character-sets" is applied.  --
   REQ  #Page-dimensions{$BASIC-PAG-DIM | $NON-BASIC-PAG-DIM},
   REQ  #Medium-types{
        REQ#Nominal-page-size
        {$NON-BASIC-NOM-PAG-SIZ},
        REQ#Side-of-sheet{ANY_VALUE},
   PERM #Char-presentation-features{
        PERM#Character-path{{$NON-BASIC-CHAR-PATH}+ };
        PERM#Character-orientation
        {$NON-BASIC-CHAR-ORIENTATION};
        PERM#Character-spacing{ANY_INTEGER <100},
        PERM#Line-spacing{100 | 150 | ANY_INTEGER > 200},
        PERM#Graphic-char-subrepertoire
        { {$NON-BASIC-SUBREPERTOIRES}+},
        PERM#Graphic-character-sets
        {{$BASIC-CHAR-SET | $NON-BASIC-CHAR-SET}+},
   PERM #Ra-gr-presentation-features{
        PERM#Pel-path{'180-degrees'},
        PERM#Line-progression{'90-degrees'},
        PERM#Compression{'uncompressed'}}
};

PERM    Additional-doc-characteristics{
   REQ  #Fonts-list{ANY_VALUE},
   PERM #Unit-scaling{ANY_VALUE} };
```

17.7.1.2.3Document management attributes

```
PERM    Document-description{ANY_VALUE};
PERM    Dates-and-times{ANY_VALUE};
PERM    Originators{ANY_VALUE};
PERM    Other-user-information{ANY_VALUE};
PERM    External-references{ANY_VALUE};
PERM    Local-file-references{ANY_VALUE};
PERM    Content-attributes{ANY_VALUE};
PERM    Security-information{ANY_VALUE};
```

17.7.2 Logical constituent constraints

Note:The production rules for the Generator-for-subordinates for the logical constraint objects has not as yet been aligned with the notation used in the PAGODA DAPs.

17.7.2.1Diagrams of relationships of logical constituents

The notation used for the structure diagrams is that specified in Appendix A of ISO 8613-2.

The following diagrams represent the primary graph for the complete generator set of logical object class descriptions.

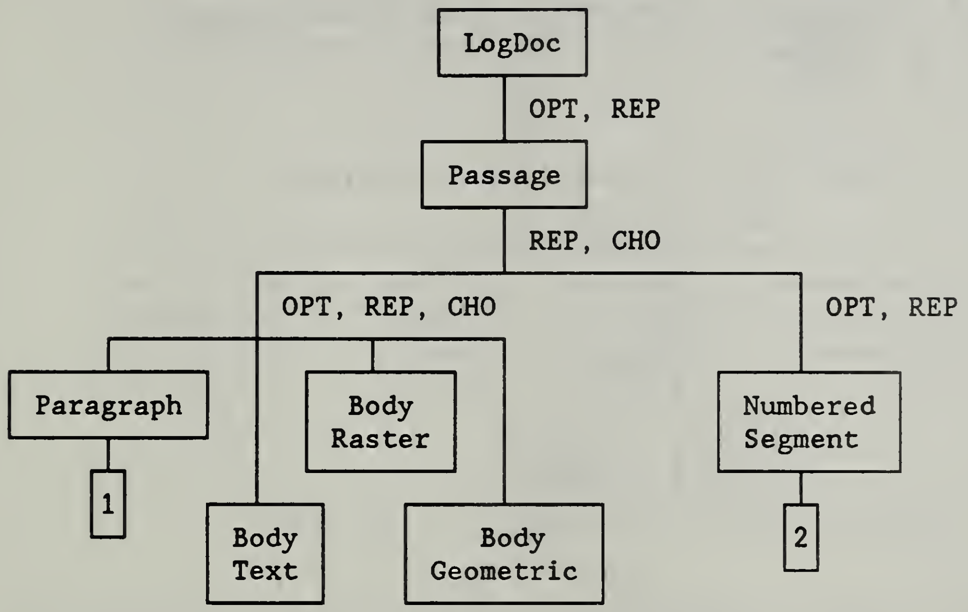


Figure 17.4: Structure for LogDoc and Passage

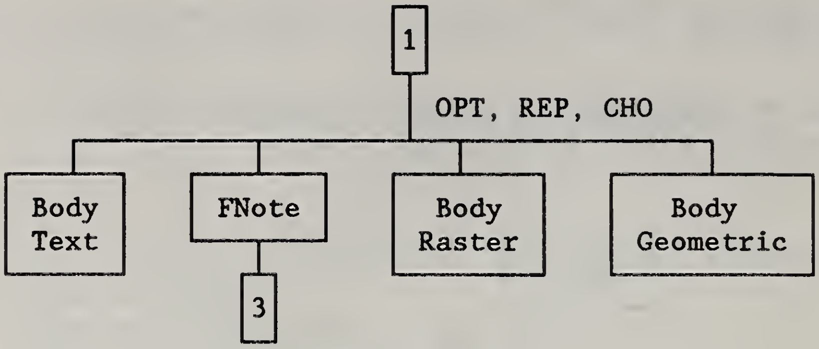


Figure 17.5: Structure for Paragraph

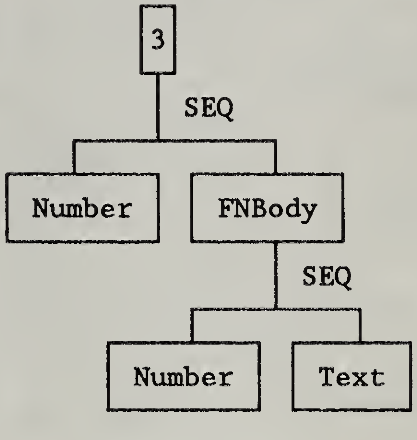


Figure 17.6: Structure for FNote

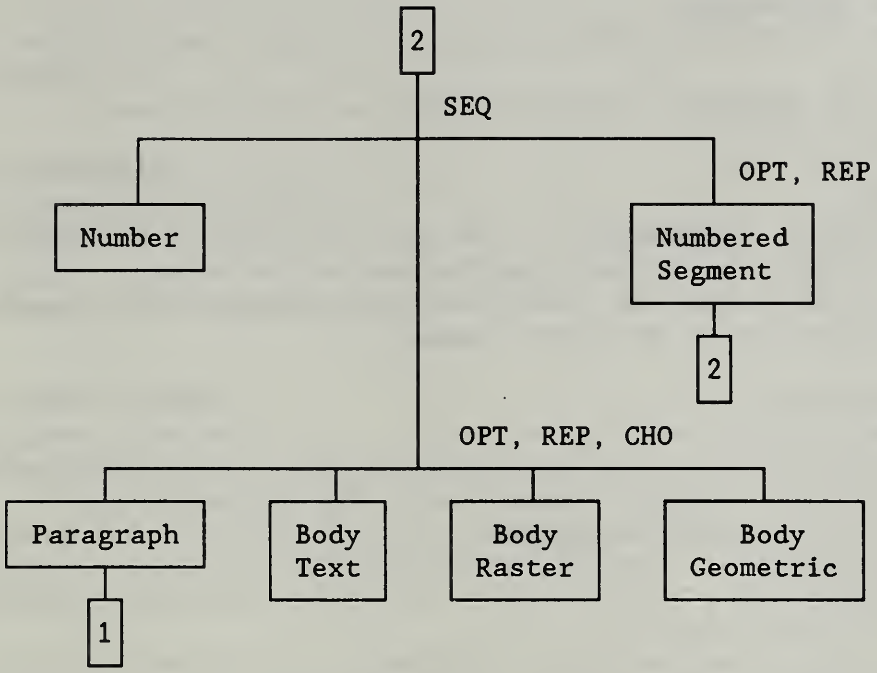


Figure 17.7: Structure for NumberedSegment

The following diagram corresponds to the logical object class descriptions referenced by the attribute "Logical Source" in layout components.

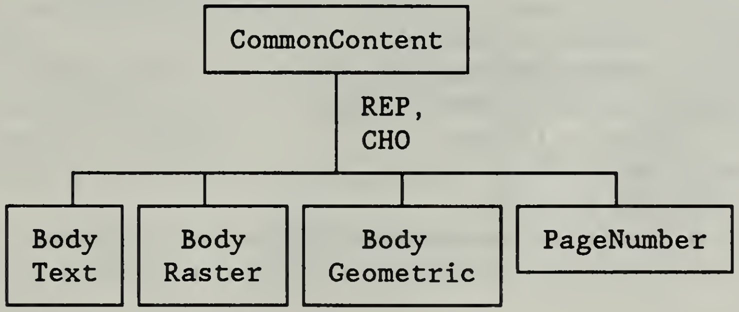


Figure 17.8: Structure for CommonContent

17.7.2.2Macro definitions

```
DEFINE(N,"
<n>     ::=--any character string from the set of characters:
"0","1",..."9"-- ")

DEFINE(NUMBERS,"
<numbers>::="number-"+<$N> ")
   --  This binding can be instanced for use as the numeric values for
   use in a segment number or footnote number bindings.  The instances
   are differentiated by the suffix number.  --

DEFINE(NUMBERSTRINGS,"
<numberstrings>::="numberstring-"+<$N> ")
   --  This binding can be instanced for use as the string value for the
   segment number or footnote number text.  The instances are
   differentiated by the suffix number.  --

DEFINE(PREFIXES,"
<prefixes>::="prefix-"+<$N> ")

DEFINE(SUFFIXES,"
<suffixes>::="suffix-"+<$N> ")

DEFINE(SEPARATORS,"
<separators>::="separator-"+<$N> ")

DEFINE(STRINGFUNCTION,"
<string-function>::=MK_STR | U_ALPHA | L_ALPHA | U_ROM | L_ROM | ' 'H ")

DEFINE(INITIALISEANY,"
<binding-pair-constraint> ::=
        <$PREFIXES>, STRING_LITERAL |
        <$SUFFIXES>, STRING_LITERAL |
        <$SEPARATORS>, STRING_LITERAL |
        <$NUMBERS>, NUMERIC_LITERAL |
        <$NUMBERSTRINGS>, " " |
        "PGnum", NUMERIC_LITERAL
   --  Used to initialize any of the bindings.  --
")

DEFINE(USENUMBERSTRING,"
<binding-pair-constraint>::=
   <$NUMBERSTRINGS>, <hierarchic-expr> | <simple-expr>

<hierarchic-expr>::=
   B_REF(SUP(CURR_OBJ)) (<$NUMBERSTRINGS>) +B_REF(SUP(CURR_OBJ))
   (<$SEPARATORS>)+<simple-expr)")

<simple-expr>::=
   <$STRINGFUNCTION> (B_REF(CURR_OBJ)($NUMBERS)) |
   <$STRINGFUNCTION> (ORD(CURR_OBJ)) | STRING-LITERAL
   --  Used to make a simple or compound string out of the number
   bindings.  --
")

DEFINE(USENUMBERS,"
<binding-pair-constraint>::=
   <$NUMBERS>,INC(B_REF(PREC(CURR_OBJ)) (<$NUMBERS>)

   --  Used to increment any of the number bindings.  --
")

DEFINE(SEGMENTNUMBER,"
<string-expr-constraint>::=
   [<sgpre-str>]+<sgnum-str>+[<sgsuf-str>]

<sgnum-str>::=B_REF(SUP(CURR_OBJ)) (<$NUMBERSTRINGS>)
<sgpre-str>::=B_REF(SUP(CURR_OBJ)) (<$PREFIXES>) | STRING_LITERAL
<sgsuf-str>::=B_REF(SUP(CURR_OBJ)) (<$SUFFIXES>) | STRING_LITERAL

   --  This expression is allowed in content generators for the Number
   constraint object to automatically generate text for segment numbers.
   --
")

DEFINE(PGNUMBER,"
<string-expr-constraint>::=
   [<pgpre-str>]+<pgnum-str>+[<pgsuf-str>]

<pgpre-str>::=STRING_LITERAL
<pgsuf-str>::=STRING_LITERAL
<pgnum-str>::=<$STRINGFUNCTION> (<numeric-expr>)
<numeric-expr>::=
   B_REF(SUP(CURR_INST( <class-or-type1>, CURR_OBJ)))  ("PGnum") |
   B_REF(CURR_INST(<class-or-type2>, CURR_OBJ)) ("PGnum")
<class-or-type-1>::=FRAME
<class-or-type-2>::=PAGE | OBJECT_CLASS_ID_OF(Page | RPage | VPage)

   --  This expression is alloed in content generators for the
   PageNumber constraint object to automatically generate text for page
   numbers.  --
")

DEFINE(FNNUMBER,"
<string-expr-constraint>::=
   [<fnpre-str>]+<fnnum-str>+[<fnsuf-str>]

<fnnum-str>::=B_REF(SUP(CURR_OBJ)) (<$NUMBERSTRINGS>) | STRING_LITERAL
<fnpre-str>::=B_REF(SUP(CURR_OBJ)) (<$SUFFIXES>) | STRING_LITERAL

   --  This expression is allowed in content generators for the Number
   constraint object to automatically generate text for footnote numbers.
   --
")

DEFINE(LOGDOCGFS,"
<constr-expr>::=OPT(REP(OBJECT_CLASS_ID(Passage)))  ")

DEFINE(PASSAGEGFS,"
<constr-expr>::=REP(CHO(OPT(REP(CHO(OBJECT_CLASS_ID(BodyText) |
OBJECT_CLASS_ID(BodyRaster) | OBJECT_CLASS_ID(BodyGeometric) |
OBJECT_CLASS_ID(Paragraph)))) |
OPT(REP(OBJECT_CLASS_ID(NumberedSegment))))  ")

DEFINE(NUMBEREDSEGMENTGFS,"
<constr-expr>::=SEQ(OBJECT_CLASS_ID(Number),
OPT(REP(CHO(OBJECT_CLASS_ID(BodyText) | OBJECT_CLASS_ID(BodyRaster) |
OBJECT_CLASS_ID(BodyGeometric) | OBJECT_CLASS_ID(Paragraph)))),
OPT(REP(OBJECT_CLASS_ID(NumberedSegment))))  ")

DEFINE(PARAGRAPHGFS,"
<constr-expr>::=OPT(REP(CHO(OBJECT_CLASS_ID(BodyText) |
OBJECT_CLASS_ID(BodyRaster) | OBJECT_CLASS_ID(BodyGeometric) |
OBJECT_CLASS_ID(FNote))))  ")

DEFINE(FNOTEGFS,"
<constr-expr>::=SEQ(OBJECT_CLASS_ID(Number), OBJECT_CLASS_ID(FNBody))  ")

DEFINE(FNBODYGFS,"
<constr-expr>::=SEQ(OBJECT_CLASS_ID(Number), OBJECT_CLASS_ID(BodyText))
")

DEFINE(COMMONCONTENTGFS,"
<constr-expr>::=REP(CHO(OBJECT_CLASS_ID(CommonText) |
OBJECT_CLASS_ID(CommonRaster) | OBJECT_CLASS_ID(CommonGeometric) |
OBJECT_CLASS_ID(PageNumber)))  ")
```

17.7.2.3Factor constraints

```
FACTOR: ANY-LOGICAL{

GENERIC:
REQ     Object-type{VIRTUAL};
REQ     Object-class-identifier{ANY_VALUE};
PERM    Resource{ANY_VALUE};

SPECIFIC:
PERM    Object-type{VIRTUAL};
REQ     Object-identifier{ANY_VALUE};
REQ     Object-class{VIRTUAL};

SPECIFIC_AND_GENERIC:
PERM    Layout-style{VIRTUAL};
PERM    Protection{ANY_VALUE};
PERM    User-readable-comment{ANY_VALUE};
PERM    User-visible-name{ANY_VALUE};
}

FACTOR: COMP-LOGICAL:ANY-LOGICAL{

GENERIC:
REQ     Object-type{COMPOSITE_LOGICAL_OBJECT};

SPECIFIC:
REQ     Subordinates{VIRTUAL};
PERM    Object-type{COMPOSITE_LOGICAL_OBJECT};

SPECIFIC_AND_GENERIC:
PERM    Layout-style{STYLE_ID_OF(LStyle3)};
PERM    Default-value-lists{ANY_VALUE};
}

FACTOR: BASIC-LOGICAL:ANY-LOGICAL{

GENERIC:
REQ     Object-type{BASIC_LOGICAL_OBJECT};
SPECIFIC:
PERM    Object-type{BASIC_LOGICAL_OBJECT};
PERM    Content-portions{ANY_VALUE};
}

FACTOR: ANY-COMMON{

GENERIC:
REQ     Object-type{VIRTUAL};
REQ     Object-class-identifier{ANY_VALUE};
PERM    Resource{ANY_VALUE};
PERM    Bindings{VIRTUAL};
PERM    Protection{ANY_VALUE};
PERM    User-readable-comments{ANY_VALUE};
PERM    User-visible-name{ANY];
}
```

17.7.2.4Constituent constraints

17.7.2.4.1LogDoc:ANY-LOGICAL{

```
GENERIC:
REQ     Object-type{DOCUMENT_LOGICAL_ROOT};
REQ     Generator-for-subordinates{$LOGDOCGFS};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(Logdoc)};
REQ     Subordinates{{SUBORDINATE_ID_OF( Passage)+}};
PERM    Object-type{DOCUMENT_LOGICAL_ROOT}

SPECIFIC_AND_GENERIC:
PERM    Layout-style{STYLE_ID_OF(LStyle1)};
PERM    Bindings{$INITIALISEANY};
PERM    Default-value-lists{ANY_VALUE};
PERM    Application-comments{"LogDoc"};
}
```

17.7.2.4.2Passage:COMP-LOGICAL{

```
GENERIC:
REQ     Generator-for-subordinates{$PASSAGEGFS};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(Passage)};
REQ     Subordinates{{[SUBORDINATE_ID_OF( Paragraph) |
SUBORDINATE_ID_OF(BodyText) | SUBORDINATE_ID_OF( BodyRaster) |
SUBORDINATE_ID_OF( BodyGeometric)}+] | [{SUBORDINATE_ID_OF(
NumberedSegment)}+};

SPECIFIC_AND_GENERIC:
PERM    Bindings{$INITIALISEANY | $USENUMBERS};
PERM    Application-comments{"Passage"};
}
```

17.7.2.4.3NumberedSegment:COMP-LOGICAL{

```
GENERIC:
REQ     Generator-for-subordinates{$NUMBEREDSEGMENTGFS};
REQ     Bindings{$USENUMBERS};
REQ     Application-comments{"NumberedSegment"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( NumberedSegment)};
REQ     Subordinates{SUBORDINATE_ID_OF(Number),
{[SUBORDINATE_ID_OF(BodyText) | SUBORDINATE_ID_OF( BodyRaster) |
SUBORDINATE_ID_OF( BodyGeometric) | SUBORDINATE_ID_OF( Paragraph)]}+,
[{SUBORDINATE_ID_OF( NumberedSegment)}+]};
PERM    Bindings{$INITIALISEANY | $USENUMBERS};
PERM    Application-comments{"NumberedSegment"};
}
```

17.7.2.4.4Number:BASIC-LOGICAL{

```
GENERIC:
REQ     Content-generator{$SEGMENTNUMBER | $FNNUMBER};
REQ     Application-comments{"Number"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(Number)};
PERM    Application-comments{"Number"};

SPECIFIC_AND_GENERIC:
PERM    Presentation-style{STYLE_ID_OF(PStyle1)};
PERM    Layout-style{STYLE_ID_OF(LStyle4)};
PERM    Content-architecture-class{$CF | $CP | $CFP};
}
```

17.7.2.4.5Paragraph:COMP-LOGICAL{

```
GENERIC:
REQ     Generator-for-subordinates{$PARAGRAPHGFS};
REQ     Application-comments{"Paragraph"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( Paragraph)};
REQ     Subordinates{[{SUBORDINATE_ID_OF(BodyText) | SUBORDINATE_ID_OF(
BodyRaster) | SUBORDINATE_ID_OF( BodyGeometric)}+]};
PERM    Application-comments{"Paragraph"};
}
```

17.7.2.4.6FNote:COMP-LOGICAL{

```
GENERIC:
REQ     Generator-for-subordinates{$FNOTEGFS};
REQ     Application-comments{"FNote"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(FNote)};
REQ     Subordinates{SUBORDINATE_ID_OF(Number),
SUBORDINATE_ID_OF(FNBody)};
PERM    Application-comments{"FNote"};

SPECIFIC_AND_GENERIC:
PERM    Bindings{$INITIALISEANY | $USENUMBERS};
}
```

17.7.2.4.7FNBody:COMP-LOGICAL{

```
GENERIC:
REQ     Generator-for-subordinates{$FNBODYGFS};
REQ     Application-comments{"FNBody"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(FNBody)};
REQ     Subordinates{SUBORDINATE_ID_OF(Number),
SUBORDINATE_ID_OF(BodyText)};
PERM    Application-comments{"FNBody"};
}
```

17.7.2.4.8**BodyText**:BASIC-LOGICAL{

```
GENERIC:
REQ     Application-comments{"BodyText"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( BodyText)};
PERM    Application-comments{"BodyText"};

SPECIFIC_AND_GENERIC:
PERM    Content-architecture-class{$CF | $CP | $CFP};
PERM    Content-portions{ANY_VALUE};
PERM    Presentation-style{STYLE_ID_OF(PStyle2)};
PERM    Layout-style{STYLE_ID_OF(LStyle5)};
}
```

17.7.2.4.9**BodyRaster**:BASIC-LOGICAL{

```
GENERIC:
REQ     Application-comments{"BodyRaster"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( BodyRaster)};
PERM    Application-comments{"BodyRaster"};

SPECIFIC_AND_GENERIC:
PERM    Content-architecture-class{$RFP};
PERM    Content-portions{ANY_VALUE};
PERM    Presentation-style{STYLE_ID_OF(PStyle3)};
PERM    Layout-style{STYLE_ID_OF(LStyle6)};
}
```

17.7.2.4.10**BodyGeometric**:BASIC-LOGICAL{

```
GENERIC:
REQ     Application-comments{"BodyGeometric"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( BodyGeometric)};
PERM    Application-comments{"BodyGeometric"};

SPECIFIC_AND_GENERIC:
```

```
PERM    Content-architecture-class{$GFP};
PERM    Content-portions{ANY_VALUE};
PERM    Presentation-style{STYLE_ID_OF(PStyle4)};
PERM    Layout-style{STYLE_ID_OF(LStyle6)};
}
```

17.7.2.4.11CommonContent:ANY-COMMON{

```
GENERIC:
REQ     Object-type{COMPOSITE_LOGICAL_OBJECT};
REQ     Generator-for-subordinates{COMMONCONTENTGFS};
REQ     Application-comments{"CommonContent"};
PERM    Default-value-list{ANY_VALUE};
}
```

17.7.2.4.12PageNumber:ANY-COMMON{

```
GENERIC:
REQ     Object-type{BASIC_LOGICAL_OBJECT};
REQ     Content-generator{$PGNUMBER};
PERM    Presentation-style{STYLE_ID_OF(PStyle2)};
PERM    Content-architecture-class{$CP};
PERM    Layout-style{STYLE_ID_OF(LStyle2)};
PERM    Application-comments{"PageNumber"};
}
```

17.7.3 Layout constituent constraints

Note:The production rules for the Generator-for-subordinates for the layout constraint objects has not as yet been aligned with the notation used in the PAGODA DAPs.

17.7.3.1Diagrams of relationships of layout constituents

The notation used for the structure diagrams is that specified in Appendix A of ISO 8613-2.

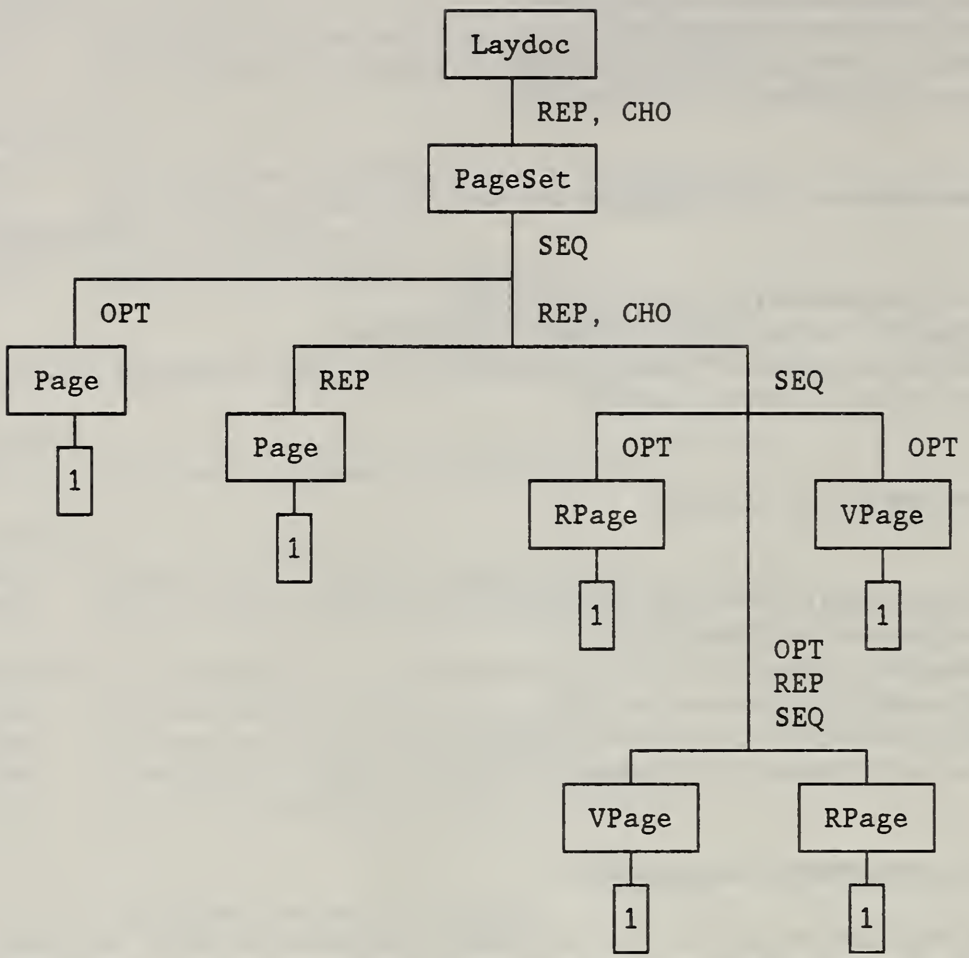


Figure 17.9: Structure for LayDoc and PageSet

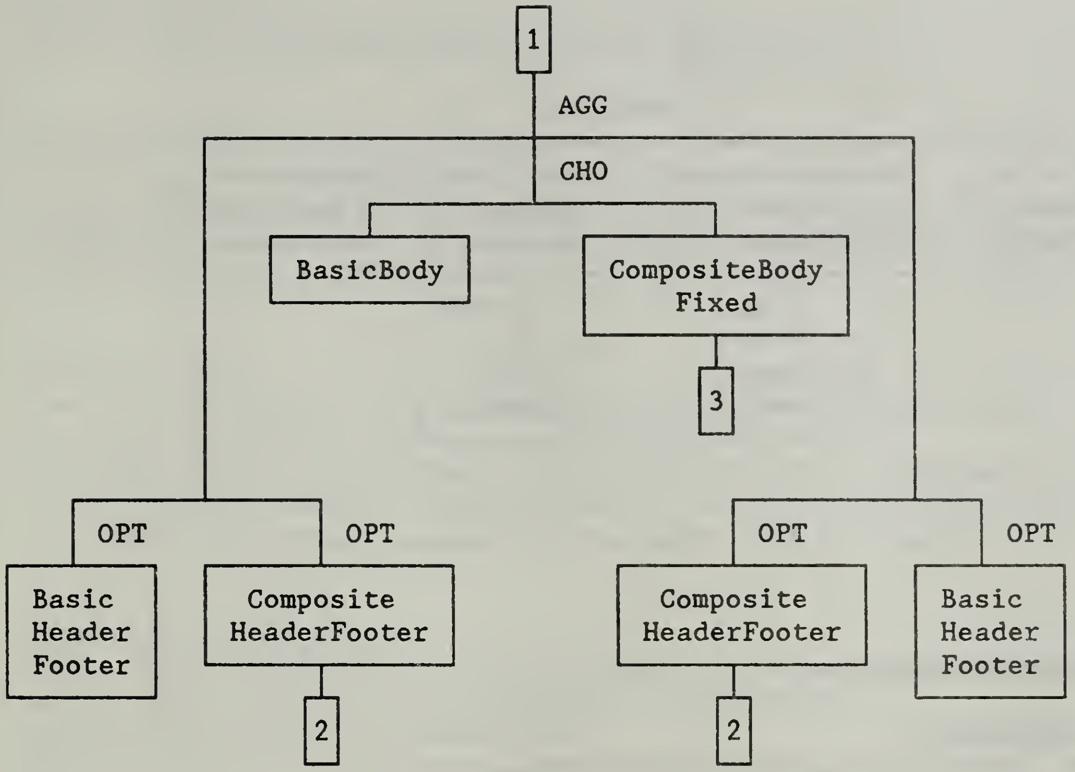


Figure 17.10: Structure for Page, VPage and RPage

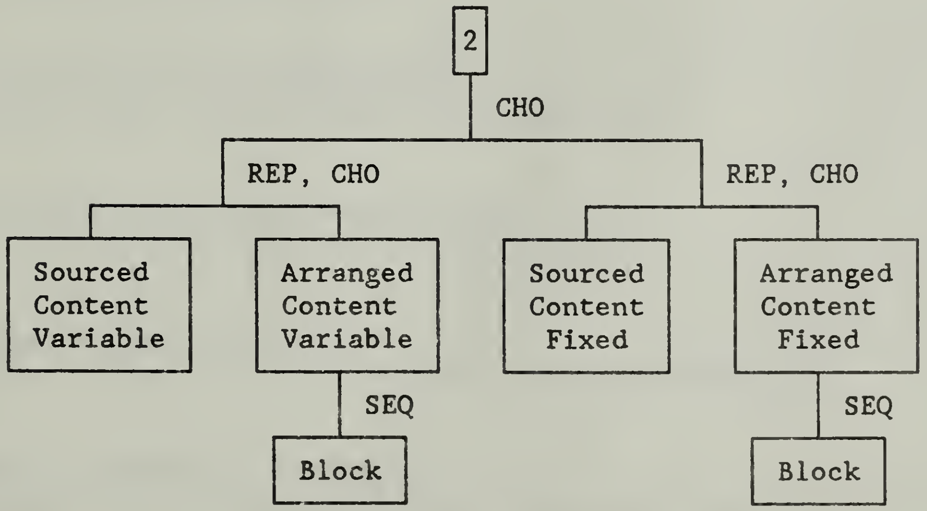


Figure 7.11: Structure for CompositeHeaderFooter

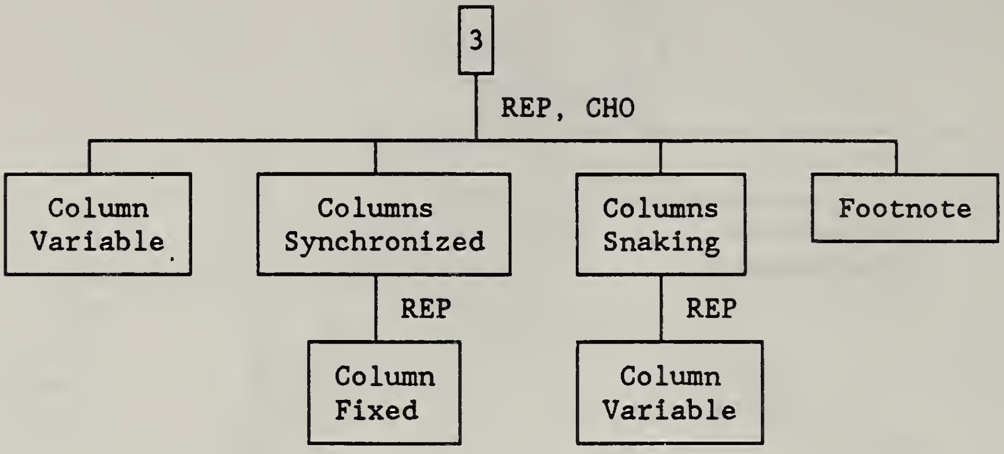


Figure 17.12: Structure for CompositeBody

17.7.3.2Macro definitions

```
DEFINE(USEPGNUMBER,"
   "PGnum", INC(B_REF(PREC(CURR_OBJ)) ("PGnum") ")
```

```
DEFINE(LAYDOCGFS,"
<constr-expr>::=REP(CHO(OBJECT_CLASS_ID(PageSet)))
")

DEFINE(PAGSETGFS,"
<constr-expr>::=SEQ(OPT(OBJECT_CLASS_ID(Page)),
REP(CHO(REP(OBJECT_CLASS_ID(Page)), SEQ(OPT(OBJECT_CLASS_ID(RPage)),
OPT(REP(SEQ(OBJECT_CLASS_ID(VPage), OBJECT_CLASS_ID(RPage)))),
OPT(OBJECT_CLASS_ID(VPAGE))))))
")

DEFINE(PAGEGFS,"
<constr-expr>::=AGG(CHO(OPT(OBJECT_CLASS_ID( BasicHeaderFooter)),
OPT(OBJECT_CLASS_ID( CompositeHeaderFooter))),
CHO(OBJECT_CLASS_ID(BasicBody), OBJECT_CLASS_ID(CompositeBodyFixed)),
CHO(OPT(OBJECT_CLASS_ID( CompositeHeaderFooter)), OPT(OBJECT_CLASS_ID(
BasicHeaderFooter))))
")

DEFINE(COMPOSITEHFGFS,"
<constr-expr>::=CHO(REP(CHO( OBJECT_CLASS_ID(SourcedContentVariable),
OBJECT_CLASS_ID(ArrangedContentVariable))), REP(CHO(
OBJECT_CLASS_ID(SourcedContentFixed),
OBJECT_CLASS_ID(ArrangedContentFixed))))
")

DEFINE(COMPOSITEBODYFIXEDGFS,"
<constr-expr>::=REP(CHO(OBJECT_CLASS_ID(ColumnVariable),
OBJECT_CLASS_ID(ColumnsSynchronized), OBJECT_CLASS_ID(ColumnsSnaking),
OBJECT_CLASS_ID(Footnote)))
")

DEFINE(ARRANGEDCONTENTFIXEDGFS,"
<constr-expr>::=REP(OBJECT_CLASS_ID(Block))
")

DEFINE(ARRANGEDCONTENTVARIABLEGFS,"
<constr-expr>::=REP(OBJECT_CLASS_ID(Block))
")

DEFINE(COLUMNSSNAKINGGFS,"
<constr-expr>::=REP(OBJECT_CLASS_ID(ColumnVariable))
")

DEFINE(COLUMNSSYNCHRONIZEDGFS,"
<constr-expr>::=SEQ(OBJECT_CLASS_ID(ColumnFixed)
")
```

17.7.3.3Factor constraints

```
FACTOR: ANY-LAYOUT(
GENERIC:
REQ     Object-type{VIRTUAL};
REQ     Object-class-identifier{ANY_VALUE};

SPECIFIC:
PERM    Object-type{VIRTUAL};
REQ     Object-identifier{ANY_VALUE};
REQ     Object-class{VIRTUAL};
REQ     Subordinates{VIRTUAL};

SPECIFIC_AND_GENERIC:
PERM    User-visible-name{ANY_VALUE};
PERM    User-readable-comment{ANY_VALUE};
}

FACTOR: ANY-PAGE:ANY-LAYOUT{

GENERIC:
REQ     Object-type{PAGE};
        CASE {$DAC OF
   $FDA:
PERM    Generator-for-subordinates{$PAGEGFS};
   $PDA:
REQ     Generator-for-subordinates{$PAGEGFS};
   $FPDA:
REQ     Generator-for-subordinates{$PAGEGFS};
PERM    Bindings{[$INITIALISEANY], $USEPGNUMBER};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Subordinates{[SUBORDINATE_ID_OF( BasicHeaderFooter |
CompositeHeaderFooter)], SUBORDINATE_ID_OF(BasicBody |
CompositeBodyFixed), [SUBORDINATE_ID_OF( CompositeHeaderFooter |
BasicHeaderFooter)] };
PERM    Object-type{PAGE};

SPECIFIC_AND_GENERIC:
PERM    Dimensions{ANY-VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Page-position{ANY_VALUE};
PERM    Bindings{$USEPGNUMBER};
}

FACTOR: ANY-FRAME:ANY-LAYOUT{

GENERIC:
REQ     Object-type{FRAME};

SPECIFIC:
PERM    Object-type{FRAME};
REQ     Subordinates{VIRTUAL};
PERM    Layout-path{VIRTUAL};
}
```

17.7.3.4Constituent constraints

17.7.3.4.1LayDoc:ANY-LAYOUT{

```
GENERIC:
REQ     Object-type{DOCUMENT_LAYOUT_ROOT};
REQ     Generator-for-subordinates{$LAYDOCGFS};
PERM    Resource{ANY_VALUE};

SPECIFIC:
        CASE {$DAC OF
   $FDA:
PERM    Object-class{OBJECT_CLASS_ID(LayDoc)};
   $PDA:
REQ     Object-class{OBJECT_CLASS_ID_OF(LayDoc)};
   $FPDA:
REQ     Object-class{OBJECT_CLASS_ID_OF(LayDoc)};
REQ     Subordinates{SUBORDINATE_ID_OF( PageSet)+};
PERM    Object-type{DOCUMENT_LAYOUT_ROOT};

SPECIFIC_AND_GENERIC:
PERM    Default-value-lists{ANY_VALUE};
PERM    Bindings{$INITIALISEANY)};
PERM    Application-comments{"LayDoc"};
}
```

17.7.3.4.2PageSet:ANY-LAYOUT{

```
GENERIC:
REQ     Object-type{PAGE_SET};
REQ     Generator-for-subordinates{$PAGESETGFS};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(PageSet)};
REQ     Subordinates{ {[SUBORDINATE_ID_OF(Page)]+,
[SUBORDINATE_ID_OF(RPage)], [SUBORDINATE_ID_OF(VPage),
SUBORDINATE_ID_OF(RPage)]+, [SUBORDINATE_ID_OF(VPage)]} };
PERM    Object-type{PAGE_SET};

SPECIFIC_AND_GENERIC:
PERM    Bindings{$INITIALISEANY};
PERM    Application-comments{"PageSet"};
}
```

17.7.3.4.3Page:ANY-PAGE{

```
SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(Page)};
```

```
SPECIFIC_AND_GENERIC:
REQ     Medium-type{NON_BASIC};
PERM    Application-comments{"Page"};
}
```

17.7.3.4.4RPage:ANY-PAGE{

```
SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(RPage)};

SPECIFIC_AND_GENERIC:
REQ     Medium-type{NON_BASIC};
PERM    Application-comments{"RPage"};
}
```

17.7.3.4.5VPage:ANY-PAGE{

```
SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF(VPage)};

SPECIFIC_AND_GENERIC:
REQ     Medium-type{NON_BASIC};
PERM    Application-comments{"VPage"};
}
```

17.7.3.4.6CompositeHeaderFooter:ANY-FRAME{

```
GENERIC:
REQ     Generator-for-subordinates{$COMPOSITEHFGFS};
REQ     Position{#fixed{ANY_VALUE}};
REQ     Dimensions{#horizontal{ #fixed{ANY_VALUE}},
        #vertical{#fixed{ANY_VALUE}}};
REQ     Application-comments{"CompositeHeaderFooter"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( CompositeHeaderFooter)};
REQ     Subordinates{{SUBORDINATE_ID_OF( SourcedContentVariable) |
SUBORDINATE_ID_OF( ArrangedContentVariable)}+ | {SUBORDINATE_ID_OF(
SourcedContentFixed) |  SUBORDINATE_ID_OF( ArrangedContentFixed)}+ } };
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"CompositeHeaderFooter"};

SPECIFIC_AND_GENERIC:
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
PERM    Layout-path{ANY_VALUE};
}
```

17.7.3.4.7CompositeBodyFixed:ANY-FRAME{

```
GENERIC:
REQ     Generator-for-subordinates{$COMPOSITEBODYFIXEDGFS};
REQ     Position{#fixed{ANY_VALUE}};
REQ     Dimensions{#horizontal{ #fixed{ANY_VALUE}},
        #vertical{#fixed{ANY_VALUE}}};
REQ     Application-comments{"CompositeBodyFixed"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF (CompositeBodyFixed)};
REQ     Subordinates{{SUBORDINATE_ID_OF( ColumnVariable) |
SUBORDINATE_ID_OF( SynchronizedColumns) | SUBORDINATE_ID_OF(
SnakingColumns) | SUBORDINATE_ID_OF(FootNote)}+ };
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
        #vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"CompositeBodyFixed"};

SPECIFIC_AND_GENERIC:
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};

}
```

17.7.3.4.8**ColumnFixed**:ANY-FRAME{

```
GENERIC:
REQ     Position{#fixed{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #maximum-size},
#vertical{#rule-b{ANY_VALUE} | #maximum-size}};
REQ     Application-comments{"ColumnFixed"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( ColumnFixed)};
REQ     Subordinates{{OBJECT_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"ColumnFixed"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
}
```

17.7.3.4.9**ColumnVariable**:ANY-FRAME{

```
GENERIC:
REQ     Position{#variable{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #maximum-size},
#vertical{#rule-b{ANY_VALUE} | #maximum-size}};
REQ     Application-comments{"ColumnVariable"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( ColumnVariable)};
REQ     Subordinates{{OBJECT_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"ColumnVariable"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
}

17.7.3.4.10SnakingColumns:ANY-FRAME{

GENERIC:
REQ     Generator-for-subordinates{$SNAKINGCOLUMNSGFS};
REQ     Position{#variable{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #maximum-size},
#vertical{#rule-b{ANY_VALUE}}};
REQ     Application-comments{"SnakingColumns"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( SnakingColumns)};
REQ     Subordinates{{SUBORDINATE_ID_OF( ColumnVariable)}+ };
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"SnakingColumns"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
PERM    Layout-path{'0-degrees' | '90-degrees' | '270-degrees'};
}

17.7.3.4.11SynchronisedColumns:ANY-FRAME{

GENERIC:
```

```
REQ     Generator-for-subordinates{$SYNCHRONISEDCOLUMNSGFS};
REQ     Position{#variable{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #maximum-size},
#vertical{#rule-b{ANY_VALUE}}};
REQ     Application-comments{"SynchronisedColumns"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( SynchronisedColumns)};
REQ     Subordinates{{SUBORDINATE_ID_OF( ColumnFixed)}+ };
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"SynchronisedColumns"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
   --  Subordinates of SynchronisedColumns must all have different values
   for Permitted-categories  --
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
PERM    Layout-path{'0-degrees' | '90-degrees' | '180-degrees' |
'270-degrees'};
PERM    Balance{ANY_VALUE};
}

17.7.3.4.12FootNote:ANY-FRAME{

GENERIC:
REQ     Position{#variable{ #offset{ANY_VALUE},  #separation{ANY_VALUE},
#alignment{ANY_VALUE}, #fillorder{'reversed'}}};
REQ     Dimensions{#horizontal{#maximum-size}, #vertical{-
#rule-b{ANY_VALUE}}};
REQ     Application-comments{"FootNote"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( FootNote)};
REQ     Subordinates{{SUBORDINATE_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{#maximum-size},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"FootNote"};

SPECIFIC_AND_GENERIC:
REQ     Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
}
```

```
17.7.3.4.13ArrangedContentFixed:ANY-FRAME{

GENERIC:
REQ     Position{#fixed{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #rule-b{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE} | #rule-b{ANY_VALUE}}};
REQ     Application-comments{"ArrangedContentFixed"};
PERM    Generator-for-subordinates{$ARRANGEDCONTENTFIXED};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( ArrangedContentFixed)};
REQ     Subordinates{SUBORDINATE_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"ArrangedContentFixed"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
}

17.7.3.4.14ArrangedContentVariable:ANY-FRAME{

GENERIC:
REQ     Position{#variable{ANY_VALUE}};
REQ     Dimensions{#horizontal{#fixed{ANY_VALUE} | #rule-b{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE} | #rule-b{ANY_VALUE}}};
REQ     Application-comments{"ArrangedContentVariable"};
PERM    Generator-for-subordinates{$ARRANGEDCONTENTVARIABLE};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( ArrangedContentVariable)};
REQ     Subordinates{SUBORDINATE_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Imaging-order{ANY_VALUE};
PERM    Application-comments{"ArrangedContentVariable"};

SPECIFIC_AND_GENERIC:
PERM    Permitted-categories{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
}
```

17.7.3.4.15SourcedContentFixed:ANY-FRAME{

```
GENERIC:
REQ     Position{#fixed{ANY_VALUE}};
REQ     Dimensions{#horizontal{ #fixed{ANY_VALUE}}, #vertical{
#rule-b{ANY_VALUE}}};
REQ     Logical-source{OBJECT_CLASS_ID_OF( CommonContent)};
REQ     Application-comments{"SourcedContentFixed"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( SourcedContentFixed)};
REQ     Subordinates{{SUBORDINATE_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Application-comments{"SourcedContentFixed"};

SPECIFIC_AND_GENERIC:
PERM    Border{ANY_VALUE};
PERM    Layout-path{ANY_VALUE};
}
```

17.7.3.4.16SourcedContentVariable:ANY-FRAME{

```
GENERIC:
REQ     Position{#variable{ANY_VALUE}};
REQ     Dimensions{#horizontal{ #fixed{ANY_VALUE}}, #vertical{
#rule-b{ANY_VALUE}}};
REQ     Logical-source{OBJECT_CLASS_ID_OF( CommonContent)};
REQ     Application-comments{"SourcedContentVariable"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( SourcedContentVariable)};
REQ     Subordinates{{SUBORDINATE_ID_OF(Block)}+};
PERM    Position{ANY_VALUE};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Application-comments{"SourcedContentVariable"};

SPECIFIC_AND_GENERIC:
PERM    Border{ANY_VALUE};
PERM    Layout-path{ANY_VALUE};
}
```

17.7.3.4.17BasicHeaderFooter:ANY-FRAME{

```
GENERIC:
REQ     Logical-source{ANY_VALUE};
REQ     Application-comments{"BasicHeaderFooter"};
```

```
SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( BasicHeaderFooter)};
REQ     Subordinates{{SUBORDINATE_ID_OF(Block)}+};
PERM    Application-comments{"BasicHeaderFooter"};

SPECIFIC_AND_GENERIC:
PERM    Position{#fixed{ANY_VALUE}};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Layout-path{'270-degrees'};
}
```

17.7.3.4.18BasicBody:ANY-FRAME{

```
GENERIC:
REQ     Application-comments{"BasicBody"};

SPECIFIC:
REQ     Object-class{OBJECT_CLASS_ID_OF( BasicBody)};
REQ     Subordinates{{SUBORDINATE_ID_OF(Block)}+};
PERM    Application-comments{"BasicBody"};

SPECIFIC_AND_GENERIC:
PERM    Position{#fixed{ANY_VALUE}};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Layout-path{'270-degrees'};
}
```

17.7.3.4.19Block:ANY-LAYOUT{

```
GENERIC:
REQ     Object-type{BLOCK};
REQ     Content-architecture-class{$FC | $PC | $FPC | $FPR | $FPG};
PERM    Content-portions{ANY_VALUE};
REQ     Application-comments{"Block"};
PERM    Resource{ANY_VALUE};

SPECIFIC:
REQ     Content-architecture-class{ANY_VALUE};
PERM    Position{#fixed{ANY_VALUE}};
PERM    Dimensions{#horizontal{ #fixed{ANY_VALUE}},
#vertical{#fixed{ANY_VALUE}}};
PERM    Initial-offset{ANY_VALUE};
PERM    Formatting-indicator{ANY_VALUE};
PERM    Graphic-rendition{ANY_EXCEPT 26,
        --  Variable spacing  --
        50};
        --  Not variable spacing  --
PERM    Graphic-character-set{ANY_VALUE};
PERM    Application-comments{"Block"};
```

```
SPECIFIC_AND_GENERIC:
PERM    Transparency{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Border{ANY_VALUE};
PERM    Presentation-style{STYLE_ID_OF(PStyle1 | PStyle2 | PStyle3 |
PStyle4};
}
```

17.7.4 Layout style constraints

Note: This section has not been aligned with the logical and layout constraint objects defined in sections 17.7.2 and 17.7.3.

17.7.4.1Factors constraints

```
FACTOR ANY-LAYOUT-STYLE {

REQ     Layout-style-identifier{ANY_VALUE};
PERM    User-visible-name{ANY_VALUE};
PERM    User-readable-comments{ANY_VALUE};
}
```

17.7.4.2Constituent constraints

```
17.7.4.2.1LStyle1:ANY-LAYOUT-STYLE{
   -- Used for LogDoc only --
REQ     Layout-object-class{OBJECT_CLASS_ID_OF(Laydoc)};
}
17.7.4.2.2LStyle2:ANY-LAYOUT-STYLE{
                     -- Used for PageNumber only --
PERM    Block-alignment{ANY_VALUE};
PERM    Concatenation{ANY_VALUE};
PERM    Indivisibility{ANY_VALUE};
PERM    Layout-category{ANY_VALUE};
PERM    Layout-object-class{ANY_VALUE};
PERM    New-layout-object{ANY_VALUE};
PERM    Same-layout-object{ANY_VALUE};
PERM    Offset{ANY_VALUE};
PERM    Separation{ANY_VALUE};
}

17.7.4.2.3LStyle3:ANY-LAYOUT-STYLE{
           -- Used for Passage, Paragraph, Numbered Segment, --
                       -- FNote and FNBody only --
PERM    Indivisibility{ANY_VALUE};
PERM    Layout-object-class{ANY_VALUE};
PERM    New-layout-object{ANY_VALUE};
PERM    Same-layout-object{ANY_VALUE};
PERM    Synchronization{ANY_VALUE};
}

17.7.4.2.4LStyle4:ANY-LAYOUT-STYLE{
```

```
                          -- Used for Number only --
PERM    Block-alignment{ANY_VALUE};
PERM    Concatenation{ANY_VALUE};
PERM    Indivisibility{ANY_VALUE};
PERM    Layout-category{ANY_VALUE};
PERM    Layout-object-class{ANY_VALUE};
PERM    New-layout-object{ANY_VALUE};
PERM    Same-layout-object{ANY_VALUE};
PERM    Offset{ANY_VALUE};
PERM    Separation{ANY_VALUE};
PERM    Synchronisation{ANY_VALUE};
}
```

```
17.7.4.2.5LStyle5:ANY-LAYOUT-STYLE{
                         -- Used for BodyText only --
PERM    Block-alignment{ANY_VALUE};
PERM    Concatenation{ANY_VALUE};
PERM    Indivisibility{ANY_VALUE};
PERM    Layout-category{ANY_VALUE};
PERM    Layout-object-class{ANY_VALUE};
PERM    New-layout-object{ANY_VALUE};
PERM    Same-layout-object{ANY_VALUE};
PERM    Offset{ANY_VALUE};
PERM    Separation{ANY_VALUE};
PERM    Synchronisation{ANY_VALUE};
PERM    Fill-order{ANY_VALUE};
}
```

```
17.7.4.2.6LStyle6:ANY-LAYOUT-STYLE{
           --  Used for BodyRaster and BodyGeometric only --
PERM    Block-alignment{ANY_VALUE};
PERM    Indivisibility{ANY_VALUE};
PERM    Layout-category{ANY_VALUE};
PERM    Layout-object-class{ANY_VALUE};
PERM    New-layout-object{ANY_VALUE};
PERM    Same-layout-object{ANY_VALUE};
PERM    Offset{ANY_VALUE};
PERM    Separation{ANY_VALUE};
PERM    Synchronisation{ANY_VALUE};
}
```

17.7.5 Presentation style constraints

Note: This section has not been aligned with the logical and layout constraint objects defined in sections 17.7.2 and 17.7.3.

17.7.5.1Macro definitions

```
DEFINE(C-PRES-ATTR,"
PERM    Alignment{ANY_VALUE};
PERM    Character-fonts{ANY_VALUE};
PERM    Character-orientation{$BASIC-CHAR-ORIENTATION |
        $NON-BASIC-CHAR-ORIENTATION};
PERM    Character-path{$BASIC-CHAR-PATH | $NON-BASIC-CHAR-PATH};
PERM    Character-spacing{ANY_VALUE};
PERM    Code-extension-announcers{$CODE-EXT-ANNOUNCERS};
PERM    First-line-offset{ANY_VALUE};
PERM    Formatting-indicator{ANY_VALUE};
PERM    Graphic-character-sets{$BASIC-CHAR-SET | $NON-BASIC-CHAR-SET};
PERM    Character-subrepertoire{$BASIC-SUBREPERTOIRES |
        $NON-BASIC-SUBREPERTOIRES};
PERM    Graphics-rendition{ANY_EXCEPT 'variable-spacing',
        'not-variable-spacing'};
PERM    Indentation{ANY_VALUE};
PERM    Initial-offset{ANY_VALUE};
PERM    Itemization{ANY_VALUE};
PERM    Kerning-offset{ANY_VALUE};
PERM    Line-layout-table{ANY_VALUE};
PERM    Line-progression{ANY_VALUE};
PERM    Line-spacing{ANY_VALUE};
PERM    Orphan-size{ANY_VALUE};
PERM    Pairwise-kerning{ANY_VALUE};
PERM    Proportional-line-spacing{ANY_VALUE};
PERM    Widow-size{ANY_VALUE}; ")

DEFINE(R-PRES-ATTR,"
PERM    Pel-path{'0-degrees' | '270-degrees'};
PERM    Line-progression{'0-degrees' | '270-degrees'};
PERM    Pel-spacing{ANY_INTEGER <=1200};
DIS     Spacing-ratio{ANY_VALUE};
PERM    Clipping{ANY_VALUE};
PERM    Image-dimensions{ANY_VALUE}; ")

DEFINE(G-PRES-ATTR,"
PERM    Geometric-graphics-encoding-announcer
{  #VDC-type{ANY_VALUE},
   #Integer-precision{16 | 32},
   #Real-precision{{0 9 23} | {1 16 16}},
   #Index-precision{8 | 16},
   #Colour-precision{8 | 16},
   #Colour-index-precision{8 | 16},
   #Maximum-colour-index{ANY_VALUE},
   #Colour-value-extent{ANY_VALUE},
   #Colour-selection-mode{ANY_VALUE};
   #VDC-integer-precision{16 | 32},
   #VDC-real-precision{{0 9 23} | {1 16 16}} };
PERM    Line-rendition{ANY_VALUE};
PERM    Marker-rendition{ANY_VALUE};
PERM    Text-rendition
{
   #Font-list{ANY_VALUE},
   #Character-set-list{$BASIC-CHAR-SET | $NON-BASIC-CHAR-SET},
   #Character-coding-announcer{basic-7-bit | basic-8-bit},
   #Text-bundle-index{ANY_VALUE},
   #Text-font-index{ANY_VALUE},
   #Text-precision{ANY_VALUE},
   #Character-expansion-factor{ANY_VALUE},
   #Character-spacing{ANY_VALUE},
   #Text-colour{ANY_VALUE},
   #Character-height{ANY_VALUE},
   #Character-orientation{ANY_VALUE},
   #Text-path{ANY_VALUE},
   #Text-alignment{ANY_VALUE},
   #Character-set-index{ANY_VALUE},
   #Text-asf{ANY_VALUE}
   #Text-bundle-representation{ANY_VALUE} };
PERM    Filled-area-rendition
{
   #Fill-bundle-index{ANY_VALUE},
   #Interior-style{ANY_VALUE},
   #Fill-colour{ANY_VALUE},
   #Hatch-index{ANY_VALUE},
   #Pattern-index{1 .. 8},
   #Fill-reference-point{ANY_VALUE},
   #Pattern-size{ANY_VALUE},
   #Pattern-table-representation
   {    #Pattern-table-index{1 .. 8},
        #Number-of-columns{1 .. 16},
        #Number-of-rows{1 .. 16},
        #Local-colour-precision{0 | 1 | 8 | 16},
        #Colour-arry{ANY_VALUE} },
   #Fill-asf{ANY_VALUE} };
PERM    Edge-rendition{ANY_VALUE},
PERM    Colour-representation{ANY_VALUE};
PERM    Transparency-specification{ANY_VALUE};
PERM    Transformation-specification{ANY_VALUE};
PERM    Region-of-interest-specification {ANY_VALUE};
PERM    Picture-orientation{ANY_VALUE};
PERM    Picture-dimensions{ANY_VALUE}; ")
```

17.7.5.2Factor constraints

```
FACTOR: ANY-PRESENTATION-STYLE {
REQ     Presentation-style-identifier{ANY_VALUE};
PERM    User-readable-comments{ANY_VALUE};
PERM    User-visible-name{ANY_VALUE};
PERM    Border{ANY_VALUE};
PERM    Colour{ANY_VALUE};
PERM    Transparency{ANY_VALUE};
}
```

17.7.5.3Constituent constraints

17.7.5.3.1PStyle1:ANY-PRESENTATION-STYLE{

```
PERM    Presentation-Attributes{$C-PRES-ATTR};
```

```
}
```

17.7.5.3.2PStyle2:ANY-PRESENTATION-STYLE{

```
CASE    (Document-profile(Document-characteristics
    #Content-archicture-class)) OF
$FDA:
REQ     Content-architecture-class{$CF};
$PDA:
REQ     Content-architecture-class{$CP};
$FPDA:
REQ     Content-architecture-class{$CFP};
-- ENDCASE --
PERM    Presentation-attributes{$C-PRES-ATTR};
}
```

17.7.5.3.3PStyle3:ANY-PRESENTATION-STYLE{

```
REQ     Content-architecture-class{$RFP};
PERM    Presentation-attributes{$R-PRES-ATTR};
}
```

17.7.5.3.4PStyle4:ANY-PRESENTATION-STYLE{

```
REQ     Content-architecture-class{$GFP};
PERM    Presentation-attributes{$G-PRES-ATTR};
}
```

17.7.6 Content portion constraints

17.7.6.1Character content portion

```
SPECIFIC_AND_GENERIC:
PERM Content-identifier-layout{ANY_VALUE};
PERM Content-identifier-logical{ANY_VALUE};
REQ  Type-of-coding{2 8 3 6 0};
PERM Alternative-representation{ANY_VALUE};
PERM Content-information{
     {    #Character{ANY_VALUE},
-- Shared Control Functions --
          #CR{},
          #GCC{ANY_VALUE},
          #IGS{$BASIC-SUBREPERTOIRE | $NON-BASIC-SUBREPERTOIRE},
     --  Note:The use of IGS is suppose to be deprecated.  --
          #LF{},
          #PLD{},
          #PLU{},
          #SCS{ANY_VALUE},
          #SGR{ANY_VALUE},
          #SHS{0 | 1 | 2 | 3},
          #SLS{ANY_VALUE},
          #SRS{ANY_VALUE},
          #STAB{ANY_VALUE},
          #SUB{},
          #SVS{ANY_VALUE},
          #VPB{ANY_VALUE},
          #VPR{ANY_VALUE},
-- Layout Control Functions --
          #HPB{ANY_VALUE},
          #HPR{ANY_VALUE},
          #JFY{ANY_VALUE},
          #SACS{ANY_VALUE},
          #SRCS{ANY_VALUE},
          #SSW{ANY_VALUE},
-- Logical Control Functions --
          #BPH{},
          #NBH{},
          #PTX{ANY_VALUE},
-- Delimiter Functions --
          #SOS{},
          #SP{},
          #ST{} };
```

17.7.6.2 Raster graphics content portion

```
DEFINE(T6,"{2 8 3 7 0}")
DEFINE(T41D,"{2 8 3 7 1}")
DEFINE(T42D,"{2 8 3 7 2}")
DEFINE(BITMAP,"{2 8 3 7 3}")

PERM Content-identifier-logical{ANY_VALUE};
PERM Content-identifier-layout{ANY_VALUE};
REQ  Type-of-coding{$T6 | $T41D | $T42D | $BITMAP};
PERM Alternative-representation{ANY_VALUE};
PERM Coding-attributes{
     {    #Compression{ANY_VALUE},
          #Number-of-lines{ANY_VALUE},
          #Number-of-pels-per-line{ANY_VALUE},
          };
PERM Content-information{ANY_VALUE};
```

17.7.6.3 Geometric graphics content portion

```
PERM Content-identifier-logical{ANY_VALUE};
PERM Content-identifier-layout{ANY_VALUE};
REQ  Alternative-representation{ANY_VALUE};
PERM Content-information{ANY_VALUE};

-- Section 17.10.1 contains a recommended functional subset of
the CGM standard for this document application profile --
```

17.7.7 Additional usage constraints

No other usage constaints are currently defined.

17.8 Interchange format

Interchange format class "A" is to be used in this application profile, as defined in ISO 8613-5.

The encoding is in accordance with the Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), as defined in ISO 8825.

17.8.1 ASN.1 generation constraints

The following are additional constraints imposed on the ASN.1 generation beyond those defined in ISO 8824 and ISO 8825.

17.8.2 Encoding of application comments

ISO 8613-5 define the encoding of the attribute Application Comments as an octet string. This document application profile requires that the encoding within that octet string be in accordance with the ASN.1 syntax specified in the following module definition.

```
NISTDAPSpecification
DEFINITION::=BEGIN
EXPORTS Application-Comments-Encoding;

Application-Comments-Encoding::=SEQUENCE {
     Constraint-name[0]IMPLICIT PrintableString
     External-data[1]EXTERNAL}
END
```

17.8.3 Encoding of raster content information

The encoding of raster content information in the bitmap encoding scheme is that specified in clause 9.3 of the raster graphics content architecture part of the base standard. The encoding of the code words in the Group 4 facsimile encoding scheme is such that the first or only bit of the first code word shall be placed in the most significant bit of the first octet. Subsequent bits of the first and following code words are placed in the direction of less significant bits in the first and following octets.

17.9 GSS/RSS proforma

17.9.1 Generator support statement proforma

Note:This section is being written in conjunction with the ODA document application profile International alignment activity in PAGODA.

17.9.2 Receiver support statement proforma

Note:This section is being written in conjunction with the ODA document application profile International alignment activity in PAGODA.

17.10 Informative Recommendations

17.10.1 ISO 8632 (CGM) constraints for this DAP

It is recommended that geometric graphics content information contain only those elements listed in this portion of the agreements, in addition to the constraints imposed by ISO 8613-8. It is believed that this subset of the CGM is sufficiently widely implemented to enable interworking of geometric graphics for application conforming this document application profile.

The content information of a content portion description that conforms to this content architecture is an ASN.1 octet string representing a Computer Graphics Metafile (CGM) conforming to the following constraints:

a) Conform to part 1 of the ISO 8632 standard;
b) Conform to the binary encoding defined in part 3 of the ISO 8632 standard;
c) Consist of a single picture;
d) Conform to the ISO pdISP FCG13, except as noted with respect to font and colour table support;
e) Generalized Drawing Primitives are ignored;
f) ESCAPE Elements are ignored;
g) External Elements may be ignored.

The following list is a description of the constraints for each of the CGM elements. Where an element has parameters, recommended constraints on the values are given. The "--" symbol indicates that there is no recommended constraint.

Requirements in ISO 8632 and ISO 8613-8 concerning mandatory elements, parameters must be fulfilled.

17.10.1.1 Delimiter elements

Begin MetafileSee Note 1
End Metafile--
Begin PictureSee Note 1
Begin Picture Body--
End Picture--

17.10.1.2 Metafile description elements

Metafile Version1
Metafile DescriptionSee Notes 1, 2
Real Precision(0,9,23), (1,16,16)
Index Precision16

Colour Precision8, 16
Colour Index Precision8, 16
Maximum Colour Index0-255
Colour Value Extent3-tuple in range (0,32767)
Metafile Element List-1,1
Metafile Defaults ReplacementSee Note 3
Font List See Note 4
Character Set ListSee Note 5
Character Coding Announcer(Editor) Above FCG12
basic 7-bit, basic-8-bit

17.10.1.3 Picture descriptor elements

VDC Extent--
Background Colour--

17.10.1.4 Control elements

Transparency--
Clip Rectangle--
Clip Indicator--

17.10.1.5 Graphical primitive elements

Polyline See Note 7
PolymarkerSee Note 7
Text   See Note 2
Polygon   See Note 7
Polygon SetSet Note 7
Rectangle --
Circle   --
Circular Arc Centre--
Circular Arc Centre Close--
Ellipse   --
Elliptical Arc--
Elliptical Arc Close--

17.10.1.6 Attribute elements

Line Type 1-5
Line Width--
Line Colour--
Marker Type1-5
Marker Size--
Marker Colour--
Text Font Index--
Text Precision--(Editor) Above FCG12
Character Expansion Factor--(Editor) Above FCG12
Character Spacing--(Editor) Above FCG12
Text Colour--
Character Height--
Character Orientation--

Text Path --(Editor) Above FCG12
Text Alignmenthorizontal: normal, left, centre, right
vertical: normal, top, cap, half, base, bottom
Character Set Index1, 2(Editor) Above FCG12
Interior Style0, 1, 3, 4
Fill Colour--
Hatch Index1-6
Colour Table SpecificationSee Notes 8, 9

17.10.1.7 External Elements

Message   No action
Application DataSee Note 1

Note 1: Support will be provided for strings with a length up to 256 octets, except for data records which will support strings with a length up to 32767 octets.

Note 2: The METAFILE DESCRIPTION string parameter will be used to include the sub-string "ISO FCG12" to label the content information as conforming to this agreement. In addition, generator of content are encouraged to append a sub-string that identifies the company and product that produced the CGM.

Note 3: The METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. No part of the element will be partitioned. Multiple occurrences of the MDR element may be used to avoid the need for partitioning. The MDR element must appear in the CGM to establish the defaults for TEXT PRECISION and any other elements whose defaults are different than those specified in ISO 8632-1 and -3.

Note 4: The only fonts that may be specified are those specified in the document profile. The font list must be in the same order as that specified in the document profile.

Note 5: The only character sets that may be specified are ISO 6937/2 (0, 4/0) and ISO 8859/1 (0, 4/2). The order of the specification of these characters must match the order specified in the document profile.

Note 6: The Scale Factor parameter of SCALING MODE element is always a 32-bit floating point value, even when the REAL PRECISION has selected fixed point for other real numbers. It is not apparent in ISO 8632 what the precision of this floating point value is when fixed point has been selected. Its precision shall be (0,9,23).

Note 7: The minimum support for the length of point lists is 1024 elements.

Note 8: The COLOUR TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. The COLOUR TABLE element shall appear prior to any graphical primitive elements to assure that interpreting systems without dynamic colour update can render the intended effect.

Note 9: The minimum support for the length of the Colour List parameter in the COLOUR TABLE element is 61. This will support a 63 entry colour table.

17.10.2 Registration of entities

The NIST OSI Implementor's Workshop as allocated a name space to the NIST ODA SIG. The name space is intended to be used by the NIST ODA SIG for registration of entities within its domain. For example, object identifiers for ODA document application profiles. Other possibilities include private content or font identifiers. The name space is identified by the following path specification:

```
{ISO(1)IdentifiedOrganization(3)OIW(14)ODAISG(11)}
```

To facilitate the management of this name space, factorization of identifiers is required. The ODASIG level of the path shall contain a subordinate level for ODA document application profiles, with identifier value 0. Other subordinates at this level may be defined in the future. Subordinate to the document application profiles level is a level for each functional level of document application profile. At present, two levels have been identified for the Level 3 DAP (identifier value of 0) and the Level 2 DAP (identifier value of 1). Subordinate to this level is an identifier for each instance of a document application profile defined at this level of profile.

The NIST Level 2 DAP defined in by this implementation agreements document is defined by the object identifier:

```
{ISO(1)IdentifiedOrganization(3)OIW(14)ODASIG(11)
DAPs(0)Level2(1)Alpha(0)}
```

In order to reduce the verbosity of this representation, it is recommended that the following notation be used, in place of the complete description:

```
{1 3 14 11 0 1 0}
```

17.10.3 Conveyance of ODA over CCITT X.400-1984

This recommendation describes how ODA body parts are to be encoded for transmission over a CCITT X.400-1984 service.

17.10.3.1 P2 protocol encoding

An ODA document will be transferred as a single body part with tag 12:

```
oda [12] IMPLICIT OCTET STRING
```

The content of the OCTET STRING will contain an ASN.1/BER encoded segment with a value of type OdaBodyPart, which is a SEQUENCE containing the OdaBodyPartParameters and OdaData components:

```
OdaBodyPart ::= SEQUENCE{
     OdaBodyPartParameters,
     OdaData }
```

The OdaBodyPartParameters and the OdaData components are each aligned to Annex-E of ISO 8613-1 and CCITT T.411-1988.

The OdaBodyPartParameters component is a SET containing the document-application-profile and the document-architecture-class identifiers:

```
OdaBodyPartParameters ::= SET {
     document-application-profile
     [0] IMPLICIT OBJECT IDENTIFER,
     document-architecture-class
     [1] IMPLICIT INTEGER {
     formatted (0),
     processable (1),
     formatted-processable (2) }}
```

The OdaData component is a SEQUENCE OF Interchange-Data-Element as defined by ISO 8613-5:

```
OdaData ::= SEQUENCE OF Interchange-Data-Element
```

17.10.3.2 P1 protocol encoding

The Encoded Information Type (EIT) for an ODA body part will be the 'ODA' bit, bit 10. The 'Undefined' bit, bit 0, must be set as well. An MTA can test for deliverability on the basis of the presence of an ODA body part.
The EITs are unable to record the document-application-profile. Also, the G3Fax EIT bit must not be set even if the ODA document contains G3Fax content portions.

17.10.4 Interoperability with SGML applications

The recommended method for the exchange of documents between Standard Generalized Markup Language (ISO 8879, SGML) based systems and systems based on this ODA document application profile is by means of exchanging a document representation conforming to these agreements in an encoded form of the SGML language known as the Office Document Language (ODL). ODL is a standardized SGML application for representing documents conforming to the ODA base standard. Such a representation can be converted into the Office Document Interchange Format (ODIF) supported by this document application profile.

18. NETWORK MANAGEMENT

Editor's Note: There is currently no text for subsections 8, 9, and 10 (described below).

Editor's Note: The notes in this section are meant to be placeholders for future text. They are included here to reflect SIG activity in these areas.

18.1 INTRODUCTION

Within the community of OSI researchers, users, and vendors, there is a recognized need to address the problems of initiating, terminating, monitoring, and controlling communication activities and assisting in their harmonious operation, as well as handling abnormal conditions. The activities that address these problems are collectively called network management.

Network management can then be viewed as the set of operational and administrative mechanisms necessary to:

a. bring up, enroll, and/or alter network resources,

b. keep network resources operational,

c. fine tune these resources and/or plan for their expansion,

d. manage the accounting of their usage, and

e. manage their protection from unauthorized use/tampering.

As such, network management is typically concerned with management activities in at least the following five functional areas: configuration management, fault management, performance management, accounting management, and security management. In order to accomplish these management activities, information must be exchanged among management processes. Managing processes have the responsibility for carrying out one or more management activities. Agent processes act on behalf of managing processes, forwarding notifications from and manipulating managed objects.

In this section, there are Implementation Agreements (IA's) for providing interoperable OSI management information communication services among OSI systems. Also contained here are agreements on management information, or pointers to other sections of this document or other documents where such additional agreements appear.

These agreements pertain to the exchange of management information and management commands between open systems operating in a multivendor environment. Therefore, the goal is to ensure that a management system built by one vendor can manage network objects built by another vendor.

In progressing work on OSI management in the NIST/OSI NMSIG, the OSI management framework specified in ISO 7498/Part 4 (as presented in reference [FRMWK]) shall be used as the basis for concepts and terminology relevant (a) to OSI management activities, and (b) to management services supported by OSI management protocols. Thus, these agreements are based on, and employ, protocols developed in accord with the OSI Reference Model. Furthermore, they attempt to eliminate ambiguities in interpretations of management protocol standards and management information standards.

18.1.1 References

The following documents are referenced in the statements of the agreements relating to NIST/OSI network management.

OSI Systems Management References:

[ADDRMVP] ISO/IEC 9596/PDAD 2, Common Management Information Protocol: Add/Remove Protocol, ISO/IEC JTC1/SC21 N3306, January 1989.

[ADDRMVS] ISO/IEC 9595/PDAD 2, Common Management Information Service: Add/Remove Service, ISO/IEC JTC1/SC21 N3305, January 1989.

[ALS] ISO/IEC DIS 9545 (Ballot), Information Processing Systems - Open Systems Interconnection - Application Layer Structure, 15 September 1988.

[AMWD] Information Processing Systems - Open Systems Interconnection - Accounting Management Working Document, ISO/IEC JTC1/SC21 N3314, December 1988.

[CANGETP] ISO/IEC 9596/PDAD 1, Common Management Information Protocol: CancelGet Protocol, ISO/IEC JTC1/SC21 N3304, January 1989.

[CANGETS] ISO/IEC 9595/PDAD 1, Common Management Information Service: CancelGet Service, ISO/IEC JTC1/SC21 N3303, January 1989.

[CMIP] ISO/IEC DIS 9596-2, Information Processing Systems - Open Systems Interconnection - Management Information Protocol Specification - Part 2: Common Management Information Protocol, 22 December 1988.

[CMIS] ISO/IEC DIS 9595-2, Information Processing Systems - Open Systems Interconnection - Management Information

Service Definition - Part 2: Common Management Information Service, 22 December 1988.

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

[DMA] ISO/IEC DP 10165-3, Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 3: Definitions of Management Attributes, ISO/IEC JTC1/SC21 N3302, January 1989.

[DSO] ISO/IEC DP 10165-2, Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 2: Definitions of Support Objects, ISO/IEC JTC1/SC21 N3301, January 1989.

[ERIRF] ISO/IEC DP 10164-4, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 4: Error Reporting and Information Retrieval Function, ISO/IEC JTC1/SC21 N3298, 31 January 1989.

[FMWD] Information Processing Systems - Open Systems Interconnection - Systems Management - Fault Management Working Document, ISO/IEC JTC1/SC21 N3312, January 1989.

[FRMWK] ISO 7498-4 (DIS), Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: OSI Management Framework - Revision of DIS 7498-4 following Editing Meeting (Sydney), 4 January 1989.

[GDMO] ISO/IEC DP 10165-4, Information Processing Systems - Open Systems Interconnection - SMI - Part 4: Guidelines for the Definition of Managed Objects, ISO/IEC JTC1/SC21 N3509, May 1989.

[LCF] First Working Draft For Systems Management: Log Control Function, ISO/IEC JTC1/SC21 N3309, January 1989.

[MIM] ISO/IEC DP 10165-1, Working Draft for Structure of Management Information - Part 1: Management Information Model, ISO/IEC JTC1/SC21 Nxxxx, May 1989.

[MSC] Proposed DP 10164-5, Information Processing Systems - Open Systems Interconnection - Systems Management - Management Service Control, ISO/IEC JTC1/SC21 N3299, January 1989.

[OMF] ISO/IEC DP 10164-1, Information Processing Systems - Open Systems Interconnection - Systems Management -

Part 1: Object Management Function, ISO/IEC JTC1/SC21 N3295, 31 January 1989.

[OSIMIL] Management Information Library (MIL) - Revision 1.0, OSI MIB Working Group of NMSIG of NIST/OSI Implementors Workshop, March 1989.

[PMWD] Information Processing Systems - Open Systems Interconnection - Performance Management Working Document (Third Draft), ISO/IEC JTC1/SC21 N3313, 18 January 1989.

[RMF] ISO/IEC DP 10164-3, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 3: Relationship Management Function, ISO/IEC JTC1/SC21 N3297, 31 January 1989.

[SMF] ISO/IEC DP 10164-2, Information Processing Systems - Open Systems Interconnection - Systems Management - Part 2: State Management Function, ISO/IEC JTC1/SC21 N3296, 31 January 1989.

[SMO] ISO/DP 10040, Information Processing Systems - Open Systems Interconnection - Systems Management Overview, ISO/IEC JTC1/SC21 N3294, January 1989.

[SMWD] Information Processing Systems - Open Systems Interconnection - Systems Management - Fifth Draft of OSI Security Management Working Document, ISO/IEC JTC1/SC21 N3315, January 1989.

Other OSI References:

[ACSEP] ISO 8650, Information Processing Systems - Open Systems Interconnection - Protocol Specification for the Association Control Service Element (Revised Final Text of DIS 8650), ISO/IEC JTC1/SC21 N2327, 21 April 1988.

[ACSES] ISO 8649, Information Processing Systems - Open Systems Interconnection - Service Definition for the Association Control Service Element (Revised Final Text of DIS 8649), ISO/IEC JTC1/SC21 N2326, 21 April 1988.

[ASN1] ISO 8824, Information Processing Systems - Open System Interconnection - Specification of Abstract Syntax Notation One (ASN.1), 19 May 1987.

[BER] ISO 8825, Information Processing Systems - Open Systems Interconnection - Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), 19 May 1987.

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

[PSD] ISO 8822, Information Processing Systems - Open Systems Interconnection - The Presentation Service Definition, October 1987.

[ROSEP] ISO 9072-2 - Information Processing Systems - Text Communications - Remote Operations Part 2: Protocol Specification, February 1988.

[ROSES] ISO 9072-1, Information Processing Systems - Text Communications - Remote Operations Part 1: Model, Notation and Service Definition, February 1988.

Other References

[MAP30] MAP 3.0 Network Management Specification, August 1988.

Editor's Note: Section editors whose text cites these references will keep them up-to-date and will provide additional references as needed, e.g., most recent ISO "N" number and date will be provided.

18.2 SCOPE AND FIELD OF APPLICATION

The purpose of this section (Section 18), is to provide implementation agreements that will enable independent vendors to supply customers with a diverse set of networking products that can be managed as part of an integrated environment. Where possible, these agreements are based upon OSI Network Management standards.

Due to the broad scope of the subject, and given that OSI Management standards are still evolving, it is reasonable to assume that a comprehensive set of network management implementors agreements will take a number of years to develop. In order to arrive at an initial set of implementation agreements in a timely fashion, a phased approach has been adopted.

As a first step in this phased approach, the NMSIG has targeted that the initial, Phase 1, interim agreements will be completed by December, 1989. These Phase 1 agreements provide limited interoperable management in a heterogeneous vendor environment. They are the cornerstone of our eventual comprehensive inventory of OSI-compatible management agreements. Furthermore, these initial agreements allow the community to gain experience with OSI management standards as they emerge.

The scope of the problem addressed in Phase 1 has been constrained in several ways. The sections below outline the nature of these constraints and thereby serve to clarify the scope and field of application associated with this version of the implementors agreements (December 1989). Subsequent phases of these agreements (post December 1989) will expand the scope of problems addressed.

Editor's Note: The following phase definitions and milestones represent the current workplan of the NMSIG. The target dates are the earliest dates at which the milestones could possibly be accomplished and depend (in part) on optimistic assumptions about the progress of relevant standards.

The scope of Phase 1 IA's will be the following:

Management Functions:
Object Management, State Management, Relationship Management, Error Reporting and Event Control

Management Information:
Information Model, Naming, Guidelines and Template for Defining Managed Objects

Management Communication:
CMIS/P, Association Policies, and Services Required

Management Object:
Support Objects required for above and 14 Managed Object Definitions under development by the OSI MIB WG

Conformance Criteria:
TBD depending on the progress of relevant ISO documents.

The milestones for Phase 1 IAs and earliest target dates are:

Milestone A: [12/89]
Freeze the scope of Phase 1 and approve first draft text for Ongoing IAs that cover all of Phase 1 except Managed Objects and Conformance Criteria.

Milestone B: [3/90]
Add the Phase 1 Managed Objects to the Ongoing IAs.

Milestone C: [6/90]
Align the Ongoing IAs pertaining to Phase 1 with ISO DIS documents. Add conformance

criteria pertaining to Phase I to the Ongoing IAs.

Milestone D: [9/90]
Progress the Ongoing IAs pertaining to Phase 1 into Stable IAs.

The preliminary milestones and earliest target dates for Phase 2 are:

Milestone E: [3/90]
Define the Scope of Phase 2 IAs.

Milestone F: [9/90]
Freeze the Scope of Phase 2 IAs and approve the first draft text covering all of Phase 2.

It is the intention of the NMSIG to freeze the content of Phase 1 at Milestone A. Only those changes required to align with the ISO DIS's will be made.

It is the intention of the NMSIG to define Phase 2 functionality as a compatible superset of Phase 1.

The following is an outline of the information provided in these agreements (Section 18):

Section 18.2-- SCOPE AND FIELD OF APPLICATION (This section): This section covers several areas. Specifically:

- Section 18.2.1 describes the relationship between these agreements and the evolving international management standards.

- Section 18.2.2.1 provides a brief overview of the management architecture described in the standards documents.

- Section 18.2.2.2 identifies the constraints imposed on Phase 1 of these agreements.

- Section 18.2.2.3 addresses migration strategies regarding subsequent phases of these agreements.

- Section 18.2.2.4 addresses interoperability with systems associated with other management specifications (including MAP/TOP) [MAP30].

- Section 18.2.3 presents an overview of the functionality supported by Phase 1 of these agreements.

Section 18.3 -- STATUS: This section describes the current status of these agreements.

Section 18.4 -- ERRATA: Once this document is incorporated into a version of the Stable Implementation Agreements for Open System Interconnection Protocols, this section will contain corrections to the stable management agreements. In addition, this section documents interim resolutions to defects found in the management standards.

Section 18.5 -- MANAGEMENT FUNCTIONS: This section documents agreements pertaining to the Systems Management Functions. In addition, it identifies agreements pertaining to the use of other application service elements (e.g. the Common Management Information Service Element (CMISE)).

Section 18.6 -- MANAGEMENT COMMUNICATIONS: This section identifies, in detail, the following:

- Agreements on Association Policies
- Agreements on the Common Management Information Services (CMIS) offered.
- Common Management Information Protocol (CMIP) agreements.
- Agreements pertaining to the services required by CMIP.

Section 18.7 -- MANAGEMENT INFORMATION: This section is based on evolving ISO documents [MIM] and [GDMO], and provides tutorial material and agreements for management information related concepts and modelling techniques. Sub-sections introduce the information model, list principles for naming managed objects and attributes, and provide guidelines for defining management information.

Managed object definitions are outside the scope of this section, and are provided in the Management Information Library (MIL). (The MIL is produced by the OSI MIB Working Group, a subgroup of the NMSIG.)

Section 18.8 -- IMPLEMENTATION PROFILES/CONFORMANCE CLASSES: This section describes the implementation profiles/conformance classes that are used to categorize management products. At the highest level, products fall into two broad categories: systems that take on a managing system role and systems that take on an agent system role representing managed objects. (Refer to Section 18.2.2 for further clarification regarding these categories.) Phase 1 of these agreements defines implementation profiles/conformance classes only for systems that take on an agent system role.

Editor's Note: The NMSIG intends for Phase 1 to ensure that the interface between managing processes and agent processes is adequately specified, thereby enabling the development of interoperable managing processes and agent processes. It is believed that, by identifying implementation profiles/conformance classes only for systems that take on an agent system role, we will also have sufficiently identified the expected behavior of systems that take on a managing system role.

Section 18.9 -- CONFORMANCE: For each of the classes identified in Section 18.8, this section outlines the criteria used to determine whether or not a given product conforms to the class specification that it purports to be. More to the point, in conjunction with Phase 1:

- Systems that take on an agent system role will be tested, via interactions with a test managing system to ensure that they appropriately represent those managed objects that they purport to represent.

  Editor's Note: Although systems that take on a managing system role are not to be tested for conformance in Phase 1, it is believed that market presence of conformant systems that take on an agent system role will provide an adequate climate for determining the suitability of systems that take on a managing system role.

Section 18.10 -- REGISTRATION REQUIREMENTS: This section identifies the management entities that must be registered. This includes a listing of those managed objects that must be defined in order to satisfy the functional requirements outlined in the Phase 1 agreements.

In addition, this section describes the mechanisms used to register management entities and the means by which one can obtain information about a registered entity.

18.2.1 Use of Evolving Standards

In general, it is the intent of the NMSIG to base these implementors agreements on existing international management standards.

Editor's Note: Table 18.1 below shows the relevant standards documents and the current schedules for progressing these documents to the IS status. The table describes the work items and associated target dates approved at the Fifth SC 21/WG 4 Meeting in Sydney, November 29 - December 9, 1988.

Table 18.1 RELEVANT STANDARDS DOCUMENTS AND THE CURRENT SCHEDULES FOR PROGRESSING THESE DOCUMENTS TO IS STATUS

	Target Dates		
Document	DP	DIS	IS
Management Framework	9/86	6/87	10/88
Systems Management Overview	12/88	8/89	8/90
Structure of Management Information			
Part 1: Management Information Model	5/89	4/90	4/91
Part 2: Definition of Support Management Objects	12/88	4/90	4/91
Part 3: Definition of Management	12/88	4/90	4/91
Attributes	12/88	4/90	4/91
Part 4: Guidelines for the Definition of Managed Objects	10/89	9/90	9/91
Common Management Information Service		9/88	9/89
Addendum 1: CancelGet	12/88	9/89	8/90
Addendum 2: Add/Remove	12/88	9/89	8/90
Common Management Information Protocol		9/88	8/89
Addendum 1: CancelGet	12/88	9/89	8/90
Addendum 2: Add/Remove	12/88	9/89	8/90
Configuration Management			
Systems Management - Part 1: Object Management Function	12/88	7/89	7/90
Systems Management - Part 2: State Management Function	12/88	4/90	4/91
Systems Management - Part 3: Relationship Management Function	12/88	4/90	4/91
Fault Management			
Systems Management - Part 4: Error Reporting and Information Retrieval Function	12/88	4/90	4/91
Systems Management - Part 5: Service Control Function	12/88	4/90	4/91
Systems Management - Part 6: Confidence and Diagnostic Testing Function	10/89	7/90	7/91
Systems Management - Part 7: Log Control Function	10/89	7/90	7/91
Security Management	10/89	7/90	7/91
Accounting Management	10/90	3/92	3/93
Performance Management	10/89	7/90	7/91

Given the current state of the standards, the ongoing Phase 1 implementors' agreements are based on documents, some of which are not yet at the DIS level. In addition, in order to meet the stated objectives of the Phase 1 agreements, some agreements have been formed in advance of the availability of DP's in the relevant areas.

As the relevant standards documents progress to DIS and IS, the agreements will be aligned.

Thus subsequent phases of these agreements will incorporate the relevant standards information as the standards become available. In general, the NMSIG will attempt to incorporate information from a standard that has progressed to the DIS or IS state into the subsequent phases of the implementors' agreements.

When a defect is found in any of the management related standards, the reported defect may be technically resolved by the appropriate international technical committee with likely approval by the voting members pending for several months. Since relevant defects can't be ignored in an implementation, these agreements will note defect resolutions which have the tentative approval of the appropriate standards committee. These interim resolutions will be recorded in Section 18.4.

Once a defect resolution has been finalized by the appropriate standards body, the agreed upon resolution will be incorporated into the next phase of these implementors agreements. If appropriate, a previous phase that relied on an interim resolution will be examined to determine whether or not errata should be issued to bring the original phase into line with the final resolution.

18.2.2 Management Architecture

18.2.2.1 Systems Management Overview

Editor's Note: This section is tutorial.

Reference [SMO] provides an overview of the OSI Systems Management Architecture. What follows is a brief summary of the information contained therein. The material contained here (i.e. Section 18.2.2.1) is tutorial in nature. It is not intended to correct deficiencies that may exist in the standards themselves. This information is primarily intended to serve as an aid to the casual reader of these requirements. For more detail, please refer to the management standards referenced below.

STANDARDS

The OSI System management standards are grouped as follows:

- References [FRMWK] and [SMO] address the general concepts.

- References [ALS], [CMIS], and [CMIP] address the communications standards.
- References [MIM], [DSO], [DMA], and [GDMO] pertain to the definition of management information (managed objects).

- References [CMO], [FMWD], [SMWD], [AMWD], and [PMWD] document functional area standards.

  Editor's Note: Due to reorganization of documents as a result of the December 1988 SC21/WG4 meeting in Sydney, functions have been separated from the management functional areas which originally developed them. The documents which describe these functions include [OMF], [SMF], [RMF], [ERIRF], and [MSC].

GENERAL CONCEPTS

Viewed abstractly, a communications environment is made up of a collection of managed objects. Management of the communications environment is viewed as being an information processing application. Management activities are carried out by using the information processing application to manipulate and monitor the managed objects that make up the environment.

Because the environment being managed is physically distributed, the components of the information processing application are themselves distributed. These distributed components take the form of management application processes. These distributed application processes may be organized in many ways, as for example, in a hierarchical manner or on a peer-to-peer basis.

Management processes are divided into two categories: managing processes and agent processes. A managing process is that part of a distributed application process that is responsible for carrying out one or more management activities. An agent process is responsible for manipulating and monitoring an associated set of managed objects. A managing process interacts with an agent process to carry out the management activities for which it is responsible.

An agent process performs the management function upon receipt of a message specifying management operations on managed objects. Agent processes may also forward messages to managing processes to convey information generated by managed objects.

APPLICATION LAYER COMMUNICATIONS

A systems management application entity (SMAE) is that portion of a management process that is responsible for communicating with other management processes (or more specifically, other SMAE's). A SMAE is made up of a collection of cooperating application service elements (ASE's).

The association control service element (ACSE) is used to establish associations with other SMAE's. Once this is done, a systems management application service element (SMASE) is used to exchange information between the associated SMAE's. The SMASE realizes the abstract notion of messages exchanged between management processes.

The SMASE relies on other (standard) ASE's to effect communications. Notably, the services of the common management information service element (CMISE) are used.

Taken as a whole, a SMAE ultimately relies on presentation layer services to communicate.

FUNCTIONAL AREAS

Systems manAgement activities are grouped into five functional areas that are intended to capture the user requirements imposed on management. These functional areas are:

- Configuration Management
- Fault Management
- Security Management
- Performance Management
- Accounting Management

Each of these functional areas is referred to as a Specific Management Functional Area (SMFA). Each SMFA gives rise to a standard that identifies the following:

- A set of functions that support the functionality within the scope of the SMFA.

- The procedures associated with the provision of each function.

- The services required to support these procedures.

- The use of the underlying OSI services to provide the communications needs.

- The classes of managed objects that the procedures will operate upon in order to provide the functionality defined by the SMFA.

18.2.2.2 Constraints/Assumptions for Phase 1

The focus of the Phase 1 agreements is to enable a managing process provided by one vendor to interoperate with an agent process provided by a different vendor for the purpose of performing limited management on a set of managed objects. Specifically, these agreements focus on the managing process/agent process interface and the techniques used to define managed objects. These agreements do not address (nor constrain) the mechanisms used by agent processes to manipulate managed objects. Nor should these agreements inhibit our ability to provide post-Phase 1 agreements that meet the long term goals associated with the area of network management.

In order to accomplish this goal in a timely fashion, several simplifying constraints have been imposed on these agreements. These constraints are summarized below.

1. These agreements support only a limited set of functionality. Refer to Sections 18.2.3 and 18.5 for a description of the functionality supported by these agreements.

2. No agreements are provided in support of managing process to managing process communications.

3. No agreements are provided regarding management domains.

4. All communications supported by these agreements rely on the use of the following application service elements: the association control service element (ACSE), the common management information service element (CMISE), Remote Operations Service Element (ROSE), and the system management application service element (SMASE) identified in Section 18.6.

5. All communications between managing processes/agent processes are based on connection-oriented presentation services.

6. These agreements do not rely on the use of Directory Services.

7. No agreements regarding the security of management are provided except for the use of access control on association initialization.

Editor's Note: The NMSIG has requested, via a liaison statement, that the Security SIG suggest appropriate security agreements to address this area. In the absence of input from the Security SIG, it should be noted that individual management products may implement proprietary security policies that do not interfere with interoperability. For example, a given managing process or agent process may decide to refuse an A-Associate request based on the calling presentation address and some locally defined criteria.

8. It is assumed that every managed object instance will be associated with exactly one agent process. This agent process is responsible for acting as the agent for the managed object with regard to all interactions with the managing systems.

18.2.2.3 Migration to Future Phases

Editor's Note: This section will document the migration plans with regard to ensuring that Phase N products can interact with Phase 1 products.

18.2.2.4 Relationship to Other Management Specifications

Editor's Note: This section will describe the degree to which implementations that conform to these agreements will interoperate with implementations that conform to the other management specifications (including MAP/TOP).

18.2.3 Management Scenarios

Editor's Note: The intent of this section is to amplify the high level NM requirements to be met by these IAs. In particular, this section will provide a high level view of the functionality supported by Phase 1 of these agreements. Based on these scenarios, one should be able to determine the scope of managed object classes that are required to satisfy these scenarios.

18.3 STATUS

Section 18 is currently a working draft of the Phase 1 Network Management Implementors Agreements.

Editor's Note: The intention is to possibly move at least some of this material to stability in 1990. Therefore, the content of this chapter should be closely examined.

18.4 ERRATA

(None as yet)

18.5 MANAGEMENT FUNCTIONS AND SERVICES

Editor's Note: To aid the casual reader, parts of this section have been written in a tutorial fashion, explaining unclear or obscure areas in the base standards. This material will be deleted when transition to the Stable Agreements Document occurs. The remaining material contains agreements relative to the base standards or to areas deemed important for interoperability but not contained in the base standards.

Editor's Note: Tutorial Material. ISO has partitioned network management into five Specific Management Functional Areas (SMFAs) as a convenience for developing requirements particular to configuration management (CM), fault management (FM), performance management (PM), security management (SM), and accounting management (AM). These requirements are specified in five separate SMFA standards ([CMO], [FMWD], [SMWD], [AMWD], and [PMWD]). Due to reorganization of documents as a result of the December 1988 SC21/WG4 meeting in Sydney, functions have been separated from the management functional areas which originally developed them. The documents which describe these functions include [OMF], [SMF], [RMF], [ERIRF], [LCF], and [MSC].

Since the SMFAs have overlapping requirements, management functions and management information applicable to one SMFA are often applicable to other SMFAs. Therefore, the SMFAs point to separate standards that contain the management functions needed to satisfy particular requirements.

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

The following schematic depicts the functional hierarchy of SMFs and SMFAs. There are seven common SMFs. They provide much of the network management capabilities needed by CM and FM. When additional requirements are identified in other SMFAs, additional SMFs may be developed.

```
                   Applications
                     |    various requirements result in
                     |    various groupings of functional
                     |    management areas
                 +--------+--------+---------+--------+
                 |        |        |         |        |

===================================================================
        |
        |    +----+    +----+   +----+   +----+   +----+
SMFAs   |    | FM |    | CM |   | PM |   | SM |   | AM |
        |    +----+    +----+   +----+   +----+   +----+
        |       |        |        |        |        |

===================================================================
SMFs    |                     PLATFORM
        |  +-------------+ +----------------------+ +-----------+
        |  |Event Control| |Service Access Control| |Log Control|
        |  +-------------+ +----------------------+ +-----------+
        |
        |  +---------------+ +-----------------+  +------------+
        |  |Error Reporting| |Error Information|  |Relationship|
        |  +---------------+ |   Retrieval     |  | management |
        |                    +-----------------+  +------------+
        |
        |  +----------------+   +-----------------+ +--------------+
        |  |State Management|   |Object Management| | Confidence & |
        |  +----------------+   +-----------------+ |  Diagnostic  |
        |                                           |     Test     |
        |                                           +--------------+
        |                     (etc ....)

===================================================================
        |
        |                      CMIS
        |

===================================================================
        |            Lower Layer Services
```

The following System Management Functions are undergoing standardization:

(1) Object Management Function [OMF]

(2) State Management Function [SMF]

(3) Relationship Management Function [RMF]

(4) Error Reporting and Information Retrieval Function [ERIRF]:

a. Error Reporting Service

b. Information Retrieval Service

(5) Management Service Control Function [MSC]:

a. Event Control Service

b. Service Access Control Service

(6) Event Log Control Function [LCF]

(7) Confidence and Diagnostic Test Function [FMWD].

For the NIST NMSIG Phase 1 network management agreements, it is agreed that only the first six functions will be supported. For each supported System Management Function (Sections 18.5.1-18.5.6, below), agreements pertinent to the accompanying management communication services are given.

18.5.1 Object Management Function Agreements

Editor's Note: Tutorial Material. This System Management Function provides the management of Objects in an Open System Environment. In this environment, a managed object (MO) can be identified as an abstraction of a data processing resource or a data communications resource that can be remotely managed through the use of OSI management communication Services (Section 18.6). An MO may be a physical item of equipment, a software component, or a combination of such. Each MO has a set of management information associated with it and an MO identifier by which the set of management information can be manipulated through the use of the OSI management communications services.

The NMSIG Phase 1 network management agreements support all the operations and services in the object management standard [OMF], i.e.,

- Object creation operation
- Object deletion operation
- Object renaming operation
- Attribute reading operation
- Attribute changing operation
- Object listing operation
- Enrol Object Service
- Deenrol Object Service
- Reenrol Object Service
- Attribute Change Event Report Service

- o Add Value Event Report Service
- o Remove Value Event Report Service

For the last three services listed above, the Event Reporting Control Model (Section 18.5.5) applies.

18.5.1.1 Object Creation Operation Agreements:

Editor's Note: Tutorial Material. The Object Creation operation is used by a managing system to ask a managed system to create an instance of a managed object in the managed system.

The following agreements and clarifications pertinent to Section 8.1 of the base standard [OMF] and regarding the semantics of the confirmed CMIS M-CREATE service (Section 8.3.4 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-CREATE request parameters:

<invokeIdentifier>

<managedObjectClass>

<managedObjectInstance> (1) If this parameter is used in the request, it will identify the distinguished name of the object instance to be created. The distinguished name of a managed object instance is created by concatenating in sequence (ordered list) the relative distinguished names of its superiors in the containment tree starting at the root and working downward towards the managed object instance to be identified.

(2) Otherwise, the performing CMISE-service-user will assign a value to this

identification of this instance.

The managed object definition will specify whether the manager or agent will provide the <managedObjectInstance> value. This means that for a given object class either (1) must always be used or (2) must always be used (refer to Section 6.1.5.2.1 of [MIM]).

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<referenceObjectInstance> When this parameter is used by the invoking CMISE-service-user, it must specify an existing object instance of the same class as the object being created.

<attributeList> This parameter must provide the attribute(s) and their initial value(s) for the object instance if they are neither provided as defaults in the object definition nor obtained from the reference object. Otherwise, a CMIS error of <invalidAttributeValue> will be returned (Section 8.3.4.1.8 of [CMIS]).

Editor's Note: If an error code of <missingAttributeValue> is defined in the standard in the future, it will be adopted here.

Editor's Note: The standards as written do not show any way (via the ATTRIBUTE macro) to define a default value for an attribute. We are assuming that it is possible to define such default values. However, it is not required that this be done for EVERY attribute.

CMIS M-CREATE response parameters:

<invokeIdentifier>

<managedObjectClass>

<managedObjectInstance> Refer to Section 18.6.3.2.8 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter specifies all of the created object attributes and values.

Editor's Note: It is anticipated that Section 18.6 of this chapter will define this in common for all M-CREATE's, at which time, the text here can refer to that section directly.

<currentTime> Refer to Section 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

Editor's Note: Can any manager other than the manager that created the object manage this new object?

Over which association(s) can this new object be managed?

- o the current association?
- o other extant associations?
- o new associations?

This issue is to be determined as part of the general association policy.

Note that there is a more general problem which applies to access rights and ownership of the created objects. Maybe the protocol section should set the policy for the CMIS M-CREATE service?

18.5.1.2 Object Deletion Operation Agreements:

Editor's Note: Tutorial Material. The Object Deletion operation is used by a managing system to ask a managed system to delete an instance of a managed object in the managed system.

The following agreements and clarifications pertinent to Section 8.3 of the base standard [OMF] and regarding the semantics of the confirmed CMIS M-DELETE service (Section 8.3.5 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-DELETE request parameters:

<invokeIdentifier>

<baseManagedObjectClass> (1) If scoping is used for multiple object selection, this parameter identifies the managed object class that is to be used as the starting point for the selection of managed objects on which the filter is to be applied.

(2) If scoping is used to select the base object only, this parameter identifies the class of the object instance to be deleted.

Editor's Note: <n> level delete is to be discussed further.

<baseManagedObjectInstance> (1) If scoping is used for multiple object selection, this parameter identifies the instance

of the managed object that is to be used as the starting point for the selection of managed objects defined by <scope> on which the filter is to be applied.

(2) When a single object is targeted for deletion (i.e. the scope is base managed object alone), this parameter specifies the managed object instance to be deleted.

Editor's Note: <n> level delete is to be discussed further.

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <BestEffort> is required.

<scope> This parameter defines the level(s) relative to the base managed object from which objects will be deleted. This is used for deleting multiple object instances. It will be set to <baseObject> if single object selection is used, or set to <n> to specify the depth of the search, or specify the whole subtree.

Editor's Note: <n> level delete is to be discussed further.

<filter>

CMIS M-DELETE response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass> <managed Object Instance>	Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.
<currentTime>	Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

18.5.1.3 Object Renaming Operation Agreements:

Editor's Note: Tutorial Material. The Object Renaming operation is used by a managing system to ask a managed system to rename an instance of a managed object in the managed system.

Editor's Note: This section is very controversial. We do not feel that we have a clear understanding of what an OBJECT NAME is. The standard seems to imply that the OBJECT NAME is the distinguishing attribute defined in the object definition. If this is so, it is a <readonly> attribute, and cannot be changed by a CMIS M-SET service. The group feels that it is more appropriate to use a specific CMIS M-ACTION service to carry out this specific operation. The group will submit comments, in this regard, to ISO by the March 1989 ANSI meeting.

The following section aligns with the current standard and may change.

Editor's Note: It is anticipated that this service will have side effects, especially with regard to associations where objects existed with old names, regarding operations with the objects under old names, and regarding discriminator object changes at the managed object's systems as well as the destination system.

The Object Renaming Operation is not supported in the network management Phase 1 IAs.

18.5.1.4 Attribute Reading Operation Agreements:

Editor's Note: Tutorial Material. The Attribute Reading operation is used by a managing system to ask a managed system to return the specified attribute values for an instance of a managed object in the managed system.

The following agreements and clarifications pertinent to Section 8.8 of the base standard [OMF] and regarding the semantics of the confirmed CMIS M-GET service (Section 8.3.1 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-GET request parameters:

<invokeIdentifier>	
<baseManagedObjectClass>	
<baseManagedObjectInstance>	
<accessControl>	Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.
<synchronization>	<bestEffort> is required.
<scope> <filter>	
<attributeIdList>	This parameter list will contain the list of attributes to be retrieved. If the list is not provided, all attributes will be retrieved.

CMIS M-GET response parameters:

<invokeIdentifier>	
<linkedIdentifier>	
<managedObjectClass> <managedObjectInstance>	Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter, provided by the managed system, returns the list of ids of these requested attributes and the values of these attributes.

If an error occurs in the retrieval process, a CMIS ERROR <GetListError> will be reported. The list will include all requested attributes, and for each attribute there will be chosen either the attribute value (choice of Tag [1]) for the successful retrieval of an attribute, or an attributeIdError (choice of Tag [0]) for the failure case. Refer to Section 8.3.1.1.14 in [CMIS] for more information.

18.5.1.5 Attribute Changing Operation Agreements:

Editor's Note: Tutorial Material. The Attribute Changing operation is used by a managing system to ask a managed system to change the values of one or more specified attributes for a managed object instance in the managed system.

The following agreements and clarifications pertinent to Section 8.9 of the base standard [OMF] and regarding the semantics of the confirmed CMIS M-SET service (Section 8.3.2 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-SET request parameters:

<invokeIdentifier>

<mode> This parameter will be set to 'confirmed'.

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeList> This parameter will contain the list of attributes whose values are to be modified and the desired new values.

CMIS M-SET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter, provided by the managed system, returns the list of attribute ids of the modified attributes and their modified values.

If an error occurs in the process, a CMIS ERROR <SetListError> will be reported. The list will include all attributes requested for modification, and for each one, choose either an <attribute> (choice of Tag [1]) for the successful modification of an attribute, or an <attributeError>

(choice of Tag [0]) for the failure case. Refer to (Section 8.3.2.1.14 in [CMIS]) for more information.

18.5.1.6 Object Listing Operation Agreements:

Editor's Note: Tutorial Material. The Object Listing operation is used by a managing system to ask a managed system to retrieve the names of a defined set of managed objects in the managed system. Other attributes can also be retrieved by specifying the attribute names in the request.

The following agreements and clarifications pertinent to Section 8.7 of the base standard [OMF] and regarding the semantics of the confirmed CMIS M-GET service (Section 8.3.1 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

Editor's Note: This section is controversial because we must again work with the problematic definition of an OBJECT NAME. Comments will be submitted to the ANSI meeting in March 1989.

The following section assumes that the OBJECT NAME is the same as the <Name> attribute which represents the distinguished Name.

CMIS M-GET request parameters:

<invokeIdentifier>

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeIdList> (1) If this parameter is used, the list will include at least the <Name> attribute.

(2) If the list is not provided, all attributes including the <Name> attribute will be retrieved.

CMIS M-GET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter, provided by the managed system, returns the attribute ids and values for the specified attributes (including the object name(s) of the requested managed object's <Name> attribute).

If an error occurs in the retrieval process, a CMIS ERROR <GetListError> will be reported. (Section 8.3.1.1.14 in [CMIS])

18.5.1.7 Object Management Services Agreements

Editor's Note: Tutorial Material. Each of the Object Management Services uses an unconfirmed M-EVENT-REPORT CMIS service (Section 8.3.1 in [CMIS]) to convey its information.

The Event Reporting Model (see Section 18.5.5 in this chapter and [ERIRF], [MSC], [DSO]) defines the following procedure: The agent receives notifications from the appropriate managed objects and causes these potential event reports to be checked against all Event Forwarding Discriminators. The result of this sieve process will yield zero, one or more event reports to be transmitted to the destination systems (according to the attributes of the relevant discriminators) according to the services defined in the subsequent sub-sections. One discriminator may cause the sending of multiple event reports, if the multi-valued attribute ManagementUserIdentification contains multiple AEtitles. Additionally, multiple discriminators may filter the same potential event reports and hence generate multiple event reports.

Editor's Note: Some of the text in this paragraph should be moved to the discussion of the Event Reporting Model in 18.5.4, while retaining some here.

The following agreements and clarifications pertinent to Sections 8.2, 8.4, 8.6, 8.10, 8.11, and 8.12 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements for all the Object Management Services Sections 8.5.1.7.1 through 8.5.1.7.6, below):

<invokeIdentifier>	
<mode>	This parameter is set to <unconfirmed>.
<managedObjectClass> <managedObjectInstance>	Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

18.5.1.7.1 Enrol Object Service Agreements

Editor's Note: Tutorial Material. The Enrol Object Service is used by the managed system to report a creation event of a new managed object instance to a managing system.

In addition to the agreements and clarifications in Section 18.5.1.7, the following agreements and clarifications pertinent to Section 8.2 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements:

CMIS M-EVENT-REPORT request parameters:

<eventType>	This parameter identifies the <enrolObject> Event whose object identifier is defined in [OMF].
<eventTime>	This parameter specifies the time when the new instance was created. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.
<eventArgument>	This parameter is not used for this service.

18.5.1.7.2 Deenrol Object Service Agreements:

Editor's Note: Tutorial Material. The Deenrol Object Service is used by the managed system to report the deletion of a managed object instance to a managing system.

In addition to the agreements and clarifications in Section 18.5.1.7, the following agreements and clarifications pertinent to Section 8.4 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements:

<eventType>	This parameter identifies the <deenrolObject> Event whose object identifier is defined in [OMF].
<eventTime>	This parameter specifies the time when the object instance was deleted. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.
<eventArgument>	This parameter is not used for this service.

18.5.1.7.3 Reenrol Object Service Agreements:

Editor's Note: Tutorial Material. The Reenrol Object Service is used by the managed system to report the renaming of a managed object instance to a managing system.

The Reenrol Object Sevice is not supported in the network management Phase 1 IAs.

18.5.1.7.4 Attribute Change Event Report Service Agreements:

Editor's Note: Tutorial Material. The Attribute Change Event Report Service is used by the managed system to report an attribute change event to the managing system. The attribute change event indicates a change in the value(s) of one or more attributes of a managed object.

In addition to the agreements and clarifications in Section 18.5.1.7, the following agreements and clarifications pertinent to Section 8.10 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements:

<eventType> This parameter identifies the <attributeChange> Event whose object identifier is defined in [OMF].

<eventTime> This parameter specifies the time when the attribute value was changed in the object instance. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<eventArgument> This parameter will contain the tuple <attributeId, oldAttributeValue, newAttributeValue> (Section 9 in [OMF]). The oldAttributeValue must be presented.

18.5.1.7.5 Add Value Event Report Service Agreements:

Editor's Note: Tutorial Material. The Add Value Event Report Service is used by the managed system to report the addition of a value to a multi-valued attribute of a managed object at an open system.

In addition to the agreements and clarifications in Section 18.5.1.7, the following agreements and clarifications pertinent to Section 8.11 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements:

<eventType> This parameter identifies the <addValue> Event whose object identifier is defined in [OMF].

<eventTime> This parameter specifies the time when the new attribute value was added to the object instance. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<eventArgument> This parameter will contain the tuple <attributeId, newAttributeValue>, where <newAttributeValue> is the attribute value just added. (Section 9 of [OMF]).

18.5.1.7.6 Remove Value Event Report Service Agreements:

Editor's Note: Tutorial Material. The Remove Value Event Report Service is used by the managed system to report the removal of a value from a multi-valued attribute of a managed object at an open system.

In addition to the agreements and clarifications in Section 18.5.1.7, the following agreements and clarifications pertinent to Section 8.12 of the base standard [OMF] and regarding the semantics of the CMIS M-EVENT-REPORT parameters are supported by the Phase 1 network management agreements:

<eventType> This parameter identifies the <removeValue> Event whose object identifier is defined in [OMF].

<eventTime> This parameter specifies the time when the attribute value was deleted from the object instance. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<eventArgument> This parameter will contain the tuple <attributeId, oldAttributeValue>, where <oldAttributeValue> is the attribute value just deleted. (Section 9 of [OMF]).

18.5.2 State Management Function Agreements

Editor's Note: Tutorial Material. The State Management Function provides for the examination, setting and notification of changes in the management state of existing managed objects. The managed state of a managed object represents its instantaneous condition of availability and operability from the point of view of configuration management. The managed state consists of (1) operational state, and (2) administrative state.

A list of the possible combinations of the operational and administrative states is given in (Table 1, Section 7.2, [SMF]). The purpose of this list is to control the availability of a managed object, and to make visible information about the general availability of a managed object.

The Phase 1 network management agreements support the two operations and one service defined in the base standard (Section 8 of [SMF]), i.e.,

- o State reading operation
- o State changing operation

- State change reporting service.

For the State change reporting Service, the Event Reporting Control Model (Section 18.5.5.1.1) applies.

18.5.2.1 State Reading Operation Agreements:

Editor's Note: Tutorial Material. The state reading operation enables the managing system to request the managed system to return the values of the configuration state attributes which include the operational and/or administrative state(s) of one or more instances of managed object(s).

The following agreements and clarifications pertinent to Section 8.1 of the base standard [SMF] and regarding the semantics of CMIS M-GET service (Section 8.3.1 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below. CMIS M-GET request parameters:

<invokeIdentifier>

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeIdList> This parameter list will include the list of state attribute(s) (<operational state>, <administrative state>) which the managing system would like to obtain. If the list is not provided, all attributes including the state attributes will be retrieved.

CMIS M-GET response paramaters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter, provided by the managed system, returns the list of requested state attributes and their values.

If an error occurs in the retrieval process, a CMIS ERROR <GetListError> will be reported. (Section 8.3.1.1.14 in [CMIS])

18.5.2.2 State Changing Operation Agreements:

Editor's Note: Tutorial Material. The state changing operation enables the managing system to request the managed system to change the value of the administrative state attribute of one or more instances of a managed object(s).

The following agreements and clarifications pertinent to Section 8.2 of the base standard [SMF] and regarding the semantics of CMIS M-SET service (Section 8.3.2 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-SET request parameters:

<invokeIdentifier>

<mode> 'Confirmed' is to be used.

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeList> This parameter will include the state attribute (<administrativeState>) and its desired new value.

CMIS M-SET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This parameter, provided by the managed system, returns the attribute ids and values for the specified attributes (including the modified state attribute).

If an error occurs in the process, a CMIS ERROR <SetListError> will be reported. (Section 8.3.2.1.14 in [CMIS])

18.5.2.3 State Change Reporting Service Agreements:

Editor's Note: Tutorial Material. The state change reporting service enables the managed system to report the change of a state attribute (i.e. either the operational state or administrative state) of a managed object to a managing system.

The following agreements and clarifications pertinent to Section 8.3 of the base standard [SMF] and regarding the semantics of CMIS M-EVENT-REPORT service (Section 8.2.1 in [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

<invokeIdentifier>

<mode> This parameter is set to <unconfirmed>.

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<eventType> This parameter identifies the <stateChange> Event whose object identifier is defined in [DMA].

<eventTime> This parameter specifies the time when the object instance state attribute value was changed. Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

<eventArgument> This parameter will contain the tuple <oldConfigurationState, newConfigurationState> for the newly changed state object instance [DMA].

18.5.3 Relationship Management Function

Editor's Note: Tutorial material. A relationship is a set of rules that describe how the operation of one part of an open system affects the operation of another part of an open system. The operation of a managed object may affect its related managed object directly or indirectly. A direct relationship exists between two managed objects when some portion of the management information associated with one managed object expressly identifies the other managed object with which it has a relationship. Indirect relationship information can be deduced from the concatenation of two or more direct relationships.

In order to manage the relationship information of two directly related managed objects, a relationship can be modelled as a third object, or a pair of bound attributes, one for each of the related managed objects. The latter approach is the one currently taken by the ISO OSI management standard [RMF]. The relationship is presented by explicitly including, as one of a set of values of each bound attribute of the pair, the name of the other managed object to which it is related. This binding is called an explicit relationship. Therefore, an explicit relationship between a pair of managed objects can be represented by a pair of conjugate values of the bound relationship attributes of the two managed objects.

At any given time, within an open systems environment, one managed object may be a part of several different types of relationships. For each type of relationship, depending on the roles of the managed objects (i.e. the set of rules governing the interactions between the two related managed objects), the relationship can be symmetric or asymmetric. If the roles of the two managed objects are the same, then the relationship (role) is symmetric, otherwise it is asymmetric. For every possible relationship role of a managed object, there exists a corresponding relationship attribute. Hence, in order to describe a symmetric relationship, two bound attributes of the same role-type of relationship attributes are needed. To describe an asymmetric relationship, two role-types of bound relationship attributes are needed. The name of a relationship attribute of a managed object implies the relationship role of the related managed objects and the type of the explicit relationship. The value of a relationship attribute for a managed object may be multi-valued or "null". These values are the names of the associated managed objects having the same type of explicit relationship with the managed object.

The types of explicit relationships defined in the standards [RMF] are: 1) Service relationship which can be described by relationship attributes of the Service Provider and the Service User; 2) Peer relationship which is a symmetric relationship and is described by the Peer attribute type; 3) Backup relationship which can be described by relationship attributes of the "Primary" operation object and the "Secondary" backup object; and 4) Group relationship which can be described by relationship attributes of "Member" and "Owner".

The collection of all relationship attributes of one managed object can be named under a group attribute. If defined, this named group attribute will be an attribute of all of the managed object classes.

Between two managed objects there may exist containment relationships in addition to the explicit relationships. A containment relationship is automatically created when the containing/contained managed objects are created. A containment relationship is implicitly reflected in the name (i.e. a sequence of AVAs) of the contained managed object. Managed object naming is part of SMI and is specified during the managed object definition. Therefore, no relationship management service is required to manage containment relationship information.

The Relationship Management Function specified by [RMF] provides the following services to add, remove, change and display the relationship attribute information for managed objects, and to report events of relationship activities.

Relationship Creation is a service which allows the managing process (or the invoker) to request the managed process (or the performer) to add a value to the specified relationship attribute of the specified managed objects in order to reflect a newly created (or to be created) relationship.

Relationship Deletion is a service which allows the invoker to request the performer to remove the value(s) from the set of its relationship attributes of specified managed objects in order to reflect a newly removed (or to be removed) relationship.

Relationship Changing is a service which allows the invoker to request the performer to replace one or more value(s) of the specified relationship attributes of the specified managed objects.

Relationship Listing is a service which allows the invoker to request the performer to return the value(s) of the specified relationship attribute(s) of the specified / selected managed object.

Related Object Listing is a service which allows the invoker to request the performer to return the name(s) and the other specified attribute(s) and value(s) of the selected managed objects which have the specified relationship attribute(s) value(s) which match successfully to the target managed object.

Relationship Creation Reporting is a service which allows a managed process to report the relationship creation event to the managing process(es) (not necessarily the original managing process).

Relationship Deletion Reporting is a service which allows the managed process to report the relationship deletion event to the other process(es) (not necessarily the original managing process).

Relationship Change Reporting is a service which allows the managed process to report the Relationship Change Event to the managing process (not necessarily the original managing process).

Since a relationship is represented by a pair of bound relationship attributes, in order to keep the integrity of relationship management information, it takes at least two services to complete a transaction. The transaction processing and the commitment control are outside the scope of this section.

Editor's Note: The following sections are to be agreed upon.

18.5.3.1 Relationship Creation Service Agreements

Editor's Note: This service is mapped to M-SET CMIS Services. It is assumed that the CMIS

Add/Remove of attribute value function is supported in Phase 1 here.

CMIS M-SET Request parameters clarifications:

<invokeIdentifier>

<mode>

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl>

<synchronization>

<scope> <Base object only> is to be used.

<filter>

<modificationList> This parameter specifies a set of (at least one) tuples: <a specified Relationship Attribute of the selected Managed Object, to-be-added relationship attribute value, add-value operation> for the relationship to be created.

CMIS M-SET Response parameters clarifications:

<invokeIdentifier>

<linkIdentifier> This parameter shall not be returned.

<ManagedObjectClass> Refer to Section 18.6.

<ManagedObjectInstance>

<attributeList> This parameter specifies a set of <Relationship Attribute of the selected managed object, the value that was added> for the relationship created.

<currentTime> Refer to Section 18.6.

18.5.3.2 Relationship Deletion Service Agreements

This Service shall use the M-SET CMIS service to carry its information with the following clarification:

CMIS M-SET request parameters:

<invokeIdentifier>

<mode>

<baseManagedObjectClass> This parameter specifies the base of the Class of the managed objects with whose instance a relationship with another managed object is to be deleted.

<baseManagedObjectInstance> This parameter specifies the instance of the base of managed object with whom a relationship with another managed object is to be deleted.

<accessControl>

<synchronization>

<scope>

<filter>

<modificationList> This parameter specifies a set of <specified Relationship Attribute name,its value to be removed, remove-value operation>.

CMIS M-SET Response parameters:

<invokeIdentifier>

<linkIdentifier>

<ManagedObjectClass> Refer to Section 18.6.

<ManagedObjectInstance>

<attributeList> This parameter specifies a set of <specified Relationship Attribute of the Managed Object, the value that is removed>.

<currentTime> Refer to Section 18.6.

18.5.3.3 Relationship Change Service Agreements

This Service shall use M-SET CMIS service to carry its information with the following clarification:

CMIS M-SET request parameters:

<invokeIdentifier>

<mode>

<baseManagedObjectClass> This parameter specifies the base of the Class of the managed objects with whose instance a relationship with another Managed Object is to be changed.

<baseManagedObjectInstance> This parameter specifies the instance of the base managed object with whom a relationship with another managed object is to be changed.

<accessControl>

<synchronization>

<scope> <base object only>

<filter>

<modificationList> This parameter specifies a set of <specified Relationship Attribute of the selected managed object, the old value to be replaced, the new value, replace operation>.

Editor's Note: This has to be verified, i.e., whether the CMIS DAD2 supports this old and new value syntax. If this is the case, we have to replace the whole set-value of the attribute.

CMIS M-SET Response parameters:

<invokeIdentifier>

<linkIdentifier> This parameter shall not be returned.

<ManagedObjectClass> Refer to Section 18.6.

<ManagedObjectInstance>

<attributeIdList> This parameter specifies a set of <specified Relationship Attribute of the selected managed object, its new value>.

<currentTime> Refer to Section 18.6.

18.5.3.4 Relationship Listing Service Agreements

This Service shall use M-GET CMIS service to carry its information with the following clarification:

CMIS M-GET request parameters:

<invokeIdentifier>

<baseManagedObjectClass> This parameter specifies the base of the managed object class with which instances, the value(s) of their specified relationship attributes, are to be listed.

<baseManagedObjectInstance> This parameter specifies the instance of the managed objects.

<accessControl>

<synchronization>

<scope>

<filter>

<AttributeIdList> This parameter specifies the list of relationship attributes with their relationship value(s) which is(are) to be returned.

CMIS M-GET Response parameters:

<invokeIdentifier>

<linkIdentifier>

<ManagedObjectClass> Refer to Section 18.6.

<ManagedObjectInstance>

<attributeList> This parameter returns a set of <relationship attribute id, attribute value(s)>.

<currentTime> Refer to Section 18.6.

18.5.3.5 Related Object Listing Service Agreements:

This Service shall use M-GET CMIS service to carry its information with the following clarification:

CMIS M-GET request parameters:

<invokeIdentifier>

<baseManagedObjectClass> This parameter specifies the Class of the base managed object from which the related object instances are to be selected for filtering.

<baseManagedObjectInstance> This parameter specifies the instance of the base Managed Object from which the related object instances are to be selected for filtering.

<accessControl>

<synchronization>

<scope> All 3 options are allowed and supported.

<filter> The filter should specify the relationship attribute name and its value to be matched (i.e. the name of the target object to which the selected objects are related).

<AttributeIdList> Refer to Section 18.6.

CMIS M-GET Response parameters:

<invokeIdentifier>

<linkIdentifier>

<ManagedObjectClass> Refer to Section 18.6.

<ManagedObjectInstance>

<attributeList> This parameter returns a set of <the name of the related object, the value(s) of the requested attribute(s)>.

<currentTime> Refer to Section 18.6.

18.5.3.6 Relationship Creation Report Service Agreements

This service uses the unconfirmed M-EVENT-REPORT CMIS service to convey its reporting information. It also uses the event reporting control Function specified in Section 18.5.5.1 to report the events.

CMIS M-EVENT-REPORT request parameters

<invokeIdentifier>

<mode> <unconfirmed>

<managedObjectClass> Refer to Section 18.6.

<managed ObjectInstance> Refer to Section 18.6.

<eventType> This parameter should identify the <RelationshipCreation> Event with the object identifier defined in [OMF].

<eventArgument> This parameter will include a set of <relationship attribute name, the value that was just added to its set-value list>.

18.5.3.7 Relationship Deletion Report Service Agreements

This service uses the unconfirmed M-EVENT-REPORT CMIS service to convey its reporting information. It also uses the event reporting control Function specified in Section 18.5.5.1 to report the events.

CMIS M-EVENT-REPORT request parameters

<invokeIdentifier>

<mode> <unconfirmed>

<managedObjectClass> Refer to Section 18.6.

<managed ObjectInstance> Refer to Section 18.6.

<eventType> This parameter should identify the <RelationshipDeletion> event type with the object identifier defined in [OMF].

<eventArgument> This parameter will include a set of <relationship attribute name, the value that was just removed from its set-value list>.

18.5.3.8 Relationship Change Report Service Agreements

This service uses the unconfirmed M-EVENT-REPORT CMIS service to convey its reporting information. It also uses the event reporting control Function specified in Section 18.5.5.1 to report the events.

CMIS M-EVENT-REPORT request parameters:

<invokeIdentifier>

<mode>

<managedObjectClass> Refer to Section 18.6.

<managed ObjectInstance> Refer to Section 18.6.

<eventType> This parameter should identify the <RelationshipChanged> Event with the object identifier defined in [OMF].

<eventArgument> This parameter will include a set of tuples of <relationship attribute name whose set-value was just replaced, the old member value which was replaced, the new replacing value>.

18.5.3.9 The usage of compound Relationship attributes 'Group' Agreements

No Relationship <group> attribute is to be used in Relationship Creation and Relationship Changing management. When a relationship <group> attribute is used in Relationship Deletion management, all relationship attribute values of the group of the selected managed object instances will be set to "null". Use of the Relationship <group> attribute is permitted for Relationship listing and Related Object Listing. Refer to [RMF] for more detail.

18.5.3.10 The usage of the combined Add/Change/Delete Services

It is possible to combine the Add, Change, Delete services in one CMIS operation, but until its complications are fully understood, it is not to be used in Phase 1.

Editor's Note: Need an example here to show the ordering operations on attributes, etc.

18.5.4 Error Reporting and Information Retrieval Function:

Editor's Note: Tutorial Material. Currently there are two services within the Error Reporting and Information Retrieval Function standard [ERIRF] that provide the ability to report errors from one open system to another system and to retrieve information from an open system. The two services are:

(1) the Error Reporting Service, and
(2) the Information Retrieval Service.

For the NMSIG Phase 1 IAs, only the Error Reporting Service of the [ERIRF] is required.

18.5.4.1 Error Reporting Service Agreements:

Editor's Note: Tutorial Material. The Structure of Management Information standard [MIM] specifies that managed objects may emit notifications. CMIS/CMIP provides the facility for reporting such notifications to a managing system. The Event Forwarding Control Function of the Management Service Control standard [MSC] provides the capability of forwarding event reports to specified destinations. This forwarding is based on information contained within the event. The Error Reporting Service defines information to be contained in the event report. This information is provided to help with understanding the cause of faults, and other information related to its side effects. This information may also be referenced within an event forwarding discriminator of the Event Forwarding Control

Function for determining if and where error reports should be sent.

The type of possible errors defined in [ERIRF] are:

(1) communication failure: errors associated with the process of sending information from one system to another. Some examples are: loss of signal, framing error, transmission error, and call establishment error.

(2) quality of service failure: errors associated with the degradation in the quality of performing a specific service by a service provider to a service user. Some examples are: response time excessive, queue size exceeded, bandwidth reduced, and retransmission rate excessive.

(3) processing failure: errors associated with processing input to produce the desired output. This is related to a software fault. Some examples are: storage capacity problem, version mismatch, corrupted data, CPU cycle limit exceeded, software error, and out of memory error.

(4) equipment failure: errors associated with equipment fault. Some examples are: power problem, timing problem, trunk card problem, line card problem, processor problem, terminal problem, external device problem, dataset problem, and multiplexer problem.

(5) environmental failure: errors associated with a condition relating to an enclosure in which the communications equipment resides. The errors may affect the performance of the equipment. Some examples are: smoke detection, enclosure door is open, high/low ambient temperature, high/low

humidity, and intrusion is detected.

Editor's Note: The above description is very general. We need contributions to further define the ProbableCauseCode. If we follow the standard, we may bite off having to explain how to categorize every error type, when to use each, when not to use each, what precedence order should be employed, etc. This is not a small task.

The following sections specify the Model, the Support Managed Object and the Error Reporting Service for the Phase 1 IAs.

18.5.4.1.1 Error Reporting Model Agreements:

For the Error Reporting Service, the Event Reporting Control Model [Section 18.5.5.1.1] applies.

18.5.4.1.2 Support Managed Object Agreements:

The Event Forwarding Discriminator object is defined in [DSO].

18.5.4.1.3 Error Reporting Service Agreements:

The following agreements and clarifications pertinent to Section 8.1 of the base standard [ERIRF] and regarding the semantics of the unconfirmed CMIS M-Event-Report service (Section 8.2.1 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-EVENT-REPORT request parameters:

<invokeIdentifier> This parameter specifies the M-Event-Report operation invocation identifier, it is to be used to distinguish this operation from others.

<mode> This parameter is set to <unconfirmed>.

<managedObjectClass> This parameter specifies the managed object class of the managed object instance which is reporting an error(s).

<managedObjectInstance> This parameter specifies the instance of the managed object that is reporting an error(s).

<eventType> This parameter specifies the type of error being reported. The five possible types are:

- Communication Error
- Quality of Service Error
- Processing Error
- Equipment Error
- Environment Error

The values for the error type are defined in [ERIRF].

<eventTime> This parameter specifies the time the error(s) occurred. Reference Section 18.6.2.3 for further IAs.

<eventArgument> For the network management Phase 1 IAs, this parameter is optional. The fields within the parameter are also optional, except where defined by the managed object class definition [MIL] or specified in the [ERIRF], [DMO] or [DMA] standards. The parameter is present if at least one of the fields below is present. The possible fields are:

<ProbableCauseCode>,
<Severity>,
<TrendIndication>,
<Backupstatus>,
<DiagnosticInfo>,
<ThresholdInfo>,
<StateChange>,
<ProposedRepairAction>,
and <OtherInformation>.

<ProbableCauseCode>

This field contains the most probable reason for the error indicated in the eventType.

<Severity>

This field contains the level of network degradation caused by the named error. Five levels of severity are defined by the base standard; they are: critical, major, minor, warning, and indeterminate. The values for the Severity code are defined in Annex A of [DMA].

<TrendIndication>

This field contains the current trend in the type of error being reported. There are two values for this attribute: TRUE, implies increase in severity, FALSE, implies decrease in severity, as defined in Annex A of [DMA].

<BackupStatus>

This field contains a value which indicates whether the failed object has been backed up or not. There are two possible values for this field: TRUE, implies backed up, and FALSE, implies not backed up, as defined in Annex A of [DMA].

<DiagnosticInfo>

This field contains information which may assist to diagnose the fault.

Editor's Note: Tutorial Material. Examples of such information may include counter values, threshold values, and configuration state, etc. as defined by managed object class.

<ThresholdInfo>

This field contains the values of the threshold which caused the error to be generated. The subfields are defined in [DMA].

<StateChange>

This field contains information, defined in Annex A of [DMA], about the administrative and operational state of the managed object at the time the error occurred.

<ProposedRepairAction>

This field contains information which may propose action to correct the fault.

Editor's Note: Tutorial Material. This information is defined by the managed object class.

<OtherInformation>
This field contains other relevant information about the managed object at the time the error occurred.

Editor's Note: Tutorial Material. This information is defined by the managed object.

18.5.4.2 Information Retrieval Function Agreements:

18.5.4.2.1 Information Retrieval Service Agreements:

18.5.5 Management Service Control Functions Agreements:

Editor's Note: Tutorial Material. There are two control functions in this category to provide the ability to specify criteria under which event operations can be controlled. The two functions are:

(1) Event Reporting Control Function, and
(2) Service Access Control Function.

The NMSIG Phase 1 network management agreements support only the Event Reporting Control Function. The Service Access Control Function is for further study.

18.5.5.1 Event Reporting Control Function Agreements:

Editor's Note: Tutorial Material. The Event Reporting Control function provides services by which event reporting can be distributed and controlled. Event report distribution means the selection of chosen events to be reported to some designated system(s) or process(es) within some selected time period. These selections are done by a filtering process using the "DiscriminatorConstruct" attribute of the "Event Forwarding Discriminator" object. Event Reporting Control is the ability to initiate, terminate, suspend, or resume event reporting through the manipulation of an Event Forwarding Discriminator object specified in Section 18.5.5.1.1. In addition, Event Reporting Control can further alter event report distribution behavior by changing the distribution attributes in an Event Forwarding Discriminator object (DiscriminatorConstruct, BeginTime and EndTime etc...).

The following sections contain the NMSIG Phase 1 network management agreements pertaining to the Event Reporting Control Model [RMF], the Support Managed Object to facilitate the Event Reporting Control Function [RMF], and the following services (defined in [RMF]):

- o Initiate event reporting service
- o Terminate event reporting service
- o Suspend event reporting service
- o Resume event reporting service
- o Modify event forwarding discriminator attributes service
- o Retrieve event forwarding discriminator attributes service.

18.5.5.1.1 Event Reporting Control Model Agreements:

The Event Reporting Control function is based on the following assumptions, pictured below:

(1) There is (at least) one managed object capable of generating notifications.

(2) There exists a conceptual event detection and processing function which receives the local notifications and forms potential event reports.

(3) There exist Event Forwarding Discriminator objects which are used for determining whether potential event reports can become real event reports which are then emitted from the open system.

(4) There exists a conceptual process which guides all potential event reports to all Event Forwarding Discriminators for evaluation.

(5) There exists a conceptual process which evaluates the potential event reports using the Event Forwarding Discriminator attributes (DiscriminatorConstruct, BeginTime, EndTime, Destination ...) to determine whether the

potential event reports are to be reported to the specified destination system(s).

```
                              Event Forwarding Discriminator
                                  Control Functions
          +--------------+    (Initiate, Terminate, Suspend,
  +---------------+ M.O. |     Resume, Update, etc...)
  | Managed Object|------+                |  |
  +---------------+  |                    |  |
         |           |                    |  |
         | Notifications                  |  |
         |           |                    |  |
         |           |                    |  |
+--------|-----------|--------------------|  |---------+
| Agent  v           v                    v  v         |
|     +----------------+            +--------------+   |
|     | Event Detection|----------> | Event Fwding |-------->
|     |       and      | potential  | Discriminator|   |Event
|     | Processing     | Events     | Processing   |   |Reports
|     |                |----------> |              |-------->
|     +----------------+            +--------------+   |
|                                                      |
+------------------------------------------------------+
```

18.5.5.1.2 Support Managed Object - Event Forwarding Discriminator Agreements

Editor's Note: Tutorial Material. The Event Report Discriminator is a management service control discriminator which is a managed object providing for specification of criteria relevant to selecting events of interest to be reported to other open systems. The criteria must be satisfied by potential event reports related to managed objects before the event report is forwarded to a particular destination. That destination is also specified by the discriminator and is the address of a remote managing process.

Editor's Note: Tutorial Material. The Event Forwarding Discriminator has the following attributes:

(1) DiscriminatorID: This attribute uniquely identifies the discriminator.

(2) DiscriminatorConstruct: This attribute specifies the conditions

which define when an event report should be generated after a event occurs. Each event which occurs in an event generating system has to be evaluated for passing the filter construct. Only those events that pass (match) the filter will result in an event report being sent to the destination system(s).

(3) ManagementUserIdentification: This attribute identifies the systems on whose behalf the event report is performed. This usually indicates the managing system.

Editor's Note: Should the Phase 1 network management IA's limit this to containing only a single system at a time? This would mean we would not require use of PDAD2 for CMIS/P.

(4) Discriminator State: This attribute specifies the state in which the Event Report Discriminator object is to be created. The Discriminator object may be created in a "locked" or "unlocked" state.

(5) Begin Time: This attribute identifies the beginning time of a 24 hour interval during which the event report service is active.

(6) End Time: This attribute identifies the ending time of a 24 hour interval during which the event report service is available.

An example: If Begin Time = 8:00 AM and End Time = 5 PM, then event reports will only be sent between the hours of 8:00 AM through 5:00 PM on the basis of this discriminator.

In Phase 1, one Event Forwarding Discriminator is defined for each destination process to which the event reports are to be sent.

18.5.5.1.3 Initiate Event Reporting Service Agreements:

Note to the Editor: Tutorial material in all subsequent sections needs to be scanned for scenario information and that material should be provided to the scenario section editor.

Editor's Note: Tutorial Material. A user at a managing system may desire that particular events generated at an event generating system be reported to a destination system. To achieve this, the user, from the managing system, will need to create Event Forwarding Discriminator objects for those particular events with the proper parameters at the event generating system.

Each Event Forwarding Discriminator object must include a DiscriminatorConstruct which specifies the desired filtering conditions under which the designated event should be reported to the destination system.

A managing system must issue a single M-CREATE CMIS service request to an event generating system to create a single Event Forwarding Discriminator. Multiple discriminators require multiple M-CREATE CMIS service requests.

Editor's Note: Once the Event Forwarding Discriminator object is created, is there an implicit assumption that the newly created object forms part of the context implied by the current association context? Can the Event Forwarding discriminator object be managed by applications using other associations other than the one over which the CMIS M-CREATE request was issued, or do they need to reassociate? This issue will be determined during the association policy discussions.

The following agreements and clarifications pertinent to Section 8.1 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-CREATE service (Section 8.3.4 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-CREATE request parameters:

<invokeIdentifier>

<managedObjectClass> The parameter value will always be the <Event Forwarding Discriminator> class. This parameter must be included in the request.

<managedObjectInstance>

(1) If this parameter is used in the request, it will identify the instance of the discriminator class by providing the DiscriminatorID and names of any superiors.

(2) Otherwise, the performing CMISE-service-user will assign a value to identify the instance.

Editor's Note: Should we agree on using (1) always in the request?

Note to the Editor: Incorporate comments from the Object Creation section, later on.

<accessControl> Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<referenceObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to this parameter.

<attributeList> This field refers to the Event Forwarding Discriminator object attributes (Section

18.5.5.1.2 of this chapter). Any attributes provided by the CMIS-service-user will be used to initialise the corresponding attributes for the newly created instance.

The <discriminatorState> attribute is set to "unlocked" by default.

CMIS M-CREATE response parameters:

<invokeIdentifier>	
<managedObjectClass>	Same as request
<managedObjectInstance>	This parameter is always returned by the response to indicate the instance name of the newly created object.
<attributeList>	This parameter specifies ALL the object attributes and values for the NEWLY created Event Forwarding Discriminator.
<currentTime>	Refer to Section 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to parameter.

18.5.5.1.4 Terminate Event Reporting Service Agreements:

Editor's Note: Tutorial Material. A user in a managing system can use this service to turn off the reporting of events from a specific event generating system.

To achieve that, the user will need to delete the Event Forwarding discriminator object(s) of the unwanted event(s) on the system. The absence of such a discriminator will not stop the generation of potential event reports caused by the managed object, it simply disables event reporting of the particular potential events from the event generating system.

A managing system must issue a single M-DELETE CMIS service request to the event generating system to delete exactly one Event Forwarding Discriminator. Multiple M-DELETE CMIS service requests are needed to delete multiple discriminators.

The following agreements and clarifications pertinent to Section 8.2 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-DELETE service (Section 8.3.5 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-DELETE request parameters:

<invokeIdentifier>

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <BestEffort> is required.

<scope>

<filter>

CMIS M-DELETE response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Section 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

18.5.5.1.5 Suspend Event Reporting Service Agreements:

Editor's Note: Tutorial Material. This service temporarily stops event reports from being sent from the event generating system to the destination system, yet retains the ability to resume the reporting if desired.

To suspend event reporting, a managing system must issue an M-SET CMIS service request to the event generating system to change the value of the <DiscriminatorState> attribute to "locked".

When the <DiscriminatorState> attribute is "locked", any events that would normally occur for this discriminator are discarded and NOT queued up for later transmission.

The following agreements and clarifications pertinent to Section 8.3 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-SET service (Section 8.3.2 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-SET request parameters:

<invokeIdentifier>

<mode> This parameter will be set to <confirmed>.

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Section 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeList> This parameter will include the Event Forwarding Discriminator attribute <discriminatorState> with the value of the attribute to be "locked". (See Section 18.5.5.1.2 of this chapter)

CMIS M-SET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<currentTime> Refer to Section 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

18.5.5.1.6 Resume Event Reporting Service Agreements:

Editor's Note: Tutorial Material. This service enables event reporting for particular types of events, thereby permitting events to be sent from a specific event generating system to a specific destination system. This operation is used to resume the reporting of events that was previously suspended.

To resume event reporting, the managing system must issue an M-SET CMIS service request to an event generating system to change the <discriminatorState> attribute to <Unlocked>.

The following agreements and clarifications pertinent to Section 8.4 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-SET service

(Section 8.3.2 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory and are as specified in Section 18.5.5.1.5, with the following difference:

<attributeList>	This parameter will contain the Event Forwarding Discriminator attribute <discriminatorState>. (See Section 18.5.5.1.2 of this chapter). The value of the attribute will be set to "unlocked".

18.5.5.1.7 Modify Event Forwarding Discriminator Attributes Service Agreements:

Editor's Note: Tutorial Material. A managing system can change the conditions of event reporting for some selected events by changing the values of the Event Forwarding Discriminator attributes which are used in the processing associated with event distribution and control. For example, the user may want to move/modify the reporting of a specific type of event to a different destination system, or change the frequency of the event reporting. To achieve such results, a managing system will need to modify the value of the <managementUserIdentification> and/or <DiscriminatorConstruct> attributes to reflect the new needs. This service can be used for locked or unlocked Event Forwarding Discriminator objects.

To change attributes of one specific Event Forwarding Discriminator in one specific event generating system, a managing system must issue a single M-SET CMIS service request to the event generating system. Changes to multiple discriminators in a single event generating system require multiple M-SET CMIS service requests.

The following agreements and clarifications pertinent to Section 8.5 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-SET service (Section 8.3.2 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-SET request parameters:

<invokeIdentifier>

<mode> This parameter will be set to <confirmed>.

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeList> This parameter will specify the Event Forwarding Discriminator attributes to be modified. The modifiable attributes are:
<DiscriminatorConstruct>,
<Management User Identification>,
<Discriminator State>,
<Begin Time>, <End Time>.

Editor's note: This parameter is going to be replaced by the <modificationList> parameter, once PDAD2 for CMIS/P is adopted.

CMIS M-SET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<attributeList> This parameter will specify the Event Forwarding Discriminator attributes that were modified.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

18.5.5.1.8 Retrieve Event Forwarding Discriminator Attributes Service Agreements:

To examine the Event Reporting Discriminator parameters associated with a specific event, a managing system must issue an M-GET CMIS service request to an event generating system to retrieve the values of specific discriminator attributes.

The following agreements and clarifications pertinent to Section 8.5 of the base standard [MSC] and regarding the semantics of the confirmed CMIS M-GET service (Section 8.3.1 of [CMIS]) are supported by the Phase 1 network management IAs. All CMIS parameters are mandatory, except where noted below.

CMIS M-GET request parameters:

<invokeIdentifier>

<baseManagedObjectClass>

<baseManagedObjectInstance>

<accessControl> Refer to Sections 18.6.2.4 and 18.6.3.1.2 (Management Communications) of this chapter for agreements pertaining to this parameter.

<synchronization> <bestEffort> is required.

<scope>

<filter>

<attributeIdList> This parameter will specify the Event Forwarding Discriminator attributes to be retrieved. The readable

attributes are:
<DiscriminatorId>,
<DiscriminatorConstruct>,
<Management User Identification>,
<Discriminator State>,
<Begin Time>, <End Time>.

Default gets all attributes.

CMIS M-GET response parameters:

<invokeIdentifier>

<linkedIdentifier>

<managedObjectClass>
<managedObjectInstance> Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.

<attributeList> This parameter will specify the retrieved Event Forwarding Discriminator attributes.

<currentTime> Refer to Sections 18.6.2.3 and 18.6.3.1.3 (Management Communications) of this chapter for agreements pertaining to this parameter.

18.5.5.2 Service Access Control Function Agreements:

Editor's Note: This section is for future study.

18.5.6 Event Logging Control Function Agreements:

18.5.6.1 Event Logging Model Agreements:

18.5.6.2 Support Managed Object Agreements:

18.5.6.2.1 Log Discriminator Agreements:

18.5.6.2.2 LOG Agreements:

18.5.6.3 Log Control Services Agreements:

18.5.6.3.1 Initiate Event Logging Service Agreements:

18.5.6.3.2 Terminate Event Logging Service Agreements:

18.5.6.3.3 Suspend Event Logging Service Agreements:

18.5.6.3.4 Resume Event Logging Service Agreements:

18.5.6.3.5 Modify Event Logging Parameters Service Agreements:

18.5.6.3.6 Event Log Parameters Retrieval Service Agreements:

18.6 MANAGEMENT COMMUNICATIONS

This section identifies, in detail, use of the management communications services and protocols, based on the standards defined in [CMIS], [CMIP], [ADDRMVS/P] and [CANGETS/P].

This section covers the agreements pertaining to the use of associations over which to carry management PDUs, agreements pertaining to the services offered to a CMIS Service User (in terms of the functions defined in Section 18.5), agreements pertaining to the protocol used to convey the management PDUs, and agreements pertaining to the services required of other layers in order to convey the management PDUs defined.

Editor's note: Tutorial Material: This draft of the Association Policy section of the Phase 1 IAs represents the output from an interim NMSIG meeting held in Peabody MA in November 1989. The purpose of the meeting was to align the draft section from the July Workshop with output from the Florence meetings and with the issues from the NMSIG Issue Log. As a result of review by the December

1989 OIW NMSIG Meeting, some additional changes were made to this text.

The participants at the interim meeting summarized the issues into 8 items. These are listed here to enable reviewers to understand the premise for the subsequent text.

Issue 1: Should there be agreements about arbitration among competing requests where agents allow multiple associations to managing systems?

> It was decided that this was really a matter for an agent implementation. If an agent does some form of arbitration (eg; temporarily lock out a request to modify an object while a prior request is being honored), it must indicate this in some agreed upon way so that the managing system can distinguish between this situation and some other error, such as access denied or no such object.
>
> This issue is not related to association types or to access control. The recommendation was placed in an appropriate section of the CMIS/P agreements in 18.6.3.

Issue 2: What is the retry policy, if any?

> It was proposed that we make some suggestions and have them reviewed by the Workshop. See section 18.6.1.4.

Issue 3: What are the connect and disconnect policies, if any?

> See section 18.6.1.4.

Issue 4: How are the roles of managing and managed system determined?

> It was felt that it was necessary to determine which role a system is playing on an association and that the Application Context Name work in the Arhus output for SMO fit the bill. See section 18.6.1.2.

Issue 5: Handling of events vs command/control.

Issue 6: Handling of monitoring vs control.

> Re Issues 5 and 6, it was felt that managed and managing systems may wish to restrict the types of

functions that may be performed on a particular association. The proposal for addressing this issue is in section 18.6.1.2.

Issue 7: Views of a MIB on an association.

It was decided to keep the output from the July workshop which states that we make no agreements regarding the scope of an association as it applies to the objects made accessible over that association. The arbitration process adds a slight wrinkle though. See section 18.6.1.4.

Issue 8: Are we making recommendations or requirements?

This draft has both. It was never really decided if recommendations are appropriate in these agreements. If they aren't then we will have to decide whether to drop the recommendations in this draft or make them requirements.

18.6.1 Association Policies

Associations are established using the procedures described in [ACSEP] with recommendations and extensions described in these implementation agreements.

Phase 1 IAs specify the different types of associations that may be established between managing and managed systems (see 18.6.1.2). The type of a given association is determined by the exchange of appropriate application context information between the systems using a negotiation process.

Phase 1 IAs recommend that managed systems reserve resources for at least one association for event reporting (see 18.6.1.3).

Phase 1 IAs require the use of A-RELEASE instead of A-ABORT. Phase 1 IAs also make recommendations regarding parameters affecting the scope of managed objects and span of time for an association and synchronization among multiple associations (see 18.6.1.4).

Phase 1 IAs specify the access control information to be used in the establishment of an association (see 18.6.1.5).

18.6.1.1 ACSE Services

The A-ASSOCIATE and A-RELEASE ACSE services are used as specified in [ACSE]. The Phase 1 IAs make certain requirements as to the use of the APDU fields noted below. Usage of all other fields is left to the implementor.

AE-TITLE (Calling AP Title and Calling AE Qualifier) usage is specified in 18.6.1.2.

Application Context Name usage is specified in 18.6.1.2.

ACSE User Information consists of three parameters (specified in [CMIS]): Functional Units, Access Control and CMIS User Information. Refer to section 18.6.3 for agreements relating to Functional Units. Refer to section 18.6.1.5 for agreements relating to Access Control. The Phase 1 IAs make no agreements relating to CMIS User Information.

18.6.1.2 Association Types

The Phase 1 IAs specify that four types of association may be negotiated between managing and managed systems. These types are:

Event — M-EVENT-REPORTs may be sent by the managed system; no other CMIP PDUs are allowed

Event/Monitor — same as Event type except that, in addition, the managing system may also issue M-GET requests and receive M-GET responses over the association

Monitor/Control — managing system may issue M-GET, M-SET, M-CREATE, M-DELETE and M-ACTION requests over the association; no event reporting is allowed

Full Mgr/Agent — all functions must be supported

The negotiation process specified for the Phase 1 IAs uses the A-ASSOCIATE and A-RELEASE services as specified in [ACSEP]. Application Context Name [SMO] is used to determine the requestor's "role" in an association (managing or managed system) and to determine the type of the association. The following negotiation rules are specified by the Phase 1 IAs:

Editor's Note: The SIG left open the question of using Application Context Names for both role and type determination. The editor investigated further to find out if there were any restrictions that would prevent such usage. Having found no restrictions, the editor updated the text to provide more detail in this direction.

Editor's Note: We need to assign Application Context Names. I suggest that we register appropriate object names under the NMSIG arc. I'll take a stab at the proper format (see RASIG output...Section 6 of the Working Document) and propose some names as a placeholder until we determine the actual format/names. (Wordsmithing and format advice are welcome.)

```
{iso(1) identified-organization(3) oiw(14)
     nmsig(2)
     manager-event-association(x)}

{iso(1) identified-organization(3) oiw(14)
     nmsig(2)
     manager-event-monitor-association(x)}

{iso(1) identified-organization(3) oiw(14)
     nmsig(2)
     manager-monitor-control-association(x)}

{iso(1) identified-organization(3) oiw(14)
      nmsig(2) manager-full-association(x)}

{iso(1) identified-organization(3) oiw(14)
      nmsig(2) agent-event-association(x)}
```

Editor's Note: Tutorial Material: Ref: [SMO] Annex A

The Application Context Name (ACN) indicates the role of the initiator of an association. The responder may alter the type indication to request a change in the type. Note that the proposed ACNs above follow the agreements on which system may request a particular type of association. Thus there is a single agent initiated ACN since agents (managed systems) may only initiate event reporting associations.

The ACNs in these agreements refine those defined in Annex A [SMO] and are used in the same fashion.

Editor's Note: We will need to add text relating to negotiation of System Management Function functional units as changes to this section as the relevant standards (10164-*) are updated. It is anticipated that the work in N740 will be used as the basis.

1. A managed system may only request an Event association and, in fact, must create an Event association if it has an event to report and no suitable association already exists.

2. Managing systems may request any association type.

3. An association is created by the requesting system issuing an A-ASSOCIATE request with the requestor's AE-TITLE and the desired application context. The responding system then returns either 1) an A-ASSOCIATE response with the requestor's AE-TITLE and the application context which it wishes to accept or 2) an A-ASSOCIATE response rejecting the association.

4. Managed systems may negotiate "downward" from Full to Monitor/Control, Event/Monitor or Event by returning the new application context in the A-ASSOCIATE response to the managing system during the association creation process. In the same fashion, managed systems may negotiate from Event/Monitor to Event.

5. When a managing system receives an application context in an A-ASSOCIATE response that differs from the context sent in an A-ASSOCIATE request it may either proceed with the new context or refuse the new context by issuing an A-RELEASE request.

Editor's Note: A-RELEASE is used when the requestor does not agree with the new context. A-ABORT is used for invalid negotiation.

Note that a system may play both managing and managed system roles, but not on the same association.

18.6.1.3 Events

Phase 1 IAs recommend that managed systems make resources available for at least one association for the purposes of event reporting. The resources allocated to an association should be re-useable. That is, if the system must report an event to multiple managers, it may have to repeatedly utilize the resources for an association to each of the managing systems. This recommendation is made to ensure that events are not lost due to a lack of associations.

Editor's Note: The status of 18.6.1.3 as a recommendation rather than a requirement is open for comments.

18.6.1.4 Scope/Span of an Association

Editor's Note: Discussions at the Florence meeting indicate the potential for an "association policy object". This object would allow for the maintenance of parameters pertaining to the behavior of an association. These parameters would include such things as number of retries and inactivity timers. This version of section 18.6 was written so that if this proposal comes to fruition, the agreements can be migrated to the ap-object by "transferring" the parameters to the object itself.

The Phase 1 IAs specify no process for negotiating the scope of an association as it pertains to the objects that may be managed within the context of that association.

Editor's Note: Text in the December 1989 Workshop draft document regarding arbitration between requests from multiple managers was moved from this section to the CMIS/P section (Section 18.6.3).

The Phase 1 IAs specify no process for negotiating a time span of an association. The managing or managed system may terminate an association based upon an implementation specific algorithm governing association durations.

Editor's Note: Text in the December 1989 Workshop draft document regarding potential parameters for managing time span and retries for associations was removed from this section.

The text has been retained "off-line" at the direction of the NMSIG.

Underlying services such as ACSE may also cause the termination of an association.

The Phase 1 IAs require that associations be terminated with A-RELEASE to avoid loss of information in an association.

Editor's Note: Tutorial Material: If A-ABORT is used to terminate an association, there exists a potential for loss of information such as pending events or confirmations. A-ABORT must be used, however, when a protocol violation occurs or where an association is not yet established.

18.6.1.5 Other Aspects of Associations

Editor's Note: The access control information in this section is based on some notes from a joint NMSIG/Security SIG meeting that took place some time ago. We should review this with the Security SIG to make sure we are still in agreement and get more information on usage and encoding. This review is tentatively planned for the March 1990 OIW.

The Phase 1 IAs specify that the following information may be used in establishing an association. A managed system, if it requires access control information, must use this format.

Unused fields must contain nulls.

Field	Name	Purpose
1	Length	length of access control data
2	Initiating Person	
3	Process Type	
4	Process ID	
5	Authorization	password
6	Access Privileges	

7	Audit Requirements	
8	Integrity Seal	universal closed community checksum; message authentication code
9	Optional Information	0-n bytes of optional data

18.6.2 Agreements on CMIS

These agreements are based on the standard defined in [CMIS].

18.6.2.1 Object Naming

Object Naming will be accomplished using Distinguished Names as specified in Section 18.7.2.

18.6.2.2 Multiple Object Selection

Editor's Note: Tutorial material: CMIS/CMIP defines the operations that may be applied to a collection of managed objects. In order to use this capability, the Functional Unit: Multiple Object Selection must have been negotiated for the association; in addition the Functional Unit: Multiple Reply must also have been negotiated for the association.

There are five aspects to Multiple Object Selection:

- Scoping, which allows the selection of one or more managed objects

- Filtering, which allows the managed object(s) defined by the scope to be further reduced by a boolean condition applied to each managed object within the defined scope, yielding a set of selected managed objects to which the operation is to be applied

- Synchronization, which defines how the operation is to be synchronized across the selected managed objects

- Linked Replies, which defines how multiple replies are to be returned for a single operation applied across the set of selected managed objects.

- Object selection across multiple associations, which defines how access to a managed object over more than one association during a given time period is to be coordinated

Multiple Object Selection applies to all management operations except Event Report and Create; however, the Phase 1 network management IAs also exclude use of Delete with Multiple Object Selection (see Section 18.6.3.2.9).

Editor's Note: The exclusion of multiple object selection with Delete is an issue.

18.6.2.2.1 Scoping

Editor's Note: Tutorial material: Scoping is used to define the scope of managed objects to which a particular management operation will apply (subject also to any filtering, as described in Section 18.6.2.2.2). For those management operations for which multiple objects may be selected, scoping is always in effect; however, by default, the Scope parameter will select only a single object (called the Base Managed Object). To select other than a single object, the Functional Unit: Multiple Object Selection bit must have been negotiated at association initialization.

Scope is always defined in terms of the containment hierarchy, and with reference to a single Base Managed Object. There are three different types of Scope permitted:

- Base Object only - this selects the one object defined by the Base Managed Object (Class and Instance), and is the default if the Scope parameter is not provided

- Nth Level Subordinates - this selects all objects in the 'N'th level down the containment tree from the Base Managed Object. Note that this is likely to include objects from different object classes - the Filter parameter (described below) may need to include the object class as a filtering criteria

- o Whole Subtree - this selects all objects, including the Base Managed Object, in the containment tree from the Base Managed Object.

Consider the following containment tree, comprising fictitious object classes System, Mux, Line, Modem and Modem Port, and each having instance names identified by a string (shown as a single character in parentheses):

```
                    System (S)
                          |
                          |
            +-------------+-------------+
            |                           |
            |                           |
          Mux (A)                     Mux (B)
            |                           |
   +--------+--------+            +-----+-----+
   |        |        |            |           |
   |        |        |            |           |
Line (X)  Line (Y)  Line (Z)    Modem (M)   Modem (N)
                                  |           |
                                  |           |
                                Modem       Modem
                               Port (P)    Port (Q)
```

If the Base Managed Object Class is System and the Base Managed Object is (S):

- o If Base Object Only is chosen, then System (S) is the selected object
- o If 1st Level Subordinate is chosen, then Mux (A) and Mux (B) are the selected objects
- o If 2nd Level Subordinate is chosen, then Line (X) Line (Y), Line (Z), Modem (M) and Modem (N) are the selected objects
- o If 3rd Level Subordinate is chosen, then Modem Port (P) and Modem Port (Q) are the selected objects
- o If 4th Level Subordinate is chosen, there are no objects that satisfy the criteria.
- o If the Whole SubTree is chosen, then System (S), Muxes (A) and (B), Lines (X), (Y) and

(Z), Modems (M) and (N) and Modem Ports (P) and (Q) are the selected objects.

These Phase 1 network management IAs define that systems need minimally support Base Object Only, and they need not support Multiple Object Selection. If a system supports Multiple Object Selection, then any of the options for the Scope parameter may be specified. However, these IAs restrict the M-DELETE operation only to permit selection of the Base Object Only - refer to Section 18.6.3.2.9.

Editor's Note: The restriction on M_DELETE is an issue.

If there are no objects that satisfy the scoping criteria, the error 'InvalidScope' is returned.

18.6.2.2.2 Filtering

Editor's Note: Tutorial material: Having selected a set of managed objects, via the Base Managed Object Class, Base Managed Object Instance and the Scope parameters, it is possible to restrict the actual set of managed objects to which the operation will be attempted to a smaller set by applying a filter, specified in the Filter parameter.

Filtering may be specified only after the Functional Unit: Multiple Object Selection has been negotiated at association initialization. Note, however, that once this capability has been negotiated, it is possible to apply a filter to a single managed object (specified by Base Object Only in the Scope parameter).

The filter condition is defined to allow very complex forms of expressions yielding a boolean result. The simplest component of a filter condition is an AttributeValueAssertion (AVA), which defines a sequence of AttributeIds and associated AttributeValues; the operator applied to each AVA can be =, >= or <=. A second filter condition is the 'presence' of an attribute indicated by an AttributeId, and the last filter condition allows string or sub-string comparisons to be performed on attributes. Filter conditions can be combined by boolean AND or OR operators (which operate on two or more filter conditions), and they can be negated by the NOT operator.

In general, a filter defines a set of assertions to be applied to the attributes of an object instance. If a filter defines an attribute value assertion for an attribute, it is only evaluated if the attribute is present in the managed object instance. If the attribute is not present, the attribute value assertion for that attribute is assigned the value FALSE.

These Phase 1 network management IAs specify that systems need not support Filtering. In this case, they do not negotiate Multiple Object Selection at association initialization. However, if they support Multiple Object Selection, then they must minimally support AND and OR with a set of two filter conditions (which must not themselves be AND or OR), and NOT. In addition, they must support the filter conditions Equality, GreaterOrEqual, LessOrEqual and Present. This means that a conforming system does not have to support compounds (AND or OR) with more than two items, and does not have to support the SubString filter condition.

If a system receives a filter parameter that it is unable to process, it shall return the error 'InvalidFilter', including the smallest portion of the CMISFilter that indicates the compound operator or filter condition that is not supported.

If, in the process of filtering from the set of selected entities, there are no managed objects selected, the error 'NoSuchObjectInstance' shall be returned.

Editor's Note: A more appropriate error, or other mechanism, will be used in place of 'NoSuchObjectInstance' when and if the standards are changed.

If a filter is applied to a single managed object (specified by Base Object Only in the Scope parameter) and the filter condition evaluates to false, the error 'NoSuchObjectInstance' will be returned.

Editor's Note: A better error or better representation of this condition (eg, the 'null return' proposed in CMIS/P ballot comments) will be used when and if the standards change.

Note that [MIM] limits the filter conditions to apply only to the selected managed object's attributes, and not to the attributes of any arbitrary containing (or otherwise) managed object.

Editor's Note: New Issue: Due to the limitations of encoding relational operators in CMIP, some unexpected behavior can result where missing attributes are involved. Consider a request by a human manager to filter from a set of managed objects based upon the number of 'errors' for each object (assuming 'error' to be an attribute defined for a number of object classes. If the condition is specified as (ERRORS > 100) by one human and (ERRORS >= 100) by another human, the results will be quite difficult. In the first case, the CMIP encoding could yield (NOT(ERRORS <= 100)), so that for an object class not supporting ERRORS, the whole expression yields TRUE, rather than FALSE, as would be the case if CMIP permitted encoding of the < and > relational operators directly.)

18.6.2.2.3 Synchronization

Editor's Note: Tutorial material: Synchronization is specified by an invoker to indicate the way in which the performer must process an operation that is to be applied to the selected managed objects (as defined by the Scope and Filter parameters). There are two choices: BestEffort (which is the default if the Synchronization parameter is omitted), whereby the performer will attempt the operation on each of the managed objects independently; and Atomic, whereby the performer must either perform the operation on all selected objects successfully or else must not perform the operation on any of the objects.

In order to support interoperability between managing systems and managed systems, these Phase 1 network management IAs define that the default synchronization (i.e., BestEffort) must be supported by all conforming systems.

If a performer is unable to comply with a synchronization request specified by an invoker, the performer shall return the error 'syncNotSupported' indicating those synchronization values that are permitted.

18.6.2.2.4 Linked Replies

Editor's Note: Tutorial material: Linked Replies are used to permit a reply to an operation to be carried in more than one distinct PDU. This capability is used, for example, to return multiple replies to a single PDU, where the operation selected multiple objects. Linked Replies may be used only when the Functional Unit: Multiple Replies has been negotiated during association initialization.

The way in which multiple linked replies are used, and the inter-relationship between the two parameters Invoke Id and Link Id is shown in the following example. Here we assume that the original request is an M-GET which selects a set of five entities (by the appropriate use of the Scope and Filter parameters). We will assume that we are in the middle of an association, where the next Invoke Id to be used by the invoker is 7, and the next Invoke Id to be used by the responder is 21. The CMIP PDUs will be as follows (see references [ROSES] and [ROSEP]):

```
M-GET Request
ROS Invoke      -------------------->
Invoke Id = 7

                                          M-LINKED-REPLY
                <--------------------     ROS Invoke
                                          Invoke Id = 21
                                          Link Id = 7

                                          M-LINKED-REPLY
                <--------------------     ROS Invoke
                                          Invoke Id = 22
                                          Link Id = 7

                                          M-LINKED-REPLY
                <--------------------     ROS Invoke
                                          Invoke Id = 23
                                          Link Id = 7

                                          M-LINKED-REPLY
                <--------------------     ROS Invoke
                                          Invoke Id = 24
                                          Link Id = 7

                                          M-GET Response
                <--------------------     Either a ROS
                                          Result or a
                                          ROS Error
                                          Invoke Id = 7
```

Note that the Link Id within each M-LINKED-REPLY contains the invoker's original Invoke Id, and each M-LINKED-REPLY has its own unique Invoke Id. The Response to the original request is contained in the last PDU which terminates the Linked Reply sequence. Note also that there is no confirmation of each M-LINKED-REPLY PDU by the M-GET invoker.

Following the above protocol exchange, the next Invoke Id to be used by the invoker will be 8, and the next Invoke Id to be used by the responder will be 25.

These Phase 1 network management IAs define that the Linked Reply capability must be provided by any system that supports the Functional Unit: Multiple Replies.

18.6.2.2.5 Object Selection Across Multiple Associations

The Phase 1 IAs specify no process for arbitrating between multiple associations which may have overlapping scopes. However, a managed system may have an internal algorithm for arbitrating between potentially conflicting requests from multiple associations (affecting the scope of the association). Such an algorithm is optional but, if implemented, the managed system returns a <requestTemporarilyDenied> error to the requesting system to indicate that the request is being denied for arbitration reasons.

Editor's Note: An object-id needs to be assigned to the above error. This needs to be documented in an appropriate section of these agreements.

ISSUE: Can we define new error codes?

18.6.2.3 Time

Editor's Note: Tutorial material: Many of the management operations allow for a current time parameter to be provided. This parameter is used to define the actual time at which the operation took place, for example when an attribute value was changed or sensed, when an object was created, or when an occurrence was detected by a managed object.

The time provided shall be as close as possible to, but not before, the actual time the operation occurred in order to provide the most accurate timestamp.

Providing this parameter on management operations allows the coordination of time between management operations and managed objects on the same open system. For example, it makes it possible to determine whether an event, indicating an abnormal condition, occurred before or after a particular management operation was executed.

Note that in the absence of mechanisms in the open systems to coordinate clocks (e.g. by the use of a standard clock source), it is not, in general, possible to define a temporal ordering for observations that are timestamped by different open systems.

Refer to Section 18.6.3.1.3 for information about how the time parameters are encoded.

(Ref issues 87/12-09 and 88/05-16)

18.6.2.4 Access Control

Editor's Note: This issue has been discussed with the Security SIG.

CMIS permits access control to be supplied, and checked, on either an association or an individual operation or both.To simplify the building of products, while still retaining essential capabilities, the Phase 1 network management IAs restrict the Access Control parameter to be permitted only in an association initialization. Use of this field in other PDUs for individual management operations is outside the scope of these IAs and conformant implementations may ignore this field.

(Ref: issues 87/12-04 and 88/06-34)

18.6.2.5 Error Handling

Editor's Note: This section needs to be written, but it is not currently clear exactly how much should be specified in this section, how much should be written about the individual error conditions for each operation listed in Section 18.6.3.2.x, and how much should be defined in Section 18.5 (Management Functions and Services).

18.6.3 Agreements on CMIP

These agreements are based on the standard defined in [CMIP]. The agreements in this section have been defined in terms of those capabilities necessary to support the functions and services defined in Section 18.5 (Management Functions and Services) and in terms of the Association Policies defined in Section 18.6.1.

18.6.3.1 General PDU Agreements

This section includes those protocol agreements that apply to a number of different CMIP PDUs.

18.6.3.1.1 Invoke Ids

Invoke IDs shall be monotonically increasing, with an increment of 1, integer values for each operation within a single association, starting at zero for the first operation across an association. Invoke IDs wrap to zero when incrementing from 2^32-1.

(Ref: issue 87/12-06)

18.6.3.1.2 Access Control

The Access Control field may be supplied on association initialization. Use of the Access Control field in other CMIP PDUs is outside the scope of these IAs and conformant implementations may ignore this field.

(Ref: issues 87/12-04 and 88/06-34)

18.6.3.1.3 Time

For the Phase 1 network management IAs, the encoding of the Current Time parameters is ASN.1 Generalised Time, UTC Type, as specified in [ASN1] Clause 30.3, b) and c), with the granularity of the time representation indicating the precision of the time measurement. For example, the string 19890613123012.333-0500 represents a local time of 12:30:12 (and 333 msecs) on 13th June 1989, in a time zone which is 5 hours behind GMT.

(Ref: issue 87/12-09)

18.6.3.2 Specific PDU Agreements

This section includes the protocol agreements that apply to each specific CMIP PDU.

18.6.3.2.1 M-Event-Report

The following agreements and clarifications, pertinent to Section 8.2.1 of the base standard [CMIS] and Section 6.3 of the base standard [CMIP] and regarding the M-EVENT-REPORT service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

Section 18.5 (Management Functions and Services) defines the various types of Event Reports that may be sent. For the Phase 1 network management agreements, only the unconfirmed mode is required.

The Event Time parameter must be set to the time that the managed object detected the condition that generated the event (or as close to, but not before, that time), rather than the time at which the M-EVENT-REPORT itself is sent.

All arguments defined for the particular event type of the managed object class (see Section 18.7, Management Information Agreements) for the M-EVENT-REPORT must be supplied in the Event Argument parameter.

M-EVENT-REPORT Request Parameters:

<Invoke Identifier> (See Section 18.6.3.1.1)
<Mode> Must be set to Unconfirmed.

<Managed Object Class>

<Managed Object Instance>

<Event Type>

<Event Time> Must be supplied - indicates the time that the managed object detected the even (See Section 18.6.3.1.3)

<Event Argument> See above.

M-EVENT-REPORT Response Parameters:

To date, no events have been defined which require the confirmed mode of the Event Report. Hence, there are no agreements pertinent to the event response parameters listed below.

<Invoke Identifier>

<Managed Object Class>

<Managed Object Instance>

<Event Type>

<Current Time>

<Event Result>

<Errors>

18.6.3.2.2 M-Get

The following agreements and clarifications, pertinent to Section 8.3.1 of the base standard [CMIS] and Section 6.4 of the base standard [CMIP] and regarding the M-GET service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

For a successful M-GET operation, the performer shall return (in the Attribute List parameter) either the attribute values for all attributes explicitly requested (in the Attribute Identifier List parameter), or the attribute values for all attributes defined for the managed object(s) selected (if the Attribute Identifier List is omitted).

For a partially successful M-GET operation, where only some attribute values were retrieved, the performer shall return (in the Errors parameter, specifically encoded as GetListError) all attribute ids and their corresponding values that were successfully retrieved from the set of attributes selected as described above, together with all attribute ids, and the corresponding error codes, for each of the attributes for which errors were detected. The invoker can assume that there was no attempt to retrieve attributes whose ids were not returned in a GetListError.

M-GET Request Parameters:

<Invoke Identifier>	(See Section 18.6.3.1.1)
<Base Object Class>	
<Base Object Instance>	
<Scope>	
<Filter>	
<Access Control>	This field need not be supplied (See Section 18.6.3.1.2)
<Synchronization>	This field may be omitted. If present, this field must have

the value of BestEffort (see Section 18.6.2.2.3)

<Attribute Identifier List>

M-GET Response Parameters:

<Invoke Identifier>

<Linked Identifier>

<Managed Object Class> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Managed Object Instance> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Current Time> This field must be supplied, and indicates the time at which the attribute values were read at the managed object.(See Section 18.6.3.1.3)

<Attribute List>

<Errors>

For the final reply of a series of linked relies, the GetResult is omitted. Hence Managed Object Class, Managed Object Instance, Current Time, Attribute List and Errors are all omitted in this one case.

18.6.3.2.3 M-Set

The following agreements and clarifications, pertinent to Section 8.3.2 of the base standard [CMIS] and Section 6.5 of the base standard [CMIP] and regarding the M-SET service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

All M-SET operations shall be confirmed, to ensure that the invoker knows the outcome of any request to change values of attributes.

For a successful M-SET operation, the performer shall return (in the Attribute List parameter) the attribute values for all attributes explicitly specified (in the Attribute List parameter) indicating their new values.

For a partially successful M-SET operation, where only some attribute values were modified, the performer shall return (in the Errors parameter, specifically encoded as SetListError) all attribute ids and their corresponding values that were successfully modified from the set of attributes ids and values supplied, and all attribute ids and the corresponding error codes for each of the attributes for which errors were detected. The invoker can assume that there was no attempt to modify attributes whose ids were not returned in a SetListError.

When multiple objects are selected for an M-SET operation, there is no ordering implied between selected objects. If the ordering is important, the requesting system may use separate operations, for individual object instances, in the desired order.

M-SET Request Parameters:

<Invoke Identifier>	(See Section 18.6.3.1.1)
<Mode>	Must be set to confirmed.
<Base Object Class>	
<Base Object Instance>	
<Scope>	
<Filter>	
<Access Control>	This field need not be supplied (See Section 18.6.3.1.2)
<Synchronization>	This field may be omitted. If present, this field must have the value of BestEffort (see Section 18.6.2.2.3)
<Attribute List>	

M-SET Response Parameters:

<Invoke Identifier>

<Linked Identifier>

<Managed Object Class> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Managed Object Instance> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Attribute List>

<Current Time> This parameter must be supplied, and indicates the time at which the attribute values were set (or were attempted to be set) at the managed object. (See Section 18.6.3.1.3)

<Errors>

18.6.3.2.3.1 Add, Remove and Set to Default

DAD2 to both CMIS and CMIP ([ADDRMVS] and [ADDRMVP]) proposes a scheme whereby M-SET is augmented to permit values to be added to a multi-valued attribute, values to be removed from a multi-valued attribute, and for an attribute to be set to its default value without the default being sent as an explicit value in the protocol.

Section 18.5 (Management Functions and Services) makes use of these capabilities, so this sub-section indicates how those services are to be used.

Where multi-valued attributes are involved in an M-SET operation, the values returned after any modification operation on them shall be the full set of values of that attribute, and not just the values that were modified (e.g., added or removed).

M-SET Request (DAD2) Parameters:

<Modification List>

M-SET Response (DAD2) Parameters:

<Attribute List>

18.6.3.2.4 M-Action

The following agreements and clarifications, pertinent to Section 8.3.3 of the base standard [CMIS] and Section 6.6 of the base standard [CMIP] and regarding the M-ACTION service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

All M-ACTION operations shall be confirmed, to ensure that the invoking system is aware of the outcome of every requested operation.

When multiple objects are selected for an M-ACTION operation, there is no ordering implied between selected objects. If the ordering is important, the requesting system may use separate operations, for individual object instances, in the desired order.

M-ACTION Request Parameters:

<Invoke Identifier> (See Section 18.6.3.1.1)

<Mode> Must be set to Confirmed.

<Base Object Class>

<Base Object Instance>

<Scope>

<Filter>

<Managed Object Class>

<Access Control> This field need not be supplied (See Section 18.6.3.1.2)

<Synchronization> This field may be omitted. If present, this field must have the value of BestEffort (see Section 18.6.2.2.3)

<Action Type>

<Action Argument>

M-ACTION Response Parameters:

<Invoke Identifier>

<Linked Identifier>

<Managed Object Class> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Managed Object Instance> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Action Type> This parameter must be supplied on all responses.

<Current Time> This parameter must be suppliedand indicates the time at which the managed object performed (or attempted to perform) the action requested. (See Section 18.6.3.1.3)

<Action Result>

<Errors>

18.6.3.2.5 M-Create

The following agreements and clarifications, pertinent to Section 8.3.4 of the base standard [CMIS] and Section 6.7 of the base standard [CMIP] and regarding the M-CREATE service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

The Managed Object Instance request parameter may be present or absent depending on whether the invoker supplies the instance name or the performer assigns the instance name automatically. The definition of each Managed Object Class shall define whether the instance name must be supplied by the invoker, or must be assigned by the performer. This definition shall apply to every management-initiated creation of instances of that managed object class.

The values of each of the attributes of the newly created object are derived in the following order, where each bullet may overide a value provided in a previous bullet:

- From the default value defined for the attribute in the managed object class definition, if any
- From the corresponding value, if any, derived from the reference object, if provided
- From the value provided in the Attribute List request parameter.

If none of these methods provides a value for any attribute, then the operation shall be considered to have failed, i.e., no new instance is created, and the error code Invalid Attribute Value shall be returned.

M-CREATE Request Parameters:

<Invoke Identifier> (See Section 18.6.3.1.1)

<Managed Object Class>

<Managed Object Instance> See description above.

<Access Control> This field need not be supplied (See Section 18.6.3.1.2)

<Reference Object Instance>

<Attribute List>

M-CREATE Response Parameters:

<Invoke Identifier>

<Managed Object Class> This parameter must always be returned.

<Managed Object Instance> This parameter must always be returned, whether or not the instance name is supplied or provided automatically.

<Attribute List> This parameter must always be returned, and contains the list of all attribute values for the newly created object.

<Current Time> This parameter must be supplied, and indicates the time at which the particular instance of the newly created managed object came into existence. (See Section 18.6.3.1.3)

<Errors>

18.6.3.2.6 M-Delete

The following agreements and clarifications, pertinent to Section 8.3.5 of the base standard [CMIS] and Section 6.8 of the base standard [CMIP] and regarding the M-DELETE service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

In order to avoid unanticipated side-effects, this service shall be used only where the scope parameter is set to 'base object only' - thus this operation may be used only to delete a single managed object. Of course, it is a straightforward programming exercise to delete multiple objects, and the intent is to avoid unintentional deletion of large numbers of objects. Any attempt to delete more than one object via a single operation shall fail, and the error 'Invalid Scope' shall be returned.

If the managed object to be deleted has contained objects, then the operation shall fail, and the error 'Access Denied' shall be returned (in the absence of a better error).

(Ref issue on <n>-level delete)

M-DELETE Request Parameters:

<Invoke Identifier> (See Section 18.6.3.1.1)

<Base Object Class>
<Base Object Instance>

<Scope> Must be set to Base Object Only.

<Filter> Must not be specified since only one object may be deleted.

<Access Control> This field need not be supplied (See Section 18.6.3.1.2)

<Synchronization> Must not be specified since only one object may be deleted.

M-DELETE Response Parameters:

<Invoke Identifier>

<Linked Identifier>

<Managed Object Class> This parameter must be supplied on all responses, even those that reference just the base managed object.

<Managed Object Instance> This parameter must be supplied on all response, even those that reference just the base managed object.

<Current Time> This parameter must be supplied,and indicates the time at which the managed object ceased to exist. (See Section 18.6.3.1.3)

<Errors>

18.6.4 Services Required by CMIP

Editor's Note: This section is to be provided.

18.7 MANAGEMENT INFORMATION

This section, which is based on ISO standards' documents [MIM] and [GDMO], deals with basic concepts and modelling techniques related to management information. It discusses (i) the information model (Section 18.7.1), (ii) principles for naming managed objects and their attributes (Section 18.7.2), and (iii) guidelines for defining management information (Section 18.7.3). It is not within the scope of this section to define specific elements of management information - such definitions can be obtained via the Management Information

Library (MIL) produced by the OSI MIB Working Group (a subgroup of the NMSIG).

Editor's Note: Tutorial Material: Management information comprises all information in the network that is of interest to network management. A computer node in a network, a transport connection, an event log are all examples of network resources for which management information can be defined. Management information is collectively referred to as the MIB or Management Information Base.

18.7.1 The Information Model

This subsection contains agreements related to the information model as specified in Clause 5 of [MIM].

Editor's Note: Tutorial Material: Management information is modelled using object-oriented techniques. All "things" in the network that are to be managed, are represented in terms of managed objects. A managed object is an abstraction (or a logical view) of a "manageable" physical or logical network resource. "Manageable", in this context, means that the particular resource can be managed by using OSI Management Services and Protocols. Examples of managed objects include protocol layer entities, modems, connections, etc.

Each managed object belongs to a particular object class. An object class represents a collection of managed objects with the same, or similar properties. Each object class has a pre-defined identifier assigned to it by a standards' registration authority. A particular managed object existing in a particular network can be regarded as an instance of the object class to which it belongs. Thus, an object instance represents an actual realisation of an object class. A managed object is identified by specifying its object class and object instance.

Managed objects contain properties which are referred to as attributes.

Managed objects participate in relationships with each other. The relationships that are of particular concern to the Management Information Model are a) the containment relationship, and b) the inheritance relationship. These relationships are used to construct management information hierarchies, as described below. Managed objects do participate in relationships other than the two mentioned above; e.g. the Service relationship, where a managed object uses the services provided by another managed object, as in the case of a Transport Layer object using the services provided by a Network Layer object. These relationships, however, are not particularly significant for the Information Model. They can be easily represented as either managed objects or attributes, contained within the managed objects participating in the relationship.

MANAGEMENT INFORMATION HIERARCHIES

The following Management Information Hierarchies are identified:

THE CONTAINMENT HIERARCHY

This hierarchy is constructed by applying the relationship "is contained in" to objects and attributes. Objects of one class may contain objects of the same or different class. Attributes are contained within objects at any level of the containment hierarchy. Attributes cannot contain objects or other attributes. All object classes must have at least one possible superior in the containment tree. The definition of a class may permit it to have more than one such superior. However, individual instances of such a class are nevertheless contained in only one instance of a possible containing class. A special object called "root" is the ultimate superior in the containment hierarchy.

The containment hierarchy is important because it is used for naming object instances. It also defines an existence dependency among its components; i.e. an object or attribute can 'exist' only if the containing object also 'exists'. If an object contains other objects, it cannot be deleted until the contained objects have been deleted. The contained objects may be deleted automatically, if this is specified in the definition of the managed object class(es) of the contained objects.

THE INHERITANCE OR OBJECT CLASS HIERARCHY

This hierarchy is constructed by applying the relationship "inherits properties of" to object classes. An object class may inherit properties of another object class, with refinement obtained by adding additional properties. The inheriting class is called the subclass in this relationship, and the parent the superclass. For example, the class "Network Entity" may be a subclass of "Layer Entity" and a superclass of "X.25 Network Entity". Each class may have zero, one or more subclasses. Subclasses may in turn have furthur subclasses, to any degree. A special object called "top" is the ultimate superclass.

The inheritance hierarchy is useful in that it leads to a manageable and extensible technique for the definition of object classes. The inheritance hierarchy has NO relevance to object and/or instance naming.

THE REGISTRATION HIERARCHY

This hierarchy is not based on any particular relationship, and is independent of both the inheritance and containment hierarchies. It contains Object Identifiers for object classes and attributes, as assigned by the standards' registration authority.

The registration hierarchy is important because it is used for identifying object classes and attributes. It is used to ensure global uniqueness and to permit extensions without a centralized registration authority.

18.7.1.1 Basic Concepts

The following concepts/features of the information model are supported, as specified in Clause 5 of [MIM].

managed object
managed object class
managed object instance
attribute
group attribute
set-valued attribute
attribute value assertion
management operation
encapsulation
behaviour
notification

18.7.1.2 Management Operations Supported

The following management operations are supported, as specified in Clause 5.2 of [MIM].

Operations that apply to attributes :

Get attribute value
Replace attribute value
Set-to-default value
Add attribute value
Remove attribute value

Operations that apply to managed objects :

Create
Delete
Action

18.7.1.3 Filter

The concept of filter is supported as specified in Clause 5.3 of [MIM]. Restrictions on its usage are specified in Section 18.6.2.2.2 of these agreements.

18.7.1.4 Inheritance

All the inheritance related concepts (refinement, subclass, superclass, inheritance hierarchy, etc) presented in clause 5.5 of [MIM] are supported.

The following additional constraints need to be enforced for the Phase 1 IAs in order to remove potential ambiguities:

Subclasses must inherit ALL the optional attributes of their respective superclasses. Once inherited, these attributes may remain as optional attributes of the subclass or may become mandatory attributes of the subclass.

When an instance of a managed object class is created, it must support all the mandatory attributes defined for that class. The instance may support some or none of the optional attributes defined for its class. Once created, the managed object instance must support , throughout its lifetime, exactly the same set of attributes that were assigned to it at the time of creation, i.e. dynamic creation/deletion of attributes within an object instance is not allowed.

During the lifetime of a managed object instance, each of its attributes must have a value that is valid for the attribute syntax of that attribute.

The range of the attribute values for any attribute may not be redefined in the process of refinement. If it is anticipated that the range of attribute values may change, then the use of the ASN.1 enumerated type for the attribute syntax is discouraged.

Multiple inheritance is not supported for the Phase 1 IAs, since no requirements for it have been voiced within the NMSIG.

18.7.1.5 Polymorphism

Editor's Note: Polymorphism is a very useful concept insofar as it facilitates interoperability across different versions and vendor extensions of a managed object class. However, issues and problems related to it, especially those dealing with the naming of polymorphic classes, have not been thoroughly examined or resolved in the standards. Given this, does NMSIG feel the need to incorporate polymorphism into the Phase 1 IAs ?

Polymorphism is not supported for the Phase 1 IAs, since no requirements for it have been voiced within the NMSIG.

18.7.2 Principles of Naming

This subsection contains agreements about principles of naming as specified in Clause 6 of [MIM].

18.7.2.1 Containment Hierarchy

All concepts about the containment hierarchy presented in Clause 6.1 of [MIM] are supported.

18.7.2.2 Name Structure

18.7.2.2.1 Object Class Identification

A managed object class is identified by an ASN.1 object identifier, as specified in Clause 6.2.1 of [MIM].

18.7.2.2.2 Object Instance Identification

The distinguished name approach is supported for the identification of managed object instances.

Editor's Note: Many issues/questions regarding the naming of managed object instances have arisen because the related standards' text (Clause 6.2.2 of [MIM]) is somewhat unclear.

The following issues related to naming managed object instances are identified :

a) Referring to the first sentence of Clause 6.2.2 of [MIM], which starts with "The definition of each managed object class ...", does "an" identification attribute imply "only one" or "at least one" ? Can different name bindings for the same managed object class specify different distinguishing attributes, or is there just one distinguishing attribute per managed object class ?

b) Do name bindings get inherited ?

c) Is the distinguishing attribute of a subclass the same or different from distinguishing attribute of its superclass? If the superclass and its subclass have the same distinguishing attribute, there could be ambiguities in situations where instances of both the superclass and its subclass exist in the containment tree. If the superclass and its subclass do not have the same distinguishing attribute, polymorphism cannot be supported.

d) What is the point of reference from which managed object instances are defined - full distinguished name or partial distinguished name?

18.7.2.2.3 Selection Of Distinguishing Attributes

The distinguishing attribute for a managed object class must be very carefully selected. It must be able to distinguish not only between instances of the object class for which it is defined, but also between instances of all other object classes that have the same superior object class. For example, consider the following figure which shows the structure of a containment tree :

```
          A
         / \
        /   \
       B     C
      /
     /
    C
```

Here, A represents instances of Object Class A, B represents instances of Object Class B and C represents instances of Object Class C. As can be seen from the figure, instances of Object Class C may be contained in either instances of Object Class A, or in instances of Object Class B. When the RDN of Object Class C is defined, it is necessary to make sure that it is different from the RDN for Object Class B. If Object Class B and Object Class C were to support the same RDN, it would not be possible to unambiguously traverse down the containment tree from A.

The above example shows a simple containment tree. In the real world, however, containment trees could be much more complex, and the selection of distinguishing attributes could involve extensive checking and verification over multiple object classes.
Editor's Note: Consider the following proposal :

> "The process of selecting the correct distinguishing attribute can be made simpler if every object class supports an additional distinguishing attribute called "My Object Class", whose value identifies the object class it is contained in. If this is done, the process of

> selecting and verifying the RDN of an object class would not require the consideration of object classes other than the one defining the RDN."

The above proposal will be worked on by the NMSIG and submitted to the standards.

18.7.2.2.4 Attribute Identification

Each individual attribute of a managed object is identified by an ASN.1 object identifier, as specified in Clause 6.2.4 of [SMI Part 1].

18.7.3 Guidelines for the Definition of Management Information

This subsection contains agreements about guidelines for the definition of management information, as specified in [GDMO]. These guidelines form a normative part of the standard; hence they must be strictly followed while defining management information.

18.7.3.1 Syntactical Definitions of Management Information

18.7.3.1.1 Managed Object Class Template

For Phase 1 IAs, the template supported by NMSIG for defining managed object classes is the same as the Managed Object Class template defined in Clause 9.3.2 of [GDMO], with the agreement that the optional clauses BEHAVIOUR DEFINITIONS, DIRECTORY and POLYMORPHIC SET are not to be used. The BEHAVIOUR DEFINITIONS clause is not supported because it calls for the use of Formal Definitions Techniques, specifications of which are not currently available. Behaviourial aspects of Managed Object Classes are instead captured in the semantic definitions of management information, described in section 18.7.3.2. The DIRECTORY clause of the managed object class template is not supported because the Phase 1 IAs do not require the use of directory services. The POLYMORPHIC SET clause is not supported, as per the agreements on polymorphism specified in 18.7.1.5.

Supporting productions for "propertylist" and "modifier" are adopted as specified in Clause 9.3.2 of [GDMO].

Supporting definitions of the DERIVED FROM, POLYMORPHIC SET, ATTRIBUTES, GROUP ATTRIBUTES, OPERATIONS, CREATE, DELETE, ACTIONS, NOTIFICATIONS, OPTIONAL ATTRIBUTES AND OPTIONAL GROUP ATTRIBUTES clauses of the managed object class template are adopted as defined in Clause 9.3.3 of [GDMO].

18.7.3.1.2 Name Binding Template

The NAME BINDING template is supported as described in Clause 9.4 of [GDMO].

18.7.3.1.3 Attribute Template

The ATTRIBUTE template is supported as described in Clause 9.5 of [GDMO].

18.7.3.1.4 Group Attribute Template

The GROUP ATTRIBUTE template is supported as described in Clause 9.6 of [GDMO].

18.7.3.1.5 Action Template

The ACTION template is supported as described in Clause 9.8 of [GDMO].

18.7.3.1.6 Notification Template

The NOTIFICATION template is supported as described in Clause 9.9 of [GDMO].

18.7.3.2 Semantic Definitions of Management Information

The following details should be provided in the definition of each managed object class:

- a textual description of the network resource it represents, including its functional role in the network.

- a description of the relationships that this managed object class participates in with instances of the same or other managed object classes.

- a description of contained objects.

- a description of the operations that are supported by it, with precise definitions of the effects, side effects, if any, constraints, response notifications, failure modes, etc.

- a description of its attributes.

- specification of how instances of this managed object class are created and deleted, particularly whether they can be created/deleted via the management CREATE/DELETE operations.

- a description of applicable thresholds, tidemarks, etc.

- a description of events that can be generated, the conditions that generate them, their contents and side-effects, if any.

- other constraints, including those involving other managed object classes.

18.7.3.3 Other Guidelines

The Systems Management functions have defined various attributes and events, as indicated in section 18.5 of these agreements. Object Definers are encouraged to make use of these attributes and events wherever applicable.

19. REMOTE DATABASE ACCESS (RDA)

Editor's Note: This section serves as a placeholder for text provided by the newly-formed Remote Database Access (RDA) Special Interest Group.

20. MANUFACTURING MESSAGE SPECIFICATION (MMS)

20.1 INTRODUCTION

This section defines Implementors Agreements based on ISO Manufacturing Message Specification (MMS), as defined in ISO 9506. This International Standard has two parts. Part 1 of the IS defines the Virtual Manufacturing Device (VMD) as well as defining the services, and Part 2 defines the Protocol. Future parts may define companion standards.

MMS, 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 agreements can be used over any T-Profile (see ISO DTR 10000) specifying the OSI connection-mode transport service. In addition, these MMS agreements can be used over the Transport profiles used in support of MAP (Manufacturing Automation Protocol) or TOP (Technical and Office Protocols).

20.1.1 References

Application Layer - MMS

ISO 9506-1: 1988	Manufacturing Message Specification Service Definition
ISO 9506-2: 1988	Manufacturing Message Specification Protocol Specification

20.2 SCOPE AND FIELD OF APPLICATION

There will be a phased grouping of implementation agreements. These agreements will be based on selected subsets of MMS services as defined in ISO 9506-1. Agreements will be defined in phases which will be added as needed.

20.2.1 Phase I Agreements

These agreements will be implementation agreements pertaining to the services as specified as Table 1.

20.3 STATUS

20.3.1 Status of Phase 1 Agreements

Phase 1 is in progress.

20.4 ERRATA

None at time of publication.

20.5 SPECIFIC SERVICE AGREEMENTS

20.5.1 Initiate

20.5.1.1 Max Serv Outstanding

- An MMS Implementation which intends to conform only with the Client Conformance Requirements for Requester CBBs shall:

    1. propose 1 or greater for the value of the Proposed Max Serv Outstanding Calling parameter in the Initiate service when initiating the application association (calling).

    2. offer 1 or greater for the value of the Negotiated Max Serv Outstanding Called parameter in the Initiate service when receiving the application association initiation (called).

- An MMS Implementation which intends to conform to one or more Server Conformance Requirements for Responder CBBs shall:

    1. propose 1 or greater for the value of the Proposed Max Serv Outstanding Called parameter in the Initiate service when initiating the application association (calling).

2. offer 1 or greater for the value of the Negotiated Max Serv Outstanding Calling parameter in the Initiate service when receiving the application association initiation (called).

20.5.1.2 Version Number

- The value of zero, for the proposed Version Number in the Initiate request and the negotiated Version Number in the Initiate response service primitives, is reserved to enable interoperability with existing DIS based implementations.

Tutorial:

There is an installed base of real DIS 9506 based implementations. Providing support for application connectivity to both DIS and IS is desired as a migration strategy. It was found that the Abstract Syntax name object identifiers of both DIS and IS were identical. Therefore, the use of Version 0 allows differentiation between an IS and a DIS based implementation.

Note: The value of zero is a valid value for these parameters in the DIS and not in the IS.

20.5.1.3 Minimum Supported PDU Size

MMS implementations must be able to parse and process 64 octets of MMS pdu as they would be encoded in ASN.1 Basic Encoding Rules. However, it is recommended that 512 be supported.

20.5.1.4 Max Supported PDU Size

The max_mms_pdu_size is defined as the maximum number of octets in an MMS pdu encoded using the negotiated transfer syntax. This size shall apply to all MMS PDU's with the exception of the initiate-Request PDU, initiate-Response PDU, and initiate-Error PDU. The max_mms_pdu_size shall be negotiated during connection initiation using the Local Detail Calling and Local Detail Called parameters of the MMS initiate service.

The semantics of these parameters follows:

Local Detail Calling

The local detail calling parameter in the initiate request primitive shall specify the max_mms_pdu_size guaranteed to be supported by the calling MMS-user. The local detail calling parameter in the initiate indication primitive shall specify the max_mms_pdu_size guaranteed to be supported by both the Calling MMS-user and the MMS-provider. This shall be less than or equal to the max_mms_pdu_size specified in the initiate request primitive.
If the local detailcalling parameter is absent from the request primitive, then the calling MMS-user guarantees support for an unlimited max_mms_pdu_size. If the MMS-provider is not able to make this guarantee, then this parameter shall be supplied in the indication primitive with the largest non-zero value which the MMS-provider is capable of supporting. Otherwise, it shall be absent from the indication primitive, indicating that the Calling MMS-user and the MMS-provider are prepared to support an unbounded max_mms_pdu_size.

If present in the request or indication primitives, the local_detail_calling parameter shall not be less than 64.

Local Detail Called

The local detail called parameter in the initiate response shall specify the negotiated max_mms_pdu_size for the application association.

If the local detail calling parameter was omitted in the indication primitive, then the local_detail_called parameter:

1. may be omitted from the response primitive, indicating that the calling MMS-user, the MMS-provider and the Called MMS-user are prepared to support an unbounded max_mms_pdu_size, or,

2. may be specified in the response primitive, indicating a requirement to support the specified value for max_mms_pdu_size.

If the local detail calling parameter was included in the indication primitive, then the value of this parameter shall be less than or equal to the value of the local detail calling parameter of the indication primitive.

If present in the response or confirm primitives, the local detail called parameter shall not be less than 64.

The negotiated max_mms_pdu_size shall be applied as follows: Any received MMSpdu which is less than or equal to the negotiated max_mms_pdu_size shall be properly parsed and processed.

When rejecting an MMS-pdu because it exceeds the negotiated max_mms_pdu_size, an MMS implementation shall use a pdu type of pdu_error and a reject code of invalid_pdu in the resulting reject PDU.
An MMS implementation shall not send an MMSpdu whose size exceeds the negotiated max_mms_pdu_size.

> If an MMS implementation is unable to send a service response because the response would exceed the max_mms_pdu_size, then it shall return a Service response (-) with an error class of SERVICE and an error code of OTHER.

20.5.1.5 Negotiation of MMS Abstract Syntaxes

On initiate response, the MMS responder shall not accept more than one presentation context derived from an MMS abstract syntax (in this context, only the core MMS abstract syntax and the Companion Standard defined abstract syntaxes, are considered MMS abstract syntaxes).

20.5.2 Scattered Access

It is strongly recommended that for services which use variable access, a Variable List Name or List of Variable be used instead of Scattered Access.

No implementations shall be required to propose or accept the VSCA Parameter CBB.

20.6 INTEROPERABILITY AGREEMENTS

These implementation agreements will allow IS based implementations to interoperate with DIS based implementations as described in Appendix A. To achieve this interoperability, the IS implementation shall support all of the agreements in this section.

TUTORIAL SECTION:

There are three types of implementations when considering MMS interoperabilty.

IMP-1: An implementation based on DIS 9506 as described in Appendix A.

IMP-2: An implementation based on IS 9506 with no interoperability agreements applied.

IMP-3: An implementation based on IS 9506 which includes the interoperabilty agreements described below.

IMP-1, IMP-2, and IMP-3 can interoperate with each other in all combinations with the exception of the IMP-1 and IMP-2 combination. The remainder of this section describes additional agreements which change an IMP-2 implementation into an IMP-3 implementation.

20.6.1 Calling MMS-user Interoperability Agreements

A calling MMS-user shall be capable of receiving and supporting a negotiatedVersionNumber parameter in the Initiate Service Confirm of zero.

A calling MMS-user which has received a negotiatedVersionNumber parameter in the Initiate Service Confirm of zero shall support the modifications described in section 20.6.3.

A calling MMS-user shall ignore the Application Context Name parameter in the A-Associate Confirm.

20.6.2 Called MMS-user Interoperability Agreements.

A called MMS-user shall be capable of receiving and supporting a proposedVersionNumber parameter in the Initiate Service Indication of zero.

A called MMS-user which has received a proposedVersionNumber parameter in the Initiate Service Indication of 0 shall support the modifications in section 20.6.3.

A called MMS-user shall ignore the Application Context Name parameter in the A-Associate Indication.

20.6.3 General Interoperability Agreements

20.6.3.1 VMD Logical Status

If the current VMD State is SUPPORT-SERVICES-ALLOWED and the association minor version number is zero, then the vmdLogicalStatus parameter shall have a value of state-changes-allowed in a status response or a unsolicitedStatus request.

20.6.3.2

Further agreements are required to complete this section.

20.7 APPENDIX A: DIS 9506 MODIFICATIONS REQUIRED FOR INTEROPERABILITY

This appendix is an integral part of Chapter 20. It documents the modifications to DIS 9506 required to describe implementations for which the IS agreements provide interoperability. This appendix as applied to DIS 9506 is referred to as Version 0.

20.7.1 References

[1] MMS/1 Manufacturing Message Specification - ISO DIS 9506 - Service Definition, December 1987

[2] MMS/2 Manufacturing Message Specification -ISO DIS 9506 - Protocol Specification, December 1987

[3] NBS OSI Implementors Workshop Agreements - December 1987

20.7.2 Version 0

20.7.2.1 General

20.7.2.1.1 Implementation Base

Version 0 is based upon Reference [3] in 20.7.2 as it applies to MMS.

20.7.2.1.2 Rules of Extensibility

The following sentence is appended to the last paragraph in section 8.2.1.1.5.2 Proposed Parameter CBB and the last paragraph in section 8.2.1.2.5.2 Negotiated Parameter CBB of DIS 9506-1.

"Any additional bits shall be ignored."

20.7.2.2 Modifications to the Protocol definitions

20.7.2.2.1 Page39, Section 7.5.2 of DIS 9506-2

CHANGE

```
reportEventEnrollmentStatus  [60] IMPLICIT
          ReportEventEnrollmentStatus-Request,
```

TO

```
reportEventEnrollmentStatus  [60]
                             ReportEventEnrollmentStatus-Request,
```

20.7.2.2.2 Page 49, Section 7.6.4, DIS 9506-2

CHANGE

```
ApplicationReference ::= SEQUENCE {
   ap-title         ISO-8650-ACSE-1.AP-title OPTIONAL,
   ap-invocation-id    ISO-86 50-ACSE-1.AP-invocation-id OPTIONAL,
   ae-qualifier   ISO-8650-ACSE-1.AE-qualifier OPTIONAL,
   ae-invocation-id    ISO-8650-ACSE-1.AE-invocation-id OPTIONAL
                                  }
```

TO

```
ApplicationReference ::= SEQUENCE {
   ap-title         [0] OBJECT IDENTIFIER OPTIONAL,
   ap-invocation-id [1] INTEGER OPTIONAL,
   ae-qualifier   [2] INTEGER OPTIONAL,
   ae-invocation-id    [3] INTEGER OPTIONAL
                                  }
```

20.7.2.2.3 Page 95, Section 12.2.1 of DIS 9506-2

CHANGE

```
structure [2] IMPLICIT SEQUENCE OF SEQUENCE {
```

TO

```
structure [2] IMPLICIT SEQUENCE {
```

20.7.2.2.4 Page 96, Section12.3.1 of DIS 9506-2

CHANGE

```
named [4] IMPLICIT SEQUENCE {
```

TO

```
named [5] IMPLICIT SEQUENCE {
```

20.7.2.2.5 Page 98, Section 12.4.2 of DIS 9506-2

CHANGE

```
generalized-time [10] IMPLICIT GeneralizedTime,
```

TO

generalized-time [11] IMPLICIT GeneralizedTime,

20.7.2.2.6 Page 138, Section 15.14 of DIS 9506-2

CHANGE

```
additionalDetail        [9] IMPLICIT EE-Additional-Detail OPTIONAL
```

TO

```
additionalDetail        [9] EE-Additional-Detail OPTIONAL
```

20.7.2.2.7 Page 166, Section 17.10 of DIS 9506-2

CHANGE the transfer syntax object identifier value from

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

TO

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

20.7.2.3 Behavioral Requirements

20.7.2.3.1 Filenames

File Names are specified in accordance with the NBS Implementors'agreements for FTAM Reference [3] in 20.7.2.

20.7.2.3.2 Identify Service

In the Identify service, the vendor, model and revision fields may be of any length, but only the first 64, 16, and 16 octets respectively are treated as significant.

20.7.2.3.3 Initiate Service

An MMS Client will:

1. propose 1 or greater for the value of the Proposed Max Serv Outstanding Called parameter in the Initiate service when

initiating the application association (calling).

2. offer 1 or greater for the value of the Negotiated Max Serv Outstanding Calling parameter in the Initiate service when receiving the application association initiation (called).

An MMS Server will:

1. propose 1 or greater for the value of the Proposed Max Serv Outstanding Calling parameter in the Initiate service when initiating the application association (calling).

2. offer 1 or greater for the value of the Negotiated Max Serv Outstanding Called parameter in the Initiate service when receiving the application association initiation (called).

20.7.2.3.3.1 Segment Size

20.7.2.3.3.1.1 Minimum Segment Size

MMS implementations are able to parse and process 512 octets of MMSpdu as they are encoded in ASN.1 basic encoding rules.

20.7.2.3.3.1.2 Maximum Segment Size

The Max Segment Size is defined as the maximum number of octets in an MMS pdu encoded using the negotiated transfer syntax. This size will apply to all MMS pdus with the exception of the initiate-Request PDU, initiate-Response PDU, and the initiate-Error PDU. The max segment size will be negotiated during connection initiation using the Proposed Max Segment Size and Negotiated Max Segment Size parameters of the MMS initiate service.

The Max Segment Size will be applied as follows:

Any received MMSpdu which is less than or equal to the Max Segment Size will be properly parsed and processed.

An MMS implementation will not send an MMSpdu whose size exceeds the Max Segment Size.

20.7.2.3.3.2 Abstract Syntax Name

The ASN.1 object identifier value for the abstract syntax name will be the same as specified on Page 166, Section 17.10 DIS 9506-2.

20.7.2.3.3.3 Application Context Name

The ASN.1 object identifier value for the application context name will be the same as specified on Page 166, Section 17.11 DIS 9506-2.

An MMS-user ignores the Application Context Name in the A-Associate indication and the A-Associate confirm.

20.7.2.3.4 Minor Version Number

The Minor Version Number is zero.

20.7.2.3.5 Parameter CBB Subset

The following subset of MMS Parameter CBBs were considered during preparation of this appendix.

STR1,
NEST,
VADR,
VNAM

20.7.3 Service Subset

The following subset of MMS services were considered during preparation of this appendix.

Initiate,
Conclude,
Cancel,
Status,
GetNameList,

```
Identify,
UnsolicitedStatus,
GetCapabilityList,
InitiateDownloadSequence,
DownloadSegment,
TerminateDownloadSequence,
InitiateUploadSequence,
UploadSegment,
TerminateUploadSequence,
RequestDomainDownload,
RequestDomainUpload,
LoadDomainContent,
StoreDomainContent,
DeleteDomain,
GetDomainAttributes,
Read,
Write,
InformationReport,
GetVariableAccessAttributes,
Input,
Output,
TakeControl,
RelinquishControl,
ReportSemaphoreStatus,
ReportPoolSemaphoreStatus,
ReportSemaphoreEntryStatus,
CreateProgramInvocation,
DeleteProgramInvocation,
Start,
Stop,
Resume,
Reset,
Kill,
GetProgramInvocationAttributes,
ObtainFile,
GetEventConditionAttributes,
ReportEventConditionStatus,
GetAlarmSummary,
ReadJournal,
WriteJournal,
InitializeJournal,
CreateJournal,
DeleteJournal,
ReportJournalStatus
```

21. REFERENCES

Editor's Note: In this document, references are maintained in the individual sections as appropriate. Additional references for all of the subject covered in this document may be found in the aligned references section of the Stable Implementation Agreements Document, Version 3, Edition 1, December 1989.

READER RESPONSE FORM

Please retain my name for the next mailing of the NIST/OSI Implementors Workshop.

NAME: ______________________________

ADDRESS: ______________________________

____

____

PHONE NO.: ______________________________

Mail this page to:

National Institute of Standards and Technology
NIST Workshop for Implementors of OSI
Brenda Gray, Registrar
Building 225, Mail Stop B-217
Gaithersburg, MD 20899

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 90-4247

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
FEBRUARY 1990

4. TITLE AND SUBTITLE

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

5. AUTHOR(S)

Tim Boland, Editor

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

□ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document records current agreements on implementation details of Open Systems Interconnection Protocols among the organizations participating in the NIST/OSI Workshop Series for Implementors of OSI Protocols. These decisions are documented to facilitate organizations in their understanding of the status of agreements. This is a standing document that is updated after each workshop (about 4 times a year).

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

NIST/OSI WORKSHOP, LOCAL AREA NETWORKS: NETWORK PROTOCOLS: OPEN SYSTEMS INTERCONNECTION: TESTING PROTOCOLS

13. AVAILABILITY

XX UNLIMITED

□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

XX ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
354

15. PRICE
A16

ELECTRONIC FORM

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990 — 261 - 913/ 20516