NISTIR 89-4140

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held June 16, 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

August 1989

Issued September 1989

NOTE: As of 23 August 1988, the National Bureau of Standards (NBS) became the National Institute of Standards and Technology (NIST) when President Reagan signed into law the Omnibus Trade and Competitiveness Act.

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NISTIR 89-4140

WORKING IMPLEMENTATION AGREEMENTS FOR OPEN SYSTEMS INTERCONNECTION PROTOCOLS

Based on the proceedings of the
NIST Workshop for Implementors of OSI
Plenary Assembly Held June 16, 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

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

Table of Contents

List of Figures

List of Tables

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/OSI Workshop Series for Implementors of OSI Protocols. 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.

Only non-stable text is included in this document. Errata to Stable material 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.

- The companion document, "Stable Implementation Agreements for Open Systems Interconnection Protocols, Version 2, Edition 3, June 1989" records mature agreements considered advanced enough for use in product development or procurement reference. This document is released with a version number.

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

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

The benefit of this document is that it gives the reader a glimpse of new functionality, for planning purposes. Together with the aligned, associated stable document plus errata, 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 plus any errata described in this working document to get the true current status of stable material. In this Working Document, all references to the Stable Document are to V2, E3 (June 1989). Where appropriate, statements related to backward compatibility or interworking considerations 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 ANSC X3T5 or ANSC 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.

- 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).

- The SIG 2 chairperson should bring the matter before SIG 2 for action.

- SIG 2 should respond to the concerns or needs of SIG 1 or the individual in a timely manner.

- 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.

- 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

- to develop stable FTAM Agreements between vendors and users for the implementation of interoperable products

- in particular to develop the FTAM Phase 2 product-level specifications and maintain these specifications with respect to experiences from implementations and from testing

- to define further FTAM functionality in the Phase 3 specifications. These will contain only extensions of FTAM Phase 2. It is a goal that Phase 3 will be backward compatible with FTAM Phase 2. The set of future work items listed below may be changed by the plenary if the work is more appropriate for other SIGs.

- to conduct liaison with and contribute to other bodies working on FTAM harmonization such as CEN/CENELEC, POSI, and the ISO activities to define Functional Standards

and

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

High priority work items:

- Complete and maintain FTAM Phase 2 Agreements
- Specify implementation of Error Recovery control procedures, specifically
- Error Recovery and Restart Data Transfer functional units
- Specify Concurrency Control parameter.
- Specify implementation of Character Set ISO 6937
- Specify requirements of FTAM to a Directory Service
- Specify use of Presentation Context Management functional unit.

Low priority work items:

- Add new Document Types/Constraint Sets
- Define use of Access Control
- Specify FADU Locking functional unit
- Specify File Store management (e.g., file directories)
- Specify File Name conventions
- Specify use of Overlapped Access

X.400 (MESSAGE HANDLING SYSTEMS) SIG

Develop product-level specifications for Message Handling Systems using the CCITT X.400 Recommendations.

Develop abstract tests for X.400, as requested by the ad hoc rapporteur for this study question in CCITT. This work is to be submitted by the plenary (after its approval) to the U.S. Department of State as a proposed U.S. contribution to CCITT Study Group VII.

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,
- 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 Begin work on the implementation specification of the ISO Network Layer Routing Exchange Protocol prior to the ISO draft achieving DIS status.

The Lower Layer SIG will study both existing and emerging ISDN standards pertaining to user access and user services. The SIG will:

- o Develop implementation agreements for user-network interfaces
- o Develop conformance requirements
- o Conduct Liaison with other standards/interest groups

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:

- o to develop agreements based on IS/DIS
- o to develop/draft NWI proposals for submission to national bodies on areas not covered by existing standards work
- o to draft contributions on proposed NWIs
- o to register security objects
- o to provide consultancy to other SIGs
- o 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.

- o Provide a subset for NIST publication which is functional and forward compatible to further work by this Special Interest Group.

- o 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 Draft International Standards 9040 and 9041 and their respective addenda, in providing Basic Virtual Terminal Service.

This group will develop agreements for the implementation and testing of the following terminal types.

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

UPPER LAYERS SIG

The charter of the Upper Layers SIG is as follows.

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

- o 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.

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

The objectives of the Upper Layers SIG are to:

- o Study OSI Session, Presentation, ACSE, ROSE, and RTSE
- o Incorporate implementor's agreements in the 1988 NBS standing document,
- o Produce and maintain recommendations for implementations of these layers,
- o Where necessary provide input to the relevant standards bodies concerning Session, Presentation, ACSE, ROSE, and RTSE
- o React in a timely manner (i.e., to develop corresponding implementor's agreements) to technical changes in ISO documents.

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

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

NETWORK MANAGEMENT SIG

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

The SIG will:

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

- o Initiate and coordinate with appropriate layer SIGs product level specifications of layer-specific management information to support FM, SM, CM, AM, and PM.

As necessary, the SIG will:

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

OFFICE DOCUMENT ARCHITECTURE

The SIG will:

- o 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)
- o develop requirements for conformance testing of products purporting conformance to the (se) DAP (s)
- o specify and describe requirements for services that manage the generation and interpretation of the ODA document representation
- o determine preferred relationships between ODA and other document interchange formats
- o promote the SIG's agreements (e.g., presentations, product demonstrations, press releases)

As necessary, the SIG will:

- o establish liaison with required SIGs (e.g., X.400, FTAM, and Upper Layers SIGs) to seek efficient transfer capability for document interchange based on the ODA SIG agreements
- o 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

<u>High Priority Work Items</u>

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

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

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

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

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

<u>Low Priority Work Items</u>

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

- Develop subsequent subsets of MMS

- Produce a set of implementors agreements for the subsequent subsets

- o Provide input on Test Cases for the subsequent subsets
- o Provide input on errors for the subsequent subsets
- o Provide input to ISO on MMS ASE specific management entities.

REMOTE DATABASE ACCESS SIG

Scope:

For all RDA Implementations based on ISO 9579:

- o Develop Implementors' agreements;
- o Provide input to national and international standards organizations on RDA related standards and profiles;
- o Coordinate with other organizations on matters relevant to RDA.

Objectives:

- o Use ISO 9579 Generic RDA and the ISO SQL Specialization as a basis for Implementors' Agreements on the RDA SQL ASE and its application contexts;
- o Provide input to ANSI and ISO on the specification of an RDA ISP.

High Priority Work Items

1. To develop a work plan for RDA Implementors' Agreements with an associated time schedule, using the following tasks as a basis:
   a. review ULA agreements affecting RDA implementations,
   b. specify limits on encodings in RDA pdus,
   c. specify minimum conformance requiremnts 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	Wollongong	(415) 962-7160
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
OSINET Steering Committee	Jerry Mulvenna	NIST	(301) 975-3631
OSINET Technical Comm.	Carol Edgar	NIST	(301) 975-3613
Remote Database Access SIG	Rich Gerhardt	GM	(313) 947-0572
Registration SIG	Einar Stefferud	NMA-Northrop	(714) 842-3711
Security SIG	James Galvin	Trusted Info. Sys.	(301) 854-6889
Technical Liaison Committee	J.J. Cinecoe	Wang	(508) 967-5514
Transaction Processing SIG Vice Chair	Jeff Hildebrand	Boeing	(206) 865-7028
Upper Layers SIG	David Chappell	Cray Research	(612) 825-7928
Virtual Terminal SIG	Cyndi Jung	3COM	(415) 940-7664
X.400 SIG	Barbara Donoghue	Retix	(213) 399-2200
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 2, Edition 3, dated June 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 June 16, 1989.

2.4 ERRATA

Errata are reflected in replacement pages of Version 2, Edition 3, Stable Document, dated June 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

(Refer to Stable Implementation Agreements Document)

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_Reg 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 receive Info_Fields of 0 to 4478 bytes. The frame is defined as follows:

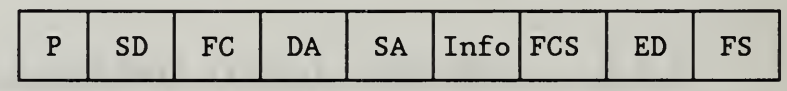


Figure 2.1 FDDI STATION

P: Preamble (4 Idle Symbols)
SD: Starting Delimiter (2 Symbols, JK)
FC: Frame Control (2 Symbols)
DA: Destination Address (12 Symbols)
SA: Source Address (12 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 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 delay which can result is 1.164 microseconds. This number, not one microsecond, should be used per PHY to compare maximum possible network delay.

2.5.5.3 Physical Layer Media Dependent (PMD, X3.166-198X)

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 2.5×10^{-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).

2.7.3 Rate Adaptation[1]

The following recommendations are made with respect to implementation of Draft T1E1.4/88-071, V.120 ISDN Rate Adaptation Specifications.

1 The preferred method of Information Transfer (V.120 Section 3.5) in Asynchronous Protocol Sensitive mode is Multiple Frame Acknowledged Information Transfer.

2 V.120 terminal adapters should not resend the last I-frame transmitted as a poll upon expiry of timer T200 (although they must respond appropriately if they receive an I-frame poll).

---

1 It is recognized that these agreements are not relevant to implementations of OSI. They were originally developed at the request of the NIST NIU Executive Committee and are temporarily included in these agreements until a comparable ISDN Agreements document is available.

3. NETWORK LAYER

Editor's Note: All references to Stable Agreements in this Section are to Version 2, Edition 3, dated June 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 June 16, 1989.

3.4 ERRATA

Errata are reflected in replacement pages of Version 2, Edition 3 Stable Document, dated June 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:

- (Refer to the Stable Implementations Agreements document).
- 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.[2]

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

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

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

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

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

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

---

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

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

Editor's Note: The intention was expressed to delete bullets 1 and 2 in 3.6.3.1 in Version 2, Edition 3, Stable Agreements Document.

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

(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:

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 Stable Agreements Document.

Editor's Note: The current intent is to possibly replace item 6 of the Stable Document with the text below.[2]

6. If the configuration notification function described in clause 6.7 of the protocol specification is implemented, a mechanism shall be provided to enable/disable this function on broadcast networks.

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

   IS Actions

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

---

2 This represents a change in functionality in the Stable Implementation Agreements.

1. Delay a random interval between 0 and PollRate seconds. (This is to avoid synchronization with other ISs.)

2. Then transmit at least one IS Hello with a Suggested ES Configuration Timer value of PollRate seconds. If more than one IS Hello is sent (to overcome possible loss) delay PollRate seconds between sending each.

3. Then start sending IS Hellos with a Suggested ES Configuration Timer of DefaultRate seconds (where Default rate is larger than PollRate).

ES Actions

When an End system receives an IS Hello which contains a Suggested ES Configuration Timer, it must recompute its Configuration Timer as described in section 6.3.2 of the protocol standard. It then determines when to send its next ES hello by choosing the minimum of:

a) the current remaining time interval before sending an ESH, and

b) a random interval between 0 and the new Configuration Timer.

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[3]

3.11.1 Introduction

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

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

---

3

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

provided for Network and Transport Layer protocols. The implication of these agreement for upper lower 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

- 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
- 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.
- 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

o 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.

o 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 2ND XMT BIT
    I/G 1ST XMT 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.

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

o 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 1ST XMT BIT
  U/L 2ND XMT 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 octent. 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 2, Edition 3, dated June 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 June 16, 1989.

4.4 ERRATA

Errata are reflected in replacement pages of Version 2, Edition 3, Stable Document, dated June 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 Rules for Negotiation

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

(For other rules refer to the Stable Implementation Agreements document).

4.5.1.2.2 Transport Class 4 Service Access Points or Selectors

(Refer to Stable Implementation Agreements Document)

4.5.1.2.3 Retransmission Timer

Refer to Stable Implementation Agreements Document.

Editor's Note: The following text may in the future go after the third paragraph of this section in the Stable Document.

As network load increases, the variability of round-trip delay also increases. In environments where load fluctuates widely, it is therefore important to use the formula:

$$T1 \leftarrow E + AR + 2k$$

where "k" is a variable which is sensitive to the variability of round-trip times rather than a constant value. The following is a suitable formula for determining "k" by estimating the variance of the round-trip times:

$$k \leftarrow k + g(|Err| - k)$$

In this formula "Err" is the difference of the previous round-trip time estimate and the latest round-trip time measurement; "g" is a value between 0 and 1 which determines how quickly the variance estimate reacts to changes in round-trip variability. A suitable value for "g" which allows for efficient implementation is .125.

This technique of variance estimation is particularly important when the optional congestion avoidance procedures are also used. In order to maintain optimum utilization of network resources, the congestion

avoidance procedures rely on timer calculations to prevent spurious retransmissions.

Temporary Note: The originator of the original contribution requested minor modifications to correct a typing error in the paper presented at the June Workshop. These have been included above but should be confirmed at the next meeting.

Editor's Note: The following text may in the future go at the end of this section in the Stable Document.

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

In the event of a retransmission timeout, the PDU should be retransmitted and the timer set with a value that is twice the previous value. When an acknowledgement of non-retransmitted data is received, the new round-trip time estimate should be calculated by the usual algorithm using the new round-trip measurement and the last estimate before the retransmission timeout(s).

4.5.1.2.4 Keep-Alive Function

(Refer to Stable Implementation Agreements Document)

4.5.1.2.5 Congestion Avoidance Policies

(Refer to the Stable Implementation Agreements document).

Mandatory Requirements

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

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

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

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

Optional Requirements

Refer to the Stable Implementation Agreements Document.

Editor's Note: For the Stable Document, it is intended in the future to modify the words "ALL STEs shall reset WS to one "under Rule 2 for STEs to read "ALL STEs shall adopt a WS (new), if WS (old) is not equal to 1, where

$$1 \leq WS\ (new) \leq \frac{WS\ (old)}{2}$$

In Rule 2, Line 3 for STEs, change WS to WS (new).

4.5.1.2.6 Use of Priority[4]

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

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

communications. In other words an explicit priority parameter shall be sent.

Additional requirements of systems implementing priority are defined below.

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

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:[6]

- "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[7].

---

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

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

[7] 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.

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. 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

4.5.3.1 Transport Class 2 Overview

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

4.5.3.2 Protocol Agreements

Transport Class 2 agreements follow:

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

  TS1: 60 seconds
  TS2: 60 seconds

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

- The rules for class negotiation shall be used.

  Negotiation from Class 2 to Class 0 is achieved by indicating Class 0 in the Alternative Protocol Class field of the CR TPDU which proposes Class 2. This is only possible when no other transport connections are assigned to the underlying network connection.

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

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

4.6 PROVISION OF CONNECTIONLESS TRANSPORT SERVICE

(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 2, Edition 3, June 1989.

5.1 INTRODUCTION

This section specifies agreements for the implementation of OSI upper layer protocols, including Session, Presentation, ACSE, ROSE, and RTSE.

5.1.1 References

(Refer to Stable Agreements Document.)

5.2 SCOPE AND FIELD OF APPLICATION

The agreements in this section apply to all ASE agreements in this document, including FTAM, X.400, Directory Services, Virtual Terminal, and OSI Network Management. All upper layer agreements specified in Chapter 5 of the NIST Special Publication "Stable Implementation Agreements for Open Systems Interconnection Protocols" (with errata) are also implicitly included in these agreements.

5.3 STATUS

This version of the upper layer agreements is under development.

5.4 ERRATA

Editor's Note: Errata are included as replacement pages in the aligned Version 2, Edition 3, Stable Document.

5.4.1 ISO Defect Reports

(See Stable Agreements Document.)

5.4.2 Session Defects

(See 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

It is the intention of the UL SIG to adopt ACSE defect report 8650/004 when it becomes stable. Values for and usees of AE-titles are outside the scope of the Upper Layer SIG.

5.5.4 ASN.1 Encoding Rules

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

5.5.5 Connectionless

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

5.6 ROSE

TBD

5.7 RTSE

TBD

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

(Refer to Stable Agreements Document.)

5.8.4 Presentation ASN.1 Encoding Rules

(Refer to Stable Agreements Document.)

5.8.5 General

5.8.5.1 Presentation Data Value (PDV)

- A Presentation data value (PDV) is a value of a type in an abstract syntax, e.g., a value of an ASN.1 type.
- A PDV may contain embedded PDVs in different contexts. A change of context within a PDV is indicated by an EXTERNAL. EXTERNAL implies an embedded PDV.
- A PDV cannot be split across PDV-lists in fully-encoded user data.
- Fully encoded data that is a series of PDVs in the same Presentation context should be encoded as one PDV-list.

5.8.6 Connection Oriented

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

5.8.7 Connectionless

The connectionless Presentation protocol shall be implemented as specified in ISO 2nd PDAD 9576.

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

5.9 SESSION

5.9.1 Introduction

(Refer to Stable Agreements.)

5.9.2 Services

(Refer to Stable Agreements.)

5.9.3 Protocol Agreements

(Refer to Stable Agreements.)

5.9.4 General

TBD

5.9.5 Connection Oriented

TBD

5.9.6 Connectionless

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

5.10 UNIVERSAL ASN.1 ENCODING RULES

5.10.1 TAGS

(Refer to Stable Document.)

5.10.2 Definite Length

(Refer to Stable Document.)

5.10.3 External

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

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

5.10.4 Integer

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

5.10.5 String Types

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

5.10.6 Bit String

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

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

5.11 CHARACTER SETS

These sections describe Information Processing Character Set policies and agreements of the NIST OSI Workshop. These policies and agreements are based upon ISO Character Set International Standards and CCITT Character Set Recommendations. The Policy section describes agreements on character set practices which the SIGs are expected to implement where the basic standards upon which Implementation Agreements are founded do not specify contrary requirements. The Agreements section records specific Workshop agreements on character sets. The Tutorial Appendix B summarizes the character set practices of each of the SIGs, including all relevant encoding information drawn from the appropriate ISO Registers, ISO Standards, and CCITT Recommendations.
The objectives of this section are to:

- Collect in one place all relevant character set information for all NIST OSI Workshop agreements and present relevant information from related standards (e.g., ASN.1),
- Establish policy for future NIST OSI Workshop Agreements,
- Describe character set conformance requirements,
- Record NIST OSI Workshop Character Set agreements, and
- Harmonize the use of character sets in conjunction with other OSI Workshops (e.g., EWOS and AOW).

5.11.1 Policy

Policy is defined to be a set of rules for formulating character set agreements. The SIGs are expected to abide by these policies to the extent possible under the constraints of their relevant standards. Exceptions should be recorded in Section 5.12.

5.11.1.1 Restrictions on Character Sets

An Application Service Element shall place no restriction on the character sets supported for user data, file contents, body parts, or other information which is passed through without processing (future processing).

5.11.1.2 Character Comparisons

All implementation agreements covering character comparisons and collation shall be recorded in this chapter.

5.11.2 Agreements

5.11.2.1 Encoding

5.11.2.1.1 Overprint, Composite Character

A composite character is defined as a diacritical in combination with an alphabetic as in ISO 6937. A composite character is considered as one character for purposes of comparison and character string computation.

With the exception of non-spacing diacriticals, 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, unless special provision has been made, subject to mutual agreement between the interchange parties. So, for example, the sequence "a BACKSPACE '" must be interpreted as three characters rather than as a single character.

5.11.2.1.2 Code Extension Facilities

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 and ESC 2/0 5/10. These sequences indicate:

- 8-bit environment,
- the G0, G1, and G2 graphic sets shall be used,
- no locking shift functions shall be used, and
- characters from G2 may be accessed by use of the single-shift 2 control function.

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

For ASN.1 T.61String ... <to be determined>

5.11.2.2 Comparisons

5.11.2.2.1 Matching Characters

A character value submitted with another character value does not have to be drawn from the same character set. However, the match is restricted to a list of pairs of character set values for which equality or inequality is defined. The result of comparing characters from a pair of character sets not in this list is undefined.

This list shows the pairs of character sets between which matching is defined.

ISO 6937-2 ISO 8859-1

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 composite characters which are formed from a diacritical followed by an alphabetic are not registered. Thus, the following table defines the match in terms of the ISO 6937-2 character name and the corresponding ISO Register name.

ISO 6937 name ISO Register Name

<to be added>

Editor's Note: The two subsections below have the same title.

5.11.2.2.2 Caseignore Comparisons

In character comparisons in which case is ignored, the matching rules of the section entitled "Matching Characters" are relaxed in that the characters are equal if their names differ only by one name having SMALL where the other name has CAPITAL.

5.11.2.2.3 Caseignore Comparisons

An agreement on comparison, other than equality or inequality, between characters requires a definition of a collating sequence. Such definitions shall be recorded in this chapter. The NIST OSI Workshop currently has no such agreements in place.

The collating sequence of letters, accented letters and other graphic symbols is not currently defined in an international standard or recommendation.

Preferred collating sequences might vary between countries.

5.11.2.2.4 Comparing 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 answer independent of each character string's ASN.1 packaging:

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

<this section will be further developed>

5.11.2.3 Agreements about Character Set Standards and Recommendations

This section covers agreements about:

- o subrepertoires supported,
- o standardized options selected,
- o 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,
- o the designation of component character sets within the ISO 2022 Code Extension Model where there is a choice to be made.

For tutorial purposes, the consequences of these agreements and the constraints of the related character set standards are brought together in Appendix B.

5.11.2.3.1 ISO 8859 Character Sets

Implementations supporting ISO 8859-1 are required to support the following two graphic character sets from the **International Register of Coded Character Sets to be used with Escape Sequences**:

6 ASCII Graphic Character Set in G0
100 Right Hand Part of Latin Alphabet No. 1 in G1

Support of ISO 8859-7 Greek Alphabet is optional as an addition to 8859-1. This option requires the following set from the **International Register of Coded Character Sets to be used with Escape Sequences**:

126 Right Hand Part of the Latin/Greek Alphabet

Within this option, sets 100 and 126 may be designated into G1 and G2 respectively, or into G2 and G1 respectively.

5.11.2.3.2 ISO 6937-2 Character Sets

Implementations supporting ISO 6937-2 are required to support ISO 6937-2 Addendum 1 and one or more of the following subrepertoires as defined in the **International Register of Subrepertoires**.

9 Western European data processing and interchange
3 Text communication in European Languages (Subrepertoire of graphic characters for teletex)

Implementations supporting ISO 6937-2 are required to use the following character sets from the **International Register of Coded Character Sets to be used with Escape Sequences**:

2 International Reference Version of ISO 646 in G0

142 Supplementary set of Latin Alphabetic and non Alphabetic Graphic Characters in G2

The supplementary set shall be designated in G2. For subrepertoires 2 and 5, the supplementary set may be omitted at the discretion of the sending application.

5.11.2.3.3 CCITT T.61

Implementations of CCITT Recommendation T.61 other than X.400-1984 must support the 1988 version.

Support for JIS X0208 is optional. If JIS is supported, it shall be designated into G1. Support for Greek is outside the scope of these agreements.

Dynamically Redefinable Character Sets (DRCS) shall not be used.

Support for T.61 as an ASN.1 GeneralString is outside of these agreements. Support of the graphic set components of T.61 as an ASN.1 GraphicString is outside the scope of these agreements.

The supplementary set of Graphic Character (ISO Registration 103) shall be designated in G2 when it is in use. It may be omitted where subsequent characters are drawn only from the basic set, or only from a standardized option.

Use of T.61 except where mandated by standards is outside the scope of these agreements. Exceptions to this rule for specific Application Service Element protocol elements must be documented in the individual chapters.

5.11.2.3.4 JIS 6226

This Japanese set is optionally supported.

Implementations supporting JIS X0208 are required to support the following two graphic sets:

6	ASCII Graphic Character Set in G0
87	Japanese Character Set JIS X0208 in G1

and optionally,

15	Japanese Katakana Character Set JIS (registration pending) in G2

These agreements are subject to verification of final text.

5.11.3 References for Character Set Text

CCITT Recommendation T.61 - 1985, "Character Repertoire and Coded Character Sets for the International Teletex Service", CCITT Red Book, Terminal Equipment and Protocols for Telematic Services, Recommendations of the T Series, International Telecommunications Union, Geneva.

DIS 8859-7 - 1987, "Information processing -- 8-bit single-byte coded graphic character sets -- Part 7: Latin/Greek alphabet", International Organization for Standardization, Geneva.

IS 2022 - 1986, "Information processing -- ISO 7-bit and 8-bit coded character sets -- Code extension techniques", International Organization for Standardization, Geneva.

IS 6429 - 1983, "Information Processing -- ISO 7-bit and 8-bit coded character sets -- Additional control functions for character-imaging devices", International organization for Standardization, Geneva.

IS 646 - 1983, "Information Processing -- ISO 7-bit coded character set for information interchange", International Organization for Standardization, Geneva.

IS 6937/1 - 1983, "Information processing -- Coded character sets for text communication -- Part 1: General introduction", International Organization for Standardization, Geneva.

IS 6937/2 - 1983, "Information processing -- Coded character sets for text communication -- Part 2: Latin alphabetic and non-alphabetic graphic characters", International Organization for Standardization, Geneva.

IS 8859-1 - 1987, "Information processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1", International Organization for Standardization, Geneva.

ISO Character Set Register - 1989, "International Register of Coded Character Sets to be Used With Escape Sequences", European Computer Manufactures Association, Geneva.

5.12 CONFORMANCE

(Refer to Stable Document.)

5.12.1 Specific ASE Requirements

(Refer to Stable Document.)

5.12.1.1 FTAM

(Refer to Stable Document.)

5.12.1.2 MHS

(Refer to Stable Document.)

5.12.1.2.1 Phase 1

(Refer to Stable Document.)

5.12.1.2.2 Phase 2, Protocol P7

(Refer to Stable Document.)

ROSE Requirements:

Operation and association classes are used as per the standard.

RTSE Requirements:

- o TWA
- o normal-mode

ACSE Requirements:

all

The use of AP-TITLE, AE-QUALIFIER, AP-INVOCATION-ID, and AE-INVOCATION-ID are prohibited; however, a receiving entity must be capable of ignoring them (if present) without refusing the connection.

Application Contexts:

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

Abstract Syntaxes:

- o "ISO 8650-ACSE1"

  Associated Transfer Syntax:
  - o "Basic Encoding of a single ASN.1 type"

Presentation Requirements:

Presentation Functional Units:

- o kernel

Presentation Contexts:

- o 2

Abstract Syntaxes:

- o ?

  Associated Transfer Syntax:

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

Session Requirements:

Session Functional Units:
- kernel
- half-duplex
- exceptions
- activity management
- minor synchronize

Version Number: 2

Maximum size of User Data parameter field: 10,240

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

5.12.1.2.3 Phase 2, Protocol P3

ROSE Requirements:
As per Phase 2, P7.

RTSE Requirements:
?

ACSE Requirements:
As per Phase 2, P7.

Application Contexts:
- "MTS-access" - mandatory

- o "MTS-reliable-access" - optional
- o "MTS-forced-access" - mandatory
- o "MTS-forced-reliable-access" - optional

Presentation Requirements:
As per Phase 2, P7.

Session Requirements:
As per Phase 2, P7.

5.12.1.2.4 Phase 2, Protocol P1

ROSE Requirements:
ROSE is not used.

RTSE Requirements:
- o Monologue
- o TWA

ACSE Requirements:
As per Phase 2, P7.

Application Contexts:
- o "MTS-transfer-protocol-1984" - mandatory
- o "MTS-transfer-protocol" - mandatory
- o "MTS-transfer" - mandatory

Presentation Requirements:
As per Phase 2, P7.

Session Requirements:
As per Phase 2, P7.

5.12.1.3 DS

(Refer to Stable Document.)

5.12.1.4 Virtual Terminal

(Refer to Stable Document.)

5.12.1.5 Network Management

5.13 REFERENCES

The following documents are referenced in these ongoing NIST

agreements on the OSI Upper Layers. Other document references may be found in the Stable Implementation Agreements for OSI Protocols.

5.13.1 ACSE

[A1] Information Processing Systems - Open Systems Interconnection - Connectionless ACSE Protocol to Provide the Connectionless-Mode ACSE Service, ISO DIS 10035: 1989-02-25 (ISO/IEC JTC1/SC21 N 3456).

5.13.2 Session Layer

[S1] Information Processing Systems - Open Systems Interconnection - Session Service Definition: Addendum 3 Covering Connectionless-Mode Session Service, ISO/DAD3 8326: 1989-02-25 (E) (ISO/IEC JTC1/SC21 N 3462).

[S2] Information Processing Systems - Open Systems Interconnection - Connectionless Session Protocol to Provide the Connectionless-Mode Session Service, ISO/DIS 9548: 1989-02-25 (E) (ISO/IEC JTC1/SC21 N 3460).

5.13.3 Presentation Layer

[P1] Information Processing Systems - Open Systems Interconnection - Presentation Service Definition: Draft Addendum 1 Covering Connectionless-Mode Presentation Service, ISO/DAD1 8822: 1989-02-25 (E) (ISO/IEC JTC1/SC21 N 3171).

[P2] Information Processing Systems - Open Systems Interconnection - Connectionless Presentation Protocol to Provide the Connectionless-Mode Presentation Service, ISO/DIS 9576: 1989-02-25 (E) (ISO/IEC JTC1/SC21 N 3172).

6. OBJECT IDENTIFIERS AND OTHER REGISTRATION ISSUES (STABLE) REGISTRATION AUTHORITY PROCEDURES FOR THE OSI IMPLEMENTATION WORKSHOP (OIW) AGREEMENTS

Editor's Note: Sections 6.1 through 6.6 contain new text. Section 6.7 (Appendix C) contains a reference to prior text in Version 2, Edition 3 Stable Document which will be reviewed for removal.

6.1 INTRODUCTION AND SCOPE

6.1.1 What is Registration?

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

Without registration authorities, chaos will result, with random name and address values being assigned to objects. Since systems would not be able to uniquely identify themselves globally, communication would become impossible. Verifying the existence of connections would become impossible; routing of protocol information would become cumbersome. For all of these reasons, registration procedures are essential in the OSI environment.

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

Hierarchy implies an inverted "treelike" structure where the number of objects increases from the "top" of the tree to the "base" of the tree. The tree may be sliced into horizontal "levels"; level one corresponds to the "top" of the tree, and the highest-numbered level corresponds to the "bottom" of the tree (or base). At the top of the tree, there is one designator that is most "powerful"; that is, it has the greatest scope of authority (largest domain). This designator assigns identifier values to objects under its authority. These objects have smaller domains than the objects immediately above. Each of these objects has a smaller scope of authority than the objects

immediately above. This process goes on continuously, moving down the tree.

Important concepts are that the scope of authority decreases as one moves down the tree, and that the number of objects increases as one moves down the tree. One authority at a specific level may crate zero, one, or many subauthorities at the next-higher level. The number of levels in such a treelike structure is arbitrary.

6.1.2 Scope

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

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

This chapter is structured as follows. Section 2 describes the information objects that need to be registered. Section 3 describes a registration procedures for OIW object identifiers. Section 4 outlines registration procedures for OSI Organization names. Appendix A lists the object identifier component values assigned to the OIW and the SIGs. Appendix B discusses object identifiers used in the 1987 and 1988 Implementation Agreements. The appendices are integral parts of this specification.

6.2 REGISTERED INFORMATION OBJECTS

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

- Application-process-titles
- Application-entity-titles
- Abstract syntaxes
- Transfer syntaxes
- Application-contexts
- MHS
  - ADMD

    - PRMD
    - Organization Names
    - Encoded information types
    - Extended body part types
    - Heading attributes
- Object Identifier values
- ASN.1 modules
- Directory
    - Relative distinguished names
    - Attribute Types
    - Attribute syntaxes
    - Object classes
    - Encryption algorithms
- VT
    - Profiles
    - Reference information objects
- Network management objects
- Network layer addresses
- System titles
- FTAM
    - Document types
    - Implementation profile types
    - Constraint sets

Not all of the above objects will be registered with the NIST Workshop. Those objects not managed by the registration authority will be managed by the appropriate registration authority under a different arc of the naming tree.

The registration authority will only administer information objects used by the OIW Implementation Agreements that are identified by the ASN.1 type OBJECT IDENTIFIER. The assignments of Identifiers and NumberForms for object identifiers is as follows:

Identifier 1	NumberForm1
iso	1
Identifier 2	**NumberForm2**
identified-organization	3
Identifier 3	**NumberForm3**
osinet	4
Identifier 4	**NumberForm4**
issuing-organization	200
Identifier 5	**NumberForm5**
	assigned (see Appendix A)

The registration authority for OSINET has assigned a unique NumberForm4 with component value 200 to the OIW, and the OIW has assigned a unique NumberForm5 to each SIG. The assigned name and

NumberForm for the OIW and for each SIG is in Appendix A. The assignment of values below Level 5 in the object identifier naming tree is the responsibility of each SIG in the OIW.

6.3 REGISTRATION PROCEDURES FOR OBJECT IDENTIFIERS

Any OIW SIG may request its Registration Officer to register an object identifier for one of its information objects. The SIG shall have been charged with the development or maintenance of the object. The registered value shall be incorporated into the appropriate OIW agreements document as a result of a positive ballot response of the plenary.

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

6.3.1 SIG Registration Authorization

An OIW SIG must be authorized by its charter and the scope of its work to submit a registration request to the OIW plenary.

6.3.2 The SRO (SIG Registration Officer)

6.3.2.1 Appointment of the SRO

The chairperson of each SIG shall appoint a member of the SIG as the SRO (SIG Registration Officer).

6.3.2.2 Duties of the SRO

The SRO is responsible for the preparation of the registration request forms for the information objects the SIG submits to the OIW plenary for approval.

The SRO is identified to the NIST-OSI Workshop plenary and will act as the liaison and correspondent for all communication between the plenary and other SIGs regarding registration of information objects.

The SRO is responsible for adhering to the procedures described in this document.

6.3.3 Requirements for Information Object Registration

6.3.3.1 Initial Registration of Information Objects

For each information object to be registered, the SIG must prepare a technical definition that describes the purpose, scope, and the unique characteristics of the information object.

For each information object to be registered, the SRO shall supply the following information:

a) assigned object identifier component values (ASN.1 NameAndNumerForm)

b) name of requesting SIG

c) dates of SIG approval and plenary vote

d) name, address, telephone/facsimile number, and e-mail address of the SRO.

Multiple information objects may be listed in one request.

The SIG's technical definition and the completed request for registration is to be submitted to the OIW Plenary for approval.

6.3.3.2 Assignment of Object Identifier Component Values

The SRO for each SIG shall register an object identifier component values for each one of its technical definitions. The NameAndNumberForm of the ObjIdComponent specified in ISO 8824/X.208 is used exclusively. This form comprises an ASN.1 identifier and, significantly, a NumberForm.

The SRO shall assign a specific numeric value to the NumberForm. To the significant root:

{ iso(1) identified-organization(3) osinet(4) oiw(200 xxxxxxxx(yy) }

(where xxxxxxxx is the identifier and yy is the NumberForm assigned by the OIW registration authority), add a SIG-assigned object identifier component value that shall be unique within the SIG register.

6.3.3.3 Rejection of Registration Request

Upon rejection of a request for registration by the Plenary, the SRO is responsible for evaluating the reason for rejection and taking the necessary action to correct omissions and administrative errors, or work with the appropriate SIG members to prepare a re-submission.

If the Request for Registration is to be withdrawn, then the SRO will notify the SIG chairperson and the secretary of the Plenary that the request has been withdrawn.

An appropriate entry is to be inserted in the minutes of the next Plenary meeting.

6.3.3.4 Registration Request Completed

Upon approval (including possible modification by the Plenary) of the information object technical definition by the Plenary, an entry for inclusion in the text and informative appendix of the appropriate chapter of the on-going OIW agreements document will be prepared.

The SRO is to announce the registration of all information objects through an entry in the minutes of the SIG and as an entry in the text and an informative appendix of the appropriate chapter of the on-going agreements.

A complete and approved request for registration is to be retained by each SRO as a permanent record of the SIG.

6.3.3.5 Changes and Revisions to the Information Object Registration

Neither the technical definition nor the request for registration may be changed or modified after registration.

The SRO will ensure that revisions of technical definitions of information objects are registered again with newly assigned object identifiers.

6.3.4 Register Maintenance

The SRO for each SIG shall keep a database that contains all the data elements described in 6.3.3.1. This same list will appear in the appropriate chapter annexes of the on-going stable agreements document.

Once registered, object identifier component values are not deleted or reassigned.

The SIG is responsible for defining the internal procedures for maintaining the SIG register. These procedures include:

a) mechanisms for maintaining the integrity of the registration data base including adequate backup

b) the design of forms (paper, electronic, or a combination of both) containing the data elements

c) the documentation of appropriate procedures to allow audits of the registration database.

6.4 Registration Procedures for OSI Organization Names

Organization names shall be assigned in the U.S. by the U.S. level registration authority. The specification of the procedures for registering organization names is "Procedures for Registering Organizational OSI Names in the United States of America." [ANSI] The registration authority for these procedures is identified by the American National Standards Institute, Inc. (ANSI). These procedures allow an organization to request the assignment of a sequentially-generated integer name and/or an alphanumeric name (supplied by the applicant). Names are recorded in the U.S.-level register.

For MHS OR Addressing, an Administrative Management Domain (ADMD) name shall be an alphanumeric name from the U.S. level register. ADMD names shall conform also to the requirements states in the Implementation Agreements (see Section xxxx). A Private Management Domain (PRMD) name shall be an alphanumeric name from the U.S. level register. PRMD names shall conform also to the requirements stated in these Implementation Agreements (see Section xxxx).

6.5 APPENDIX A: ASSIGNMENTS TO WORKSHOP ORGANIZATIONS

Name	NumberForm	
oiw	200	(Assigned to OIW by OSINET)
llsig	1	(Assigned to SIG by OIW)
nmsig	2	"
secsig	3	"
tpsig	4	"
ftamsig	5	"
mhsig	6	"
dssig	7	"
ulsig	8	"
rdasig	9	"
mmssig	10	"
odasig	11	"
vtsig	12	"
rasig	13	"

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

In the 1987 (version 1) and 1988 (version 2) of the Stable Implementation Agreements, a number of OIW-specified information objects are assigned object identifiers. These object identifiers include the following object identifier component values as a prefix:

NIST-ad-hoc OBJECT IDENTIFIER::=(1 3 9999 1)

These first four NumberForms are ambiguous. The third NumberForm values, 9999, is not (and cannot be) assigned. Consequently, use of this {1 3 9999 1} value is ambiguous and name collisions may result. OSI requires names and addresses, e.g., object identifiers, be globally unambiguous. This chapter specifies object identifier component values which are globally unambiguous. Other chapters in this document specify the correct object identifiers to be used when referencing OIW-specified information objects.

The use of the 1987 and 1988 OIW-specified object identifiers is deprecated and no longer conformant to these agreements. The object identifiers (see 1988 Section 6) listed below are impacted by these agreements:

```
{ NBS-ad-hoc abstract-syntax(2) nbs-as1(1) }

{ NBS-ad-hoc abstract-syntax(2) nbs-as2(2) }

{ NBS-ad-hoc constraint-set(4) nbs-ordered-flat(3) }

{ NBS-ad-hoc document-type(5) sequential(6) }

{ NBS-ad-hoc document-type(5) random-file(7) }

{ NBS-ad-hoc document-type(5) indexed-file(8) }

{ NBS-ad-hoc document-type(5) file-directory(9) }

nist-vte-profile OBJECT IDENTIFIER ::= { nist-ad-hoc 8 }

{ nist-vte-profile telnet-1988(0) }

{ nist-vte-profile transparent-1988(1) }
{ nist-vte-profile forms-1988(2) }

{ nist-vte-profile scroll-1988(30) }

{ nist-ad-hoc nist-vt-co(9) cotypemisc(0) }

{ nist-ad-hoc nist-vt-co(9) cotypetcco(4) }

{ nist-vt-co-misc sa(0) }

{ nist-vt-co-misc ua(1) }

{ nist-vt-co-misc st(2) }

{ nist-vt-co-misc ut(3) }

{ NBS-ad-hoc ftam-nil-ap-title(7) }
```

6.7 APPENDIX C: PRIOR TEXT

Editor's Note: Prior text is in the aligned section of Version 2, Edition 3, June 1989 of the Stable Document. This prior text is subject to review and removal.

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 2, Edition 3, June 1989.

8. MESSAGE HANDLING SYSTEMS

8.1 INTRODUCTION

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

This is an Implementation Agreement for Message Handling Systems (MHS) based on both the CCITT X.400(1988) series of Recommendations and the similar (but not identical) ISO MOTIS standard (see References). The term 'MHS' is used to refer to both sources where a distinction is unnecessary. Similarly, '1984' and '1988' are often used to distinguish between the CCITT X.400(1984) series of Recommendations and the later sources. Figure 8.1 shows the layered structure of this Agreement.

This Implementation Agreement seeks to establish a common specification which is conformant with both CCITT and ISO with a view to:

- Preventing a proliferation of incompatible communities of MHS systems which are isolated for protocol reasons,

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

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

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

- To add useful 1988 functionality (Message Store, remote UA, etc), and

- To provide a minimal conformant 1988 MHS as a firm basis for the introduction of further 1988 services and features. Subsequent versions of this Agreement will define such additional 1988 aspects as incremental enhancements.

However, it is <u>not</u> considered that the existing NIST Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems should be withdrawn at this stage and it can be anticipated that X.400(1984) implementations will continue to provide a viable alternative for applications that do <u>not</u> require the additional 1988 functionality for some time.

Interpersonal Messaging System	CCITT X.420	ISO 10021-7
Message Store	CCITT X.413	ISO 10021-5
Message Transfer System	CCITT X.411 CCITT X.419	ISO 10021-4 ISO 10021-6
Remote Operations Service Element	CCITT X.219/229	ISO 9072
Reliable Transfer Service Element	CCITT X.218/228	ISO 9066
Association Control Service Element	CCITT X.217/227	ISO 8649/50
Presentation Layer	CCITT X.216/226	ISO 8822/23
Session Layer	CCITT X.215/225	ISO 8326/27

Figure 8.1 The Layered Structure of this Implementation Agreement

8.2 SCOPE

This Agreement specifies the requirements for MHS implementations based on the 1988 MHS standards (see Figure 8.1 above).

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

(A) PRMD to PRMD,
(B) PRMD to ADMD,
(C) ADMD to ADMD,
(D) MTA to MTA (within a domain, e.g., for MTAs from different vendors),
(E) MTA to remote MS or UA, and
(F) MS to remote UA.

In case A, the PRMDs do not make use of MHS services provided by an ADMD. In cases B and C, UAs associated with an ADMD can be the source or destination for messages. Furthermore, in cases A and B, a PRMD can serve as a relay between MDs, and in cases B and C an ADMD can serve as a relay between MDs. In cases E and F, the UA is located remotely from the MTA. Figure 8.2 illustrates the interfaces to which this Agreement applies.

MHS protocols other than the Message Transfer Protocol (P1), the Message Transfer System Access Protocol (P3), the Interpersonal Messaging Protocol (P2), and the Message Store Access Protocol (P7) are beyond the scope of this Agreement. Issues arising from the use of other protocols or relating to P1 components in support of other protocols are outside the scope of this document. This Agreement

describes the minimum level of services provided at each interface shown in Figure 8.2. Provision for the use of the remaining services defined in the MHS standards is outside the scope of this document.

Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this Agreement requires the ability to exchange messages without use of bilateral agreements.

The 1988 MHS standards cover a wide and diverse range of functional areas, not all of which would be relevant to every implementation.

The initial version of this Agreement will define a minimal conformant MHS implementation which will be capable of interworking with implementations based on the CCITT X.400(1984) Recommendations as defined in Chapter 7 of the NIST Stable Implementation Agreements for OSI Protocols (Version 2 Edition 3, June 1989), and will additionally define the minimum set of requirements which are necessary to provide useful remote UA and/or Message Store services, independent of the level (i.e. 1984 or 1988) of the P1 implementation.

In order to achieve a more precise definition of conformance requirements according to the functionality supported by an implementation (and additionally to facilitate future enhancement of this initial specification), the concept of 'Functional Groups' has been introduced. Figure 8.3 shows the Functional Groups covered by this Agreement and indicates where they are defined in this Chapter. Only the MT and IPM Kernel Functional Groups have to be supported for minimal conformance to this initial Agreement.

There are two conformance levels defined for the MT Kernel in these Agreements:

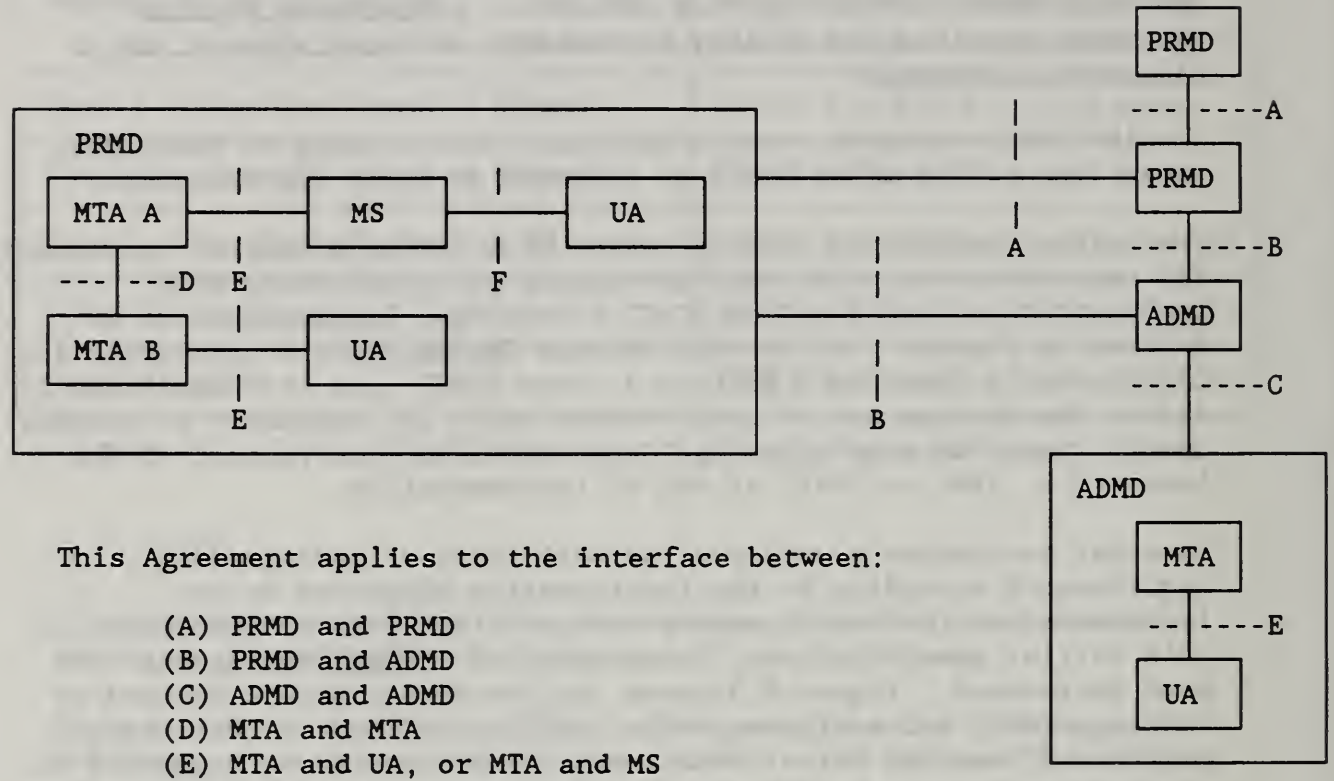
- A class 'A' MT Kernel implementation supports transfer (i.e., relaying) only;

- A class 'B' MT kernel implementation supports submission, delivery and transfer (including relaying).

  Note: This does not imply support for the P3 protocol.

In addition, the UAs and MTAs will require access to directory and routing services. Except insofar as they must be capable of providing addressing and routing as described in Section 8.9, these services and associated protocols are not described by this Agreement (see Chapter 11 - Directory Services).

PRMD = Private Management Domain
ADMD = Administration Management Domain



This Agreement applies to the interface between:

(A) PRMD and PRMD
(B) PRMD and ADMD
(C) ADMD and ADMD
(D) MTA and MTA
(E) MTA and UA, or MTA and MS
(F) UA and MS

Figure 8.2 Scenario Definition

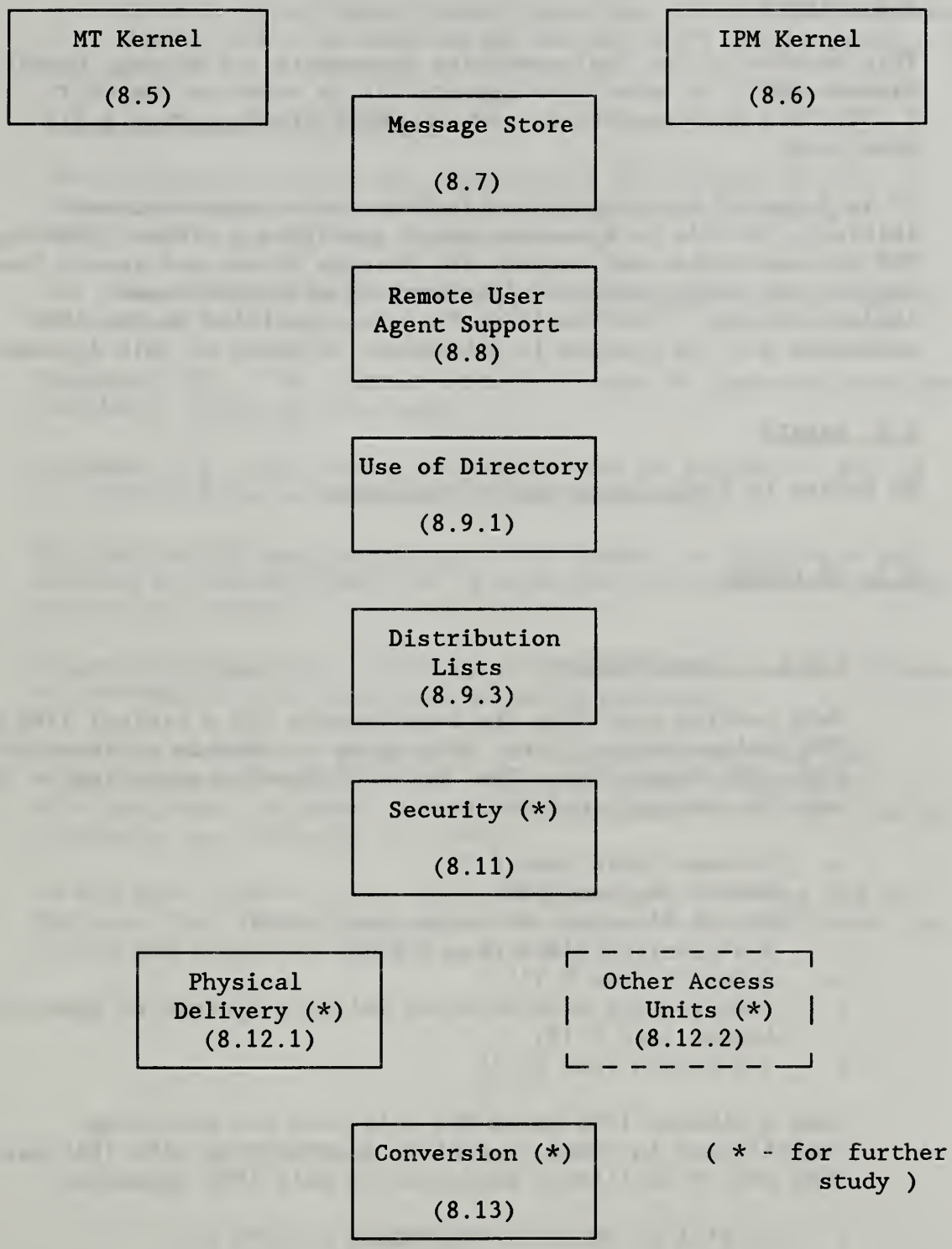


Figure 8.3 MHS Functional Groups

8.3 STATUS

This version of the Implementation Agreements for Message Handling Systems (MHS) is under development. It is based on the CCITT X.400(1988) Recommendations and ISO MOTIS (10021, parts 1-7) standards.

It is intended that the Stable Implementation Agreements will initially include an Agreement which specifies a minimal 1988-based MHS implementation and support for Message Stores and remote User Agents, and which addresses interworking with 1984-based implementations. The remaining features specified in the 1988 standards will be covered in subsequent versions of this Agreement.

8.4 ERRATA

No Errata to Stable material at this time.

8.5 MT KERNEL

8.5.1 Introduction

This section specifies the requirements for a minimal 1988-based MTS implementation (i.e., MTA) which is capable of interworking with 1984-based MTAs. The 'base' MT Service specified in this section does not include:

- Message Store (see 8.7)
- Remote UA (see 8.8)
- Use of Directory Services (see 8.9.1)
- Distribution Lists (see 8.9.3)
- Security (see 8.11)
- Interworking with Physical Delivery systems or Specialized Access (see 8.12)
- Conversion (see 8.13)

Such a minimal 1988-based MTA will have the following capabilities in order to achieve interworking with 1984-based MTAs and to facilitate migration to full 1988 operation:

- It will be protocol-conformant to 1988 P1;

- It will downgrade 1988 P1 to 1984 P1 when relaying to 1984-based MTAs, as specified in Annex B of X.419 (see 8.5.5);

- It will relay the contents of 1988 P1 messages unchanged, even when relaying to 1984-based MTAs;

o It will support both 'normal mode' and 'X.410 mode' protocol stacks (i.e., as required by ISO and CCITT respectively).

8.5.2 Elements of Service

This section specifies the requirements for support of MT Elements of Service by an MTA conforming to the MT Kernel Functional Group of this Agreement.

The classification scheme for support of Elements of Service is as follows:

Mandatory (M) - the Element of Service must be supported and made available to the service user;

Optional (O) - the Element of Service may be supported, but is not required for conformance to this Agreement;

Not Defined/Not Applicable (-) - the Element of Service is not defined by this Agreement or is otherwise not applicable in the particular context;

To Be Determined (*) - the support classification for the Element of Service has yet to be determined (temporary).

The requirements for support of MT Elements of Service for origination and reception and (where relevant) relaying are distinguished. Elements of Service which are new in the 1988 MHS standards are indicated as (1988).

An MTA must support those Basic MT Elements of Service and MT Optional User Facilities defined in clause 19 of X.400(1988) as listed and qualified in Tables 8.1 and 8.2 below.

Table 8.1 MT Kernel : Basic MT Elements of Service

Element of Service	Origination	Reception	Relaying
Access Management	M[1]	M[1]	-
Content Type Indication	M	M	-
Converted Indication	M	M	M
Delivery Time Stamp Indication	-	M	-
Message Identification	M	M	-
Non-delivery Notification	M	M	M
Original Encoded Information Types Indication	M	M	-
Submission Time Stamp Indication	M	M	-
User/UA Capabilities Registration (1988)	-	M[1]	-

Notes: 1) A local matter in the case of co-located UA/MTA and/or MS/MTA configurations.

Table 8.2 MT Kernel : MT Service Optional User Facilities

Element of Service	Origination	Reception	Relaying
Alternate Recipient Allowed	M	M[2]	-
Alternate Recipient Assignment	-	O[2]	-
Conversion Prohibition	M	M	M
Conversion Prohibition in Case of Loss of Information (1988)	O	O	O
Deferred Delivery	M[3]	O	O
Deferred Delivery Cancellation	M	-	-
Delivery Notification	M	M	-
Disclosure of Other Recipients	M	M	M
DL Expansion History Indication	-	M	-
Explicit Conversion	O	O	O
Grade of Delivery Selection	M	M	M
Hold for Delivery	-	M[1]	-
Implicit Conversion	O	O	O
Latest Delivery Designation (1988)	O	O	O
Multi Destination Delivery	M	M	M
Originator Requested Alternate Recipient (1988)	O	O	-
Prevention of Non-delivery Notification	O	-	-
Probe	M	M	M
Redirection Disallowed by Originator (1988)	O	O	-
Redirection of Incoming Messages (1988)	-	O	-
Requested Delivery Method (1988)	M	M	-
Restricted Delivery (1988)	-	O	-
Return of Content	O	O	O

Notes: 1) A local matter in the case of co-located UA/MTA and/or MS/MTA configurations.

2) If Alternate Recipient Assignment is supported on reception, then support of Alternate Recipient Allowed is Mandatory on reception; otherwise, support of Alternate Recipient Allowed is Optional on reception.

3) Support of this MT Element of Service is Mandatory for conformance reasons, but may be performed as a local matter to the originating MTA.

8.5.3 MTS Transfer Protocol (P1)

The requirements for support of MTS Transfer Protocol (P1) elements are detailed in Section 8.17.1 (Appendix A).

Support of MTS Transfer Protocol application contexts by an MTA is classified as follows:

mts-transfer-protocol-1984	Mandatory
mts-transfer-protocol	Mandatory
mts-transfer	Mandatory

Use of the underlying services to support these application contexts is specified in Section 8.14.

8.5.4 Intra Domain Considerations

To be determined.

Note: It has yet to be determined whether this section will be confined to intra-PRMD issues only or will cover all intra-domain implementation considerations.

8.5.5 Downgrading Issues

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, and

- Internal Trace Information - to be determined.

8.6 IPM KERNEL

8.6.1 Introduction

This section specifies the requirements for a minimal 1988-based IPMS implementation (i.e., UA) which is capable of interworking with 1984-based UAs. The 'base' IPM Service specified in this section does not include:

- Message Store (see 8.7)
- Remote UA (see 8.8)
- Use of Directory Services (see 8.9.1)
- Distribution Lists (see 8.9.3)
- Security (see 8.11)
- Interworking with Physical Delivery systems or Specialized Access (see 8.12)

Such a minimal 1988-based UA will have the following capabilities in order to achieve interworking with 1984-based UAs and to facilitate migration to full 1988 operation:

- It will continue to support content type P2 (encoded as integer 2) on origination and reception;

- It will support receipt of P2 (encoded as integer 22);

- It may originate P2 (22), but the guidelines specified in clause 20.2 of X.420(1988) are to be followed, i.e. the content type shall be encoded as P2 (2) unless 1988 P2 protocol elements are present.

8.6.2 Elements of Service

This section specifies the requirements for support of IPM Elements of Service by a UA conforming to the IPM Kernel Functional Group of this Agreement.

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

The requirements for support of IPM Elements of Service for origination and reception are distinguished. Elements of Service which are new in the 1988 MHS standards are indicated as (1988).

A UA must support those Basic IPM Elements of Service and IPM Optional User Facilities defined in Clause 19 of X.400(1988) as listed and qualified in Tables 8.3 and 8.4 below.

Table 8.3 IPM Kernel : Basic IPM Elements of Service

Element of Service	Origination	Reception
Access Management	M[1]	M[1]
Content Type Indication	M	M
Converted Indication	-	M
Delivery Time Stamp Indication	-	M
IP-message Identification	M	M
Message Identification	M	M
Non-delivery Notification	M	-
Original Encoded Information Types Indication	M	M
Submission Time Stamp Indication	M	M
Typed Body	M	M
User/UA Capabilities Registration (1988)	-	M[1]

Notes: 1) In the case of a co-located UA/MTA, the method and extent to which this Element of Service is provided is a local matter; it is not necessarily testable in the absence of support for the P3 protocol.

Table 8.4 IPM Kernel : IPM Service Optional User Facilities

Element of Service	Origination	Reception
Alternate Recipient Allowed	O	O/M[2]
Alternate Recipient Assignment	-	O[2]
Authorizing Users Indication	O	M
Auto-forwarded Indication	O	M
Blind Copy Recipient Indication	O	M
Body Part Encryption Indication	O	M
Conversion Prohibition	M	M
Conversion Prohibition in Case of Loss of Information (1988)	O	O
Cross Referencing Indication	O	M
Deferred Delivery	M	-
Deferred Delivery Cancellation	O	-
Delivery Notification	M	-
Disclosure of Other Recipients	O	M
DL Expansion History Indication	-	M
Expiry Date Indication	O	M
Explicit Conversion	O	-
Forwarded IP-message Indication	O	M
Grade of Delivery Selection	M	M
Hold for Delivery	-	O/M[1]
Implicit Conversion	-	O
Importance Indication	O	M
Incomplete Copy Indication (1988)	O	O
Language Indication (1988)	O	M
Latest Delivery Designation (1988)	O	-
Multi Destination Delivery	M	-
Multi-part Body	O	M
Non-receipt Notification Request	O	M
Obsoleting Indication	O	M
Originator Indication	M	M
Originator Requested Alternate Recipient (1988)	O	-
Prevention of Non-delivery Notification	O	-
Primary and Copy Recipients Indication	M	M
Probe	O	-
Receipt Notification Request Indication	O	O
Redirection Disallowed by Originator (1988)	O	-
Redirection of Incoming Messages (1988)	-	O
Reply Request Indication	O	M
Replying IP-message Indication	M	M
Requested Delivery Method (1988)	M	-
Restricted Delivery (1988)	-	O
Return of Content	O	-
Sensitivity Indication	O	M
Subject Indication	M	M

Notes: 1) Mandatory in the case of a remote UA (where the MTA does not support MSs) or a remote UA/MS.

2) If Alternate Recipient Assignment is supported on reception, then support of Alternate Recipient Allowed is Mandatory on reception; otherwise, support of Alternate Recipient Allowed is Optional on reception.

8.6.3 Interpersonal Messaging Protocol (P2)

The requirements for support of Interpersonal Messaging Protocol (P2) elements are detailed in Section 8.17.2 (Appendix A).

8.6.4 Body Part Support

This section specifies the requirements for support of IPM body part types by a UA conforming to this Agreement.

The classification scheme for support of IPM body part types is as defined in Section 8.5.2.

The requirements for support of IPM body part types for origination and reception are distinguished. Body part types which are new in the 1988 MHS standards are indicated as (1988).

A UA must support those IPM body part types defined in Annex C of X.420(1988) as listed and qualified in Table 8.5 below. If an implementation supports a particular body part type for reception, it should also be able to support that body part type for reception if it is part of a forwarded message.

Any body part type that is supported on reception must be supported as integer encoding and as object identifier (externally-defined) encoding.

All body parts with integer-encoded identifiers in the range 0 up to and including 16K-1 are legal and must be relayed. Body part integer-encoded identifiers corresponding to X.121 country codes should be interpreted as described in Note 2 of Figure 8.4. These privately-defined body part types are specified as an interim measure to provide backward compatibility with 1984 MHS implementations. For interworking between UAs based on the 1988 (or later) MHS standards, it is strongly recommended that the externally-defined body part be used instead.

Table 8.5 IPM Kernel : Body Part Types

Body Part Type		Origination	Reception
IA5Text		M	M
Voice		O	O
G3Facsimile		O	O
G4Class1	(TIF0)	O	O
Teletex		O	O
Videotex		O	O
Encrypted		O	O
Message	(ForwardedIPMessage)	O	M
MixedMode	(TIF1)	O	O
BilaterallyDefined	(Unidentified)	O	O
NationallyDefined		O	O
ExternallyDefined	(1988)	O	M[1]
PrivatelyDefined	(see Figure 8.4)	O	O

Notes: 1) Any body part type that is supported on reception as integer encoding must also be supported as object identifier encoding.

```
BodyPart ::=  CHOICE {
        ia5-text                [0] IA5TextBodyPart,
                                  .
        externally-defined     [15] ExternallyDefinedBodyPart,
                                  .
                              [234] UKBodyParts,
                                  .
                              [310] USABodyParts,
                                  .       }

    Where UKBodyParts and USABodyParts are defined as:

        SEQUENCE {BodyPartNumber, ANY}
        BodyPartNumber ::= INTEGER

Note 1) The undefined bit in P1 EncodedInformationTypes must
        be set when a message contains a privately defined
        body part.  Each UA that expects such body parts
        should include undefined in the set of deliverable
        EncodedInformationTypes it registers with the MTA.

Note 2) Body part numbers are interpreted relative to the body
        part type in which they are used.  NIST registers body
        part numbers for privately-defined formats within the
        United States.
```

Figure 8.4 Privately-Defined Body Parts

8.7 MESSAGE STORE

8.7.1 Introduction

This section specifies Agreements for implementation of the Message Store (MS) Functional Group. The MS is responsible for accepting delivery of messages on behalf of a single end-user, and retaining the messages until the end-user's UA is able to retrieve them. Message submission and some administration services are provided via "pass-through" to the MTS. Figure 8.5 illustrates the logical relationship of the MS to the UA and MTS.

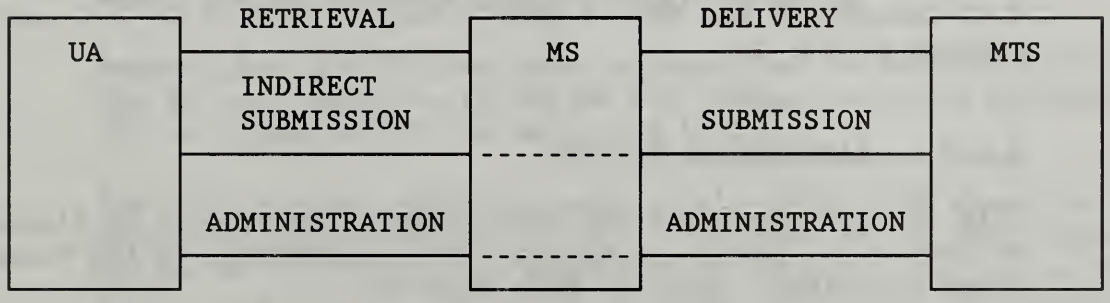


Figure 8.5 Message Store Model

The Agreements in this section specify the Message Store's use of the retrieval, delivery, and administration services. Agreements on submission services are specified in Section 8.8, which describes support for the remote UA. Agreements on the use of message management services defined in ISO 10021-5 are for future study.

The goal of the Agreements in this section is to define the minimal set of features which are necessary to provide useful Message Store services, independent of the MTA implementation version (i.e., 1984 or 1988).

8.7.2 Scope

The scope of the Agreements in this section is depicted in Figure 8.6 below, and is confined to the services and protocols between the boundaries shown (marked with asterisks). Requirements for the UA and MTA are addressed only to the extent that they affect the Message Store and remote User Agent services and protocols. This reflects the additional services required at the UA to support MS access and at the MTA to support a remote MS.

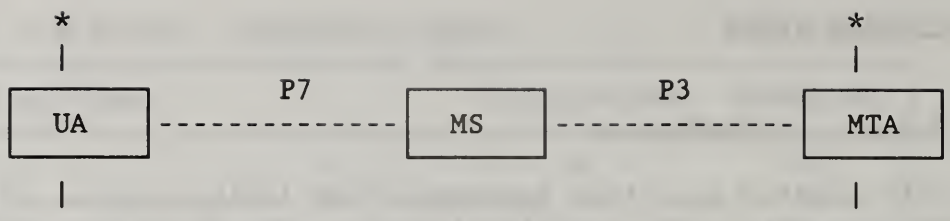


Figure 8.6 Scope of Message Store Agreements

The UA, MS and MTA configuration is not restricted; any of these components may be co-located, although they are depicted as logically separate. In the case of a co-located UA and MS, a proprietary interface may be used instead of P7. In the case of a co-located MS and MTA, a proprietary interface may be used instead of P3.

8.7.3 Elements of Service

This section specifies the requirements for support of Elements of Service to provide a Message Store conforming to the Message Store Functional Group of this Agreement.

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

Support for Elements of Service is specified both for the Message Store itself and for the User Agent.

Table 8.6 Message Store : Elements of Service

Element of Service	UA	MS
Stored Message Deletion	M	M
Stored Message Fetching	M	M
Stored Message Listing	M	M
Stored Message Summary	M	M
Stored Message Alert	O	O
Stored Message Auto Forward	O	O

8.7.4 Attribute Types

Requirements for support of the attributes used in the Message Store are detailed in Sections 8.17.5 and 8.17.6 (Appendix A). Section 8.17.5 specifies support for the General Attributes of the Message Store, while Section 8.17.6 specifies support for the IPM Message Store Attributes.

There are two classes of support for General Attributes in the Message Store.

The minimal MS only requires support for those General Attributes specified as Mandatory (M) in Section 8.17.5. The intent of the minimal MS is to support the use of the MS as a continuously available, reliable device (such as a spooling entity) for receiving, storing and forwarding messages and reports.

The standard MS requires support for all Mandatory (M) and Supported (H) attributes specified in Section 8.17.5. This form of the MS provides more flexible access to particular General Attributes of the stored messages and must be used in situations where the MS supports Interpersonal Messaging.

Support for IPM MS Attributes is specified in Section 8.17.6. The MS must support all IPM MS Attributes specified as Mandatory (M) or Supported (H) in Section 8.17.6.

User Agents must support access to all attributes specified as Mandatory (M) in Section 8.17.5. IPM User Agents must support access to all attributes specified as Mandatory (M) in Sections 8.17.5 and 8.17.6. UA access to other attributes is optional.

8.7.5 Pragmatic Constraints for Attribute Types

To be determined.

8.7.6 Implementation of the MS with 1984 Systems

While the Message Store is part of the 1988 MHS standards, implementation of MS services with a 1984 MTA is possible. In order to interoperate with other 1984 MHS systems, implementations with this configuration must adhere to the following guidelines:

- The UA must generate 1984 P2 PDUs;
- The UA must identify the content protocol as integer 2 to the MS;
- The MS must be co-located with the MTA unless 1988 P3 support is provided on the 1984 MTA as well.

To meet these guidelines, the UA may be implemented as follows:

- The UA could conform to X.420(1984), with 1988 UA extensions for utilizing the MS services;

- The UA could be a 1988 UA with restrictions on protocol elements generated and by identifying the content type as integer 2 rather than 22. No 1988-specific elements should be generated.

Details of the interface between the 1988 MS and the 1984 MTA when co-located are beyond the scope of these Agreements.

8.7.7 MS Access Protocol (P7)

The requirements for support of MS Access Protocol (P7) elements by an MS and a remote MS-user are detailed in Section 8.17.4 (Appendix A).

The requirements for support of MS Access Protocol (P7) application contexts by an MS and an MS-user are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the additional requirement that an MS-user must at least support the ms-access application context, as follows:

	MS	MS-user
ms-access	Mandatory	Mandatory
ms-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in Section 8.14.

8.7.8 MTS Access Protocol (P3)

The requirements for support of MTS Access Protocol (P3) elements by an MTA and an MS where the MS is not co-located with the MTA are detailed in Section 8.17.3 (Appendix A).

The requirements for support of MTS Access Protocol (P3) application contexts by an MTA and an MS in such a scenario are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the additional requirement that a remote MS must at least support the mts-access and mts-forced-access application contexts, as follows:

	MTA	MS
mts-access	Mandatory	Mandatory
mts-forced-access	Mandatory	Mandatory
mts-reliable-access	Optional	Optional
mts-forced-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in Section 8.14.

8.8 REMOTE USER AGENT SUPPORT

8.8.1 Introduction

This section specifies Agreements for implementation of the Remote User Agent Functional Group, i.e. for support of an IPM UA that is not co-located with its MTA. Support of other classes of UA is for further study.

The goal of the Agreements in this section is to define the minimal set of features which are necessary to provide useful remote User Agent services, independent of the MTA implementation version (i.e., 1984 or 1988).

8.8.2 Scope

The scope of the Agreements in this section is depicted in Figure 8.7, and is confined to the services and protocols between the boundaries shown (marked with asterisks). Requirements for the UA and MTA are addressed only to the extent that they affect the remote User Agent services and protocols. Access to a Message Store by a remote User Agent is covered in Section 8.7.

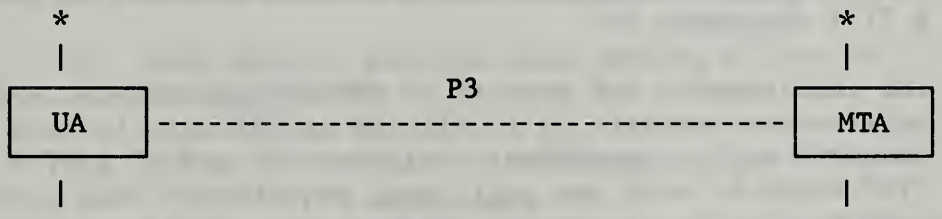


Figure 8.7 Scope of Remote User Agent Agreements

8.8.3 Elements of Service

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

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

Support for Elements of Service is specified both for the MT Service and for the IPM Service, and is in addition to the support requirements specified in Sections 8.5 and 8.6 if this Functional Group is supported.

Table 8.7 Remote User Agent Support: MT Elements of Service

Element of Service	Origination	Reception
Access Management	M	M
Hold for Delivery	-	M
User/UA Capabilities Registration	-	M

Table 8.8 Remote User Agent Support: IPM Elements of Service

Element of Service	Origination	Reception
Access Management	M	M
Hold for Delivery	-	M
User/UA Capabilities Registration	-	M

8.8.4 MTS Access Protocol (P3)

The requirements for support of MTS Access Protocol (P3) elements by an MTA and an MTS-user (whether UA or UA/MS) where the MTS-user is not co-located with the MTA are detailed in Section 8.17.3 (Appendix A).

The requirements for support of MTS Access Protocol (P3) application contexts by an MTA and an MTS-user in such a scenario are as specified in clauses 6.1 and 10.1 of X.419(1988) (ISO 10021-6) with the additional requirement that a remote MTS-user must at least support the mts-access and mts-forced-access application contexts, as follows:

	MTA	MTS-user
mts-access	Mandatory	Mandatory
mts-forced-access	Mandatory	Mandatory
mts-reliable-access	Optional	Optional
mts-forced-reliable-access	Optional	Optional

Use of the underlying services to support these application contexts is specified in Section 8.14.

8.9 NAMING, ADDRESSING & ROUTING

8.9.1 MHS Use of Directory

8.9.1.1 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 to be used in submitting messages for delivery by the MTS. The MTS may also use directory service elements to obtain information to be used in routing messages.

Some functional requirements of directories have been identified and are listed below:

- Verify the existence of a directory name;
- Return the O/R address that corresponds to the directory name presented;
- Determine whether the directory name presented denotes a user or a distribution list;
- Return a list of the members of a distribution list;
- When given a partial name, return a list of possibilities;
- Allow users to scan directory entries;
- Allow users to scan directory entries selectively;
- Return the capabilities of the entity referred to by the directory or O/R name;
- Provide maintenance functions to keep the directory up-to-date.

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

This section identifies and specifies the Use of Directory Functional Group, which is intended to cover all issues relating to the use by an MHS implementation of Directory Services which conform to the Agreements in Chapter 11.

8.9.1.2 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 8.9 Use of Directory : MT Elements of Service

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

Table 8.10 Use of Directory : IPM Elements of Service

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

8.9.2 Use of Names & Addresses

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

With the exception of the intra-domain connection agreements, these agreements make no attempt to recommend a standard practice for electronic mail addressing.

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

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

Further, each PRMD may adopt naming and addressing schemes wherein the user view may take a form entirely different from the ORName attributes specified in this Agreement, and each PRMD may

have one user view for the originator form and another for the recipient form, and perhaps other forms of user addressing. In some cases (e.g., receipt notification) these user forms must be preserved within the constraints of this Agreement. However, mapping between one PRMD user form to another PRMD user form, via the MHS ORName attributes of this Agreement, is outside the scope of this Agreement.

8.9.3 Distribution Lists

8.9.3.1 Introduction

This section identifies and specifies the Distribution Lists Functional Group, which is intended to cover all issues relating to the support of distribution lists by an MHS implementation.

8.9.3.2 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 both for the MT Service and for the IPM Service.

Table 8.11 Distribution Lists : MT Elements of Service

Element of Service	Origination	Reception
DL Expansion History Indication	*	*
DL Expansion Prohibited	*	*
Use of Distribution List	*	*

Table 8.12 Distribution Lists : IPM Elements of Service

Element of Service	Origination	Reception
DL Expansion History Indication	*	*
DL Expansion Prohibited	*	*
Use of Distribution List	*	*

8.10 MHS MANAGEMENT

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 8.13 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 8.14 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 8.15 Physical Delivery : MT Elements of Service

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

Table 8.16 Physical Delivery : IPM Elements of Service

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

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 MHS implementation.

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 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 8.17 Conversion : MT Elements of Service

Element of Service	Origination	Reception
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)

The P1 protocol is mapped onto the Reliable Transfer Service Element (RTSE) either in X.410-1984 mode or in normal mode, as specified in Section 8.5.3. In X.410-1984 mode, the RTSE makes direct use of the services provided by the Session Layer, as specified in Chapter 5 (Upper Layers) of the Stable Implementation Agreements, Version 2, Edition 3, June 1989. In normal mode, the RTSE makes use of the services provided by the Association Control Service Element (ACSE) and Presentation Layer, as defined in Chapter 5 (Upper Layers) of these Agreements.

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

The P3 and P7 protocols make use of the services provided by the Remote Operations Service Element (ROSE), Association Control Service Element (ACSE), Presentation Layer, and, optionally, the Reliable Transfer Service Element (RTSE), as defined in Chapter 5 (Upper Layers) of these Agreements. It is recommended that RTSE be used for recovery purposes when the implementation uses a Transport Class other than 4.

8.15 ERROR HANDLING

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

8.15.1 MPDU Encoding

8.15.2 Contents

8.15.3 Envelope

8.15.4 Reports

8.16 CONFORMANCE

8.16.1 Introduction

8.16.2 Configuration Options

MHS implementations may be configured as any single or multiple occurrence or combination of MTA, MS and UA, as illustrated in Figure 8.8. It is not intended to restrict the types of system that may be configured for conformance to these Agreements (although it is equally recognized that not all configuration types may be commercially viable).

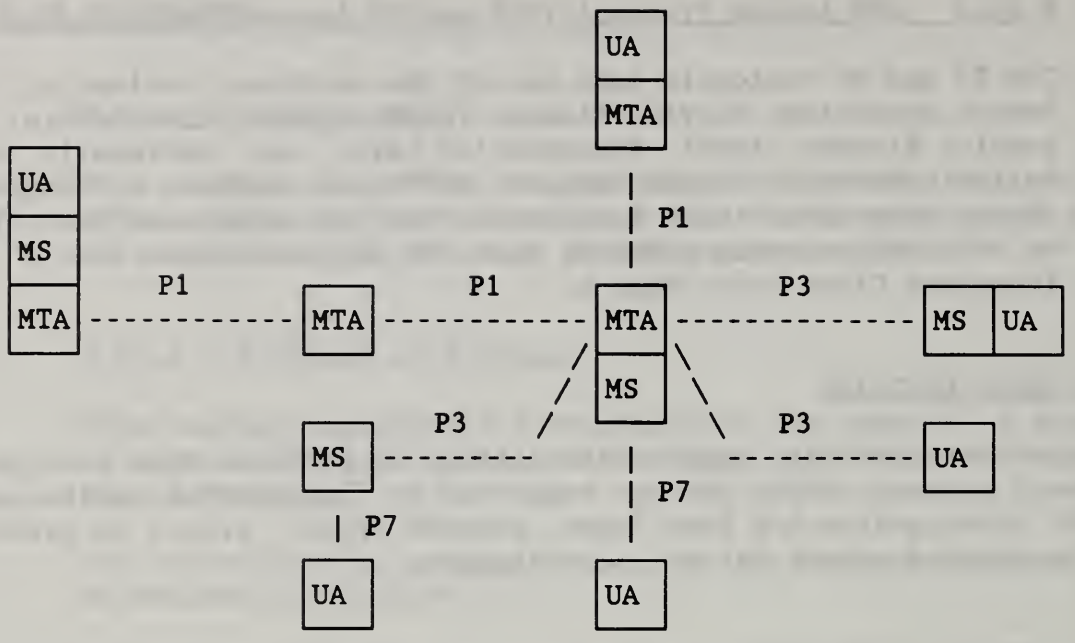


Figure 8.8 Configuration Options

8.16.3 Definition of Conformance

8.16.4 Conformance Requirements

8.17 APPENDIX A: MHS PROTOCOL SPECIFICATIONS

The following tables specify the requirements for support of MHS protocol elements for conformance to this Agreement. It should be noted that the tables specify minimum support for conformance to the relevant Kernel functional groups and where appropriate also specify enhanced support requirements where one or more further functional groups are claimed. All element support is subject to further review and may be upgraded in later versions of this Agreement.

The protocol support classification scheme used in this version of this Agreement is described below. However, it should be noted that the scheme is currently under review both within the NIST X.400 SIG and in the EWOS/ETSI MHS groups and is likely to be revised for later versions of this Agreement.

The classification of support for a protocol element specifies the requirements for implementations conforming to this Agreement to be able to generate, receive and process that protocol element, as appropriate. The classification of support for each protocol element is relative to that for its containing element. Where subelements within a containing element are not listed, then their support classification shall be assumed to be that of the containing element.

Where the range of values to be supported for an element is not specified, then all values defined in the base standard shall be supported.

Mandatory (M) - implementations conforming to this Agreement shall generate this element in all information objects in which, according to the base standards, it shall occur; receiving implementations shall process this element appropriately, and shall regard its absence as a protocol violation unless otherwise specified in the base standards;

Generatable (G) - implementations conforming to this Agreement shall be able to generate this protocol element, but it does not necessarily have to be present in every information object generated (conditions for generation are as specified in the base standards or as otherwise indicated in this Agreement); receiving implementations shall process this element appropriately if it is present;

Supported (H) - implementations conforming to this Agreement may optionally be capable of generating this protocol element, but are not required to do so; receiving implementations shall, however, process this element appropriately if it is present;

Unsupported (X) - implementations conforming to this Agreement may optionally be capable of generating this protocol element, but should not expect any specific action or processing by a receiving implementation except as required to observe criticality indication and any such use is outside the scope of this Agreement; receiving implementations conforming to this Agreement are similarly not required to be able to process this element other than to observe any criticality indication, but must at least be able to relay the semantics of this element where appropriate; the absence of this element should not be assumed by a receiving implementation to convey any significance.

To Be Determined (*) - the support classification for this protocol element has yet to be determined.

8.17.1 MTS Transfer Protocol (P1)

	Support		Comments/References
	Class B MT Kernel	Class A MT Kernel	
MTS-APDU			
message	G	H	
envelope	M	M	MessageTransferEnvelope
content	M	M	See P2 - else undefined
probe	G	H	ProbeTransferEnvelope
report	G	G	
envelope	M	M	ReportTransferEnvelope
content	M	M	ReportTransferContent
MessageTransferEnvelope			
message-identifier	M	M	MTSIdentifier
originator-name	M	M	ORName
original-encoded-information-types	G	X	EncodedInformationTypes
content-type	M	M	
built-in	G	X	
external	H	X	
content-identifier	H	X	
priority	G	H	All values
per-message-indicators	G	H	
disclosure-of-recipients	H	H	
implicit-conversion-prohibited	G	H	
alternate-recipient-allowed	G	X	
content-return-request	X	X	
deferred-delivery-time	X	X	
per-domain-bilateral-information	X	X	PerDomainBilateralInfo
trace-information	M	M	TraceInformation
extensions	G	G	ExtensionField
recipient-reassignment-prohibited	X	X	
dl-expansion-prohibited	H	H	
conversion-with-loss-prohibited	H	H	
latest-delivery-time	X	X	See X.411, 14.1.1 note 2
originator-return-address	X	X	
originator-certificate	X	X	
content-confidentiality-algorithm-identifier	X	X	
message-origin-authentication-check	X	X	
message-security-label	X	X	
content-correlator	X	X	
dl-expansion-history	H	H	DLExpansionHistory
internal-trace-information	G	G	InternalTraceInfo
PerRecipientMessageTransferFields	M	M	
recipient-name	M	M	ORName

originally-specified-recipient-number	M	M	
per-recipient-indicators	M	M	
explicit-conversion	X	X	
extensions	H	H	ExtensionField
originator-requested-alternate-recipient	X	X	
requested-delivery-method	G	H	
physical-forwarding-prohibited	X	X	
physical-forwarding-address-request	X	X	
physical-delivery-modes	X	X	
registered-mail-type	X	X	
recipient-number-for-advice	X	X	
physical-rendition-attributes	X	X	
physical-delivery-report-request	X	X	
message-token	X	X	
content-integrity-check	X	X	
proof-of-delivery-request	X	X	
redirection-history	H	H	
ProbeTransferEnvelope			
probe-identifier	M	M	MTSIdentifier
originator-name	M	M	ORName
original-encoded-information-types	G	X	EncodedInformationTypes
content-type	M	M	
built-in	G	X	
external	H	X	
content-identifier	H	X	
content-length	G	X	
per-message-indicators	G	H	
disclosure-of-recipients	X	X	
implicit-conversion-prohibited	G	H	
alternate-recipient-allowed	G	X	
content-return-request	X	X	
per-domain-bilateral-information	X	X	PerDomainBilateralInfo
trace-information	M	M	TraceInformation
extensions	G	G	ExtensionField
recipient-reassignment-prohibited	X	X	
dl-expansion-prohibited	H	H	
conversion-with-loss-prohibited	X	X	
originator-certificate	X	X	
message-security-label	X	X	
content-correlator	X	X	
probe-origin-authentication-check	X	X	
dl-expansion-history	H	H	DLExpansionHistory

internal-trace-information	G	G	InternalTraceInfo
PerRecipientProbeTransferFields	M	M	
recipient-name	M	M	ORName
originally-specified-recipient-number	M	M	
per-recipient-indicators	M	M	
explicit-conversion	X	X	
extensions	H	H	ExtensionField
originator-requested-alternate-recipient	X	X	
requested-delivery-method	G	H	
physical-rendition-attributes	X	X	
redirection-history	H	H	
ReportTransferEnvelope			
report-identifier	M	M	MTSIdentifier
report-destination-name	M	M	ORName
trace-information	M	M	TraceInformation
extensions	G	G	ExtensionField
message-security-label	X	X	
originator-and-DL-expansion-history	G	X	OriginatorAndDL ExpansionHistory
reporting-DL-name	X	X	
reporting-MTA-certificate	X	X	
report-origin-authentication-check	X	X	
internal-trace-information	G	G	InternalTraceInfo
ReportTransferContent			
subject-identifier	M	M	MTSIdentifier
subject-intermediate-trace-information	G	G	TraceInformation
original-encoded-information-types	G	G	EncodedInformationTypes
content-type	G	G	
built-in	G	G	
external	G	G	
content-identifier	G	G	
returned-content	H	X	
additional-information	X	X	
extensions	H	H	ExtensionField
content-correlator	H	H	
PerRecipientReportTransferFields	M	M	
actual-recipient-name	M	M	ORName
originally-specified-recipient-number	M	M	
per-recipient-indicators	M	M	
last-trace-information	M	M	
arrival-time	M	M	
converted-encoded-information-types	G	G	EncodedInformationTypes
report	M	M	

delivery	G	X	
message-delivery-time	M	M	
type-of-MTS-user	G	X	All values = H
non-delivery	G	G	
non-delivery-reason-code	M	M	
non-delivery-diagnostic-code	H	H	
originally-intended-recipient-name	G	G	ORName
supplementary-information	X	X	
extensions	G	G	ExtensionField
redirection-history	G	G	RedirectionHistory
physical-forwarding-address	X	X	
recipient-certificate	X	X	
proof-of-delivery	X	X	

Common Data Types

EncodedInformationTypes			
built-in-encoded-information-types	M	M	
non-basic-parameters	X	X	
external-encoded-information-types	H	H	
MTSIdentifier			
global-domain-identifier	M	M	GlobalDomainIdentifier
local-identifier	M	M	
PerDomainBilateralInfo			
country-name	M	M	
administration-domain-name	M	M	DomainName
private-domain-identifier	G	G	DomainName (only encoded as SEQ if both present)
bilateral-information	M	M	
TraceInformation			
TraceInformationElement	G	G	
global-domain-identifier	M	M	GlobalDomainIdentifier
domain-supplied-information	M	M	
arrival-time	M	M	
routing-action	M	M	
relayed	G	G	
rerouted	H	H	
attempted-domain	H	H	GlobalDomainIdentifier
deferred-time	H	H	
converted-encoded-information-types	H	H	EncodedInformationTypes
other-actions	H	H	
redirected	H	H	
dl-operation	H	H	

ExtensionField			
type	M	M	
criticality	H	H	
for-submission	X	X	
for-transfer	G	G	
for-delivery	G	G	
value	M	M	
DLExpansionHistory			
DLExpansion	M	M	
ORAddressAndOptionalDirectory Name	M	M	ORName
dl-expansion-time	M	M	
InternalTraceInfo			
InternalTraceInformationElement	M	M	
global-domain-identifier	M	M	GlobalDomainIdentifier
mta-name	M	M	
mta-supplied-information	M	M	
arrival-time	M	M	
routing-action	M	M	
relayed	G	G	
rerouted	H	H	
attempted			
mta	H	H	
domain	H	H	GlobalDomainIdentifier
deferred-time	H	H	
other-actions	H	H	
redirected	H	H	
dl-operation	H	H	
OriginatorAndDLExpansionHistory			
originator-or-dl-name	M	M	
origination-or-expansion-time	M	M	
RedirectionHistory			
Redirection	M	M	
intended-recipient-name	M	M	
ORAddressAndOptionalDirectory Name	M	M	ORName
redirection-time	M	M	
redirection-reason	M	M	
ORName			
address	M		
standard-attributes	M		
country-name	G		CountryName
administration-domain-name	G		DomainName
network-address	G		
terminal-identifier	G		
private-domain-name	G		DomainName
organization-name	G		

```
   numeric-user-identifier       G
   personal-name                 G
    surname                      M
    given-name                   G
    initials                     G
    generation-qualifier         G
   organizational-unit-names     G
    OrganizationUnitName         G
  domain-defined-attributes      G
   DomainDefinedAttribute        G
    type                         M
    value                        M
  extension-attributes           H              ExtensionAttribute
   common-name                   H
   teletex-common-name           H
   teletex-organization-name     H
   teletex-personal-name         H
   teletex-organizational-unit-
       names                     H
   teletex-domain-defined-
       attributes                H
   pds-name                      H
   physical-delivery-country-name H
   postal-code                   H
   physical-delivery-office-name H
   physical-delivery-office-number H
   extension-OR-address-
       components                H
   physical-delivery-personal-
       name                      H
   physical-delivery-
       organization-name         H
   extension-physical-delivery-
       address-components        H
   unformatted-postal-address    H
   street-address                H
   post-office-box-address       H
   poste-restante-address        H
   unique-postal-name            H
   local-postal-attributes       H
   extended-network-address      H
   terminal-type                 H
 directory-name                  X

ExtensionAttribute
 extension-attribute-type        M
 extension-attribute-value       M

GlobalDomainIdentifier
 country-name                    M              CountryName
 administration-domain-name      M              DomainName
 private-domain-identifier       G              DomainName
```

```
CountryName
 x121-dcc-code                H
 iso-3166-alpha2-code         G

DomainName
 numeric                      H
 printable                    G
```

8.17.2 Interpersonal Messaging Protocol (P2)

	Support		Comments/References
	Minimum	Enhanced	
InformationObject			
ipm	G		IPM
ipn	G		IPN
IPM			
heading	M		
this-IPM	M		IPMIdentifier
originator	G		ORDescriptor
authorizing-users	H		RecipientSpecifier
primary-recipients	G		RecipientSpecifier
copy-recipients	G		RecipientSpecifier
blind-copy-recipients	H		RecipientSpecifier
replied-to-IPM	G		IPMIdentifier
obsoleted-IPMs	H		IPMIdentifier
related-IPMs	H		IPMIdentifier
subject	G		See Note 1
expiry-time	H		
reply-time	H		
reply-recipients	H		ORDescriptor
importance	H		
sensitivity	H		
auto-forwarded	H		
extensions	H		HeadingExtension
incomplete-copy	X		
languages	H		
body	M		BodyPart
IPN			
subject-ipm	M		
ipn-originator	G		ORDescriptor
ipm-preferred-recipient	G		ORDescriptor
conversion-eits	H		EncodedInformationTypes
non-receipt-fields	G		
non-receipt-reason	M		
discard-reason	G		
auto-forward-comment	H		
returned-ipm	X		See Note 2
receipt-fields	H		
receipt-time	M		
acknowledgment-mode	H		
suppl-receipt-info	X		
HeadingExtension			
type	M		
value	M		
IPMIdentifier			
user	H		

user-relative-identifier	M	
ORDescriptor		
formal-name	H	ORName - see Note 3
free-form-name	H	
telephone-number	H	
RecipientSpecifier		
recipient	M	ORDescriptor
notification-requests	H	
reply-requested	H	
BodyPart		
ia5-text	G	
parameters	M	
repertoire	H	Support of ITA2 is for
data	M	for further study
voice	X	
parameters	M	
data	M	
g3-facsimile	X	
parameters	M	
number-of-pages	H	
non-basic-parameters	H	
data	M	
g4-class1	X	
teletex	X	
parameters	M	
number-of-pages	X	
telex-compatible	X	
non-basic-parameters	X	
data	M	
videotex	X	
parameters	M	
syntax	H	
data	M	
encrypted	X	
parameters	M	
data	M	
message	H	
parameters	M	
delivery-time	H	
delivery-envelope	H	See P3 OtherMessage DeliveryFields
data	M	
mixed-mode	X	
bilaterally-defined	X	
nationally-defined	X	
externally-defined	H	
parameters	M	
data	M	

Notes:

1) The ability to generate the maximum size subject is not required.

2) May only be included if specifically requested by the originator.

3) The ORName should be specified wherever possible.

8.17.3 MTS Access Protocol (P3)

Note: The support classifications for the IPM UA, MS and MTA below indicate the minimum level of support required by implementations conforming to these Agreements, and should not be misconstrued as a redefinition of any of the MHS application contexts.

	Support			Comments/References
	IPM UA	MS	MTA	
Operations				
MTSBind	M	M	M	MTSBind
MTSUnbind	M	M	M	
MSSE				
message-submission	M	M	M	MessageSubmission
probe-submission	O	M	M	ProbeSubmission
cancel-deferred-delivery	O	M	M	CancelDeferredDelivery
submission-control	M	M	O	SubmissionControl
MDSE				
message-delivery	M	M	M	MessageDelivery
report-delivery	M	M	M	ReportDelivery
delivery-control	O	O	M	DeliveryControl
MASE				
register	O	M	M	Register
change-credentials				
(MTS to MTSuser)	M	M	O	ChangeCredentials
(MTSuser to MTS)	O	M	M	ChangeCredentials

Note: A Message Store must pass through all MSSE and MASE operations unaltered.

Arguments/Results				
MTSBind				
ARGUMENT				
initiator-name	M	M	M	
mTS-user	G	G	H	
mTA	H	H	G	
isMessageStore	G	G	H	
messages-waiting	X	X	X	
initiator-credentials	M	M	M	
simple	G	G	G	
strong	X	X	X	
RESULT				
responder-name	M	M	M	
mTS-user	G	G	H	

mTA HHG				
isMessageStore	G	G	H	
messages-waiting	X	X	X	
responder-credentials	M	M	M	
simple	G	G	G	
strong	X	X	X	
MessageSubmission				
ARGUMENT				
envelope	M	M	M	MessageSubmission Envelope
content	M	M	M	
RESULT				
message-submission-identifier	M	M	M	See P1 MTSIdentifier
message-submission-time	M	M	M	
content-identifier	H	H	G	
extensions	X	X	X	
originating-MTA-certificate	X	X	X	
proof-of-submission	X	X	X	
ProbeSubmission				
ARGUMENT				
envelope	M	M	M	ProbeSubmission Envelope
RESULT				
probe-submission-identifier	M	M	M	See P1 MTSIdentifier
probe-submission-time	M	M	M	
content-identifier	H	H	G	
CancelDeferredDelivery				
ARGUMENT				
message-submission-identifier	M	M	M	See P1 MTSIdentifier
SubmissionControl				
ARGUMENT				
controls	M	M	M	See Note 1
restrict	H	H	X	
permissible-operations	H	H	X	
permissible-maximum-content-length	H	H	X	
permissible-lowest-priority	H	H	X	
permissible-security-context	X	X	X	
RESULT				
waiting	M	M	M	See Note 2
waiting-operations	X	X	H	
waiting-messages	X	X	H	
waiting-content-types	X	X	H	
waiting-encoded-information-types	X	X	H	See P1 Encoded InformationTypes

MessageDelivery				
ARGUMENT				
envelope	M	M	M	MessageDeliveryEnvelope
content	M	M	M	
RESULT				
recipient-certificate	X	X	X	
proof-of-delivery	X	X	X	
ReportDelivery				
ARGUMENT				
envelope	M	M	M	ReportDeliveryEnvelope
returned-content	H	H	X	
DeliveryControl				
ARGUMENT				
controls	M	M	M	See Note 3
restrict	X	X	H	
permissible-operations	X	X	H	
permissible-maximum-content-length	X	X	H	
permissible-lowest-priority	X	X	H	
permissible-content-types	X	X	H	
permissible-encoded-information-types	X	X	H	See P1 Encoded InformationTypes
permissible-security-context	X	X	X	
RESULT				
waiting	M	M	M	See Note 4
waiting-operations	H	H	X	
waiting-messages	H	H	X	
waiting-content-types	H	H	X	
waiting-encoded-information-types	H	H	X	See P1 Encoded InformationTypes
Register				See Note 5
ARGUMENT				
user-name	X	X	X	See X.411, 8.4.1.1.1.1
user-address	X	X	X	
deliverable-encoded-information-types	X	H	H	See P1 Encoded InformationTypes
deliverable-maximum-content-length	X	H	H	
default-delivery-controls	X	X	X	
restrict	X	X	X	
permissible-operations	X	X	X	
permissible-maximum-content-length	X	X	X	
permissible-lowest-priority	X	X	X	
permissible-content-types	X	X	X	
permissible-encoded-information-types	X	X	X	See P1 Encoded InformationTypes
deliverable-content-types	X	H	H	

labels-and-redirections	X	X	X	
user-security-label	X	X	X	
recipient-assigned-alternate-recipient	X	X	X	
ChangeCredentials (MTS to MTSuser)				
ARGUMENT				
old-credentials	M	M	M	
simple	H	H	X	
strong	X	X	X	
new-credentials	M	M	M	
simple	H	H	X	
strong	X	X	X	
ChangeCredentials (MTSuser to MTS)				
ARGUMENT				
old-credentials	M	M	M	
simple	X	X	H	
strong	X	X	X	
new-credentials	M	M	M	
simple	X	X	H	
strong	X	X	X	
MessageSubmissionEnvelope				See Note 6
originator-name	M	M	M	See P1 ORName
original-encoded-information-types	G	H	H	See P1 Encoded InformationTypes
content-type	M	M	M	
built-in	X	H	H	
external	X	H	H	
content-identifier	X	H	H	
priority	G	H	H	All values
per-message-indicators	G	H	H	
disclosure-of-recipients	X	H	H	
implicit-conversion-prohibited	G	H	H	
alternate-recipient-allowed	G	H	H	
content-return-request	X	H	H	
deferred-delivery-time	G	H	H	
extensions	G	H	H	
recipient-reassignment-prohibited	X	H	H	
dl-expansion-prohibited	G	H	H	
conversion-with-loss-prohibited	X	H	H	
latest-delivery-time	X	H	H	
originator-return-address	X	H	H	
originator-certificate	X	X	X	
content-confidentiality-algorithm-identifier	X	X	X	
message-origin-authentication-check	X	X	X	
message-security-label	X	X	X	
proof-of-submission-request	X	X	X	

content-correlator	X	H	H	
forwarding-request	X	H	H	MS Abstract Service only
PerRecipientMessageSubmission Fields	M	M	M	
recipient-name	M	M	M	See P1 ORName
originator-report-request	M	M	M	
explicit-conversion	X	H	H	
extensions	G	H	H	
originator-requested-alternate-recipient	X	H	H	
requested-delivery-method	G	H	H	
physical-forwarding-prohibited	X	H	H	
physical-forwarding-address-request	X	H	H	
physical-delivery-modes	X	H	H	
registered-mail-type	X	H	H	
recipient-number-for-advice	X	H	H	
physical-rendition-attributes	X	H	H	
physical-delivery-report-request	X	H	H	
message-token	X	X	X	
content-integrity-check	X	X	X	
proof-of-delivery-request	X	X	X	
ProbeSubmissionEnvelope				See Note 6
originator-name	M	M	M	See P1 ORName
original-encoded-information-types	G	H	H	See P1 Encoded InformationTypes
content-type	M	M	M	
built-in	X	H	H	
external	X	H	H	
content-identifier	X	H	H	
content-length	G	H	H	
per-message-indicators	G	H	H	
implicit-conversion-prohibited	G	H	H	
alternate-recipient-allowed	X	H	H	
extensions	G	H	H	
recipient-reassignment-prohibited	X	H	H	
dl-expansion-prohibited	G	H	H	
conversion-with-loss-prohibited	X	H	H	
originator-certificate	X	X	X	
message-security-label	X	X	X	
content-correlator	X	H	H	
probe-origin-authentication-check	X	X	X	
PerRecipientProbeSubmission Fields	M	M	M	
recipient-name	M	M	M	See P1 ORName
originator-report-request	M	M	M	
explicit-conversion	X	H	H	
extensions	G	H	H	

originator-requested-alternate-recipient	X	H	H	
requested-delivery-method	G	H	H	
physical-rendition-attributes	X	H	H	
MessageDeliveryEnvelope				See Note 7
message-delivery-identifier	M	M	M	See P1 MTSIdentifier
message-delivery-time	M	M	M	
other-fields	M	M	M	
content-type	M	M	M	
built-in	H	H	G	
external	H	H	G	
originator-name	M	M	M	See P1 ORName
original-encoded-information-types	H	H	G	See P1 Encoded InformationTypes
priority	H	H	G	All values
delivery-flags	H	H	G	
implicit-conversion-prohibited	H	H	G	
other-recipient-names	H	H	G	See P1 ORName
this-recipient-name	M	M	M	See P1 ORName
originally-intended-recipient-name	H	H	G	See P1 ORName
converted-encoded-information-types	H	H	G	See P1 Encoded InformationTypes
message-submission-time	M	M	M	
content-identifier	H	H	G	
extensions	H	H	G	
conversion-with-loss-prohibited	H	H	G	
requested-delivery-method	H	H	G	
physical-forwarding-prohibited	H	H	G	
physical-forwarding-address-request	H	H	G	
physical-delivery-modes	H	H	G	
registered-mail-type	H	H	G	
recipient-number-for-advice	H	H	G	
physical-rendition-attributes	H	H	G	
physical-delivery-report-request	H	H	G	
originator-return-address	H	H	G	
originator-certificate	X	X	X	
message-token	X	X	X	
content-confidentiality-algorithm-identifier	X	X	X	
content-integrity-check	X	X	X	
message-origin-authentication-check	X	X	X	
message-security-label	X	X	X	
proof-of-delivery-request	X	X	X	
redirection-history	H	H	G	
dl-expansion-history	H	H	G	

ReportDeliveryEnvelope				See Note 7
subject-submission-identifier	M	M	M	See P1 MTSIdentifier
content-identifier	H	H	G	
content-type	H	H	G	
built-in	H	H	G	
external	H	H	G	
original-encoded-information-types	H	H	G	See P1 Encoded InformationTypes
extensions	H	H	G	
message-security-label	X	X	X	
content-correlator	H	H	G	
originator-and-DL-expansion-history	H	H	G	See P1 OriginatorAndDL ExpansionHistory
reporting-DL-name	H	H	G	
reporting-MTA-certificate	X	X	X	
report-origin-authentication-check	X	X	X	
PerRecipientReportDeliveryFields	M	M	M	
actual-recipient-name	M	M	M	See P1 ORName
report	M	M	M	
delivery	H	H	G	
message-delivery-time	M	M	M	
type-of-MTS-user	H	H	G	
non-delivery	H	H	G	
non-delivery-reason-code	M	M	M	
non-delivery-diagnostic-code	H	H	G	
converted-encoded-information-types	H	H	G	See P1 Encoded InformationTypes
originally-intended-recipient-name	H	H	G	See P1 ORName
supplementary-information	H	H	G	
extensions	H	H	G	
redirection-history	H	H	G	See P1 Redirection History
physical-forwarding-address	H	H	G	
recipient-certificate	X	X	X	
proof-of-delivery	X	X	X	

Notes:

1) The MTS-user may interpret any restriction as simply withhold 'all' submissions.

2) No explicit action needs to be taken by the MTA.

3) The MTA may interpret any restriction as simply withhold 'all' deliveries.

4) No explicit action needs to be taken by the MTS-user.

5) The Register operation may be performed locally (see X.411). Although not required for the UA for conformance, it is considered to be a useful service and support is recommended.

6) The action to be taken by a submitting MTA is as defined in X.411 (ISO 10021-4). In the absence of any specific processing requirements for a particular element in a submission envelope, the action to be taken is simply the faithful mapping of such element to the corresponding element of the appropriate transfer envelope.

7) The action to be taken by a delivering MTA is as defined in X.411 (ISO 10021-4). In the absence of any specific processing requirements for a particular element in a delivery envelope, the action to be taken is simply the creation of such element from the corresponding element of the appropriate transfer envelope.

8.17.4 MS Access Protocol (P7)

Operations	Support IPM UA	Support MS	Comments/References
MSBind	M	M	MSBind
MSUnbind	M	M	
MSSE			
message-submission	M	M	See P3 MessageSubmission
probe-submission	O	M	See P3 ProbeSubmission
cancel-deferred-delivery	O	M	See P3 CancelDeferred Delivery
submission-control	M	M	See P3 SubmissionControl
MASE			
register	O	M	See P3 Register
change-credentials (MS to UA)	M	M	See P3 ChangeCredentials
change-credentials (UA to MS)	O	M	See P3 ChangeCredentials
MRSE			
summarize	M	M	Summarize
list	M	M	List
fetch	M	M	Fetch
delete	M	M	Delete
register-ms	O	M	Register-MS
alert	O	O	Alert
Arguments/Results			
MSBind			
ARGUMENT			
MSBindArgument	M	M	
initiator-name	M	M	
initiator-credentials	M	M	
simple	G	H	
strong	X	X	
security-context	X	X	
fetch-restrictions	X	H	
allowed-content-types	X	H	
allowed-EITs	X	H	
maximum-content-length	X	H	
MS-configuration-request	X	H	
RESULT			
MSBindResult	M	M	
responder-credentials	M	M	
simple	H	G	
strong	X	X	
available-auto-actions	H	G	
available-attribute-types	H	G	
alert-indication	H	X	
content-types-supported	H	G	

Summarize			
ARGUMENT			
SummarizeArgument	M	M	
information-base-type	X	H	InformationBase
selector	M	M	Selector
summary-requests	X	H	
RESULT			
SummarizeResult	M	M	
next	H	G	
count	M	M	
span	H	G	
lowest	M	M	
highest	M	M	
summaries	H	G	
absent	H	G	
present	H	G	
type	M	M	
value	M	M	
count	M	M	
List			
ARGUMENT			
ListArgument	M	M	
information-base-type	X	H	InformationBase
selector	M	M	Selector
requested-attributes	G	H	AttributeSelection
RESULT			
ListResult	M	M	
next	H	G	
requested	H	G	EntryInformation
Fetch			
ARGUMENT			
FetchArgument	M	M	
information-base-type	X	H	InformationBase
item	M	M	
search	G	H	Selector
precise	G	H	
requested-attributes	G	H	AttributeSelection
RESULT			
FetchResult	M	M	
entry-information	H	G	EntryInformation
list	H	G	
next	H	G	

Delete			
ARGUMENT			
DeleteArgument	M	M	
information-base-type	X	H	InformationBase
items	M	M	
selector	G	H	Selector
sequence-numbers	G	H	
RESULT			
DeleteResult	M	M	
Register-MS			
ARGUMENT			
Register-MSArgument	M	M	
auto-action-registrations	X	X	
type	M	M	
registration-identifier	G	H	
registration-parameter	M	M	See auto action registration parameters
auto-action-deregistrations	X	X	
type	M	M	
registration-identifier	G	H	
list-attribute-defaults	G	H	
fetch-attribute-defaults	G	H	
change-credentials	G	H	
old-credentials	M	M	
new-credentials	M	M	
user-security-labels	X	X	
RESULT			
Register-MSResult	M	M	
Alert			
ARGUMENT			
AlertArgument	M	M	
alert-registration-identifier	M	M	
new-entry	H	G	EntryInformation
RESULT			
AlertResult	M	M	

<u>Auto Action Registration Parameters</u>

AutoForwardRegistrationParameter			
filter	X	H	Filter
auto-forward-arguments	M	M	
originator-name	M	M	
content-identifier	X	H	
priority	X	H	
per-message-indicators	X	H	See P3
deferred-delivery-time	X	H	
extensions	X	H	See P3
per-recipient-fields	M	M	
recipient-name	M	M	
originator-report-request	M	M	

explicit-conversion	X	H	
extensions	X	H	See P3
delete-after-auto-forwarding	X	H	
other-parameters	X	H	See Note 1
auto-forwarding-comment	X	H	
cover-note	X	H	
this-ipm-prefix	X	H	
AutoAlertRegistrationParameter			
filter	X	H	Filter
alert-addresses	X	X	
address	M	M	
alert-qualifier	X	X	
requested-attributes	X	H	AttributeSelection

Notes:

1) The specified syntax of other-parameters is for IPMS use only - see X.413 clause 12.1 and X.420 clause 19.4.

Common Data Types

AttributeSelection			
type	M	M	
from	X	H	
count	X	H	
AttributeValueAssertion			
type	M	M	
value	M	M	
EntryInformation			
sequence-number	M	M	
attributes	H	G	
type	M	M	
values	M	M	
Filter			
item	G	H	FilterItem
and	X	X	
or	X	X	
not	X	X	
FilterItem			
equality	G	H	AttributeValueAssertion (Support is X if ORname)
substrings	X	X	
type	M	M	
strings	M	M	
greater-or-equal	X	H	AttributeValueAssertion
less-or-equal	X	H	AttributeValueAssertion
present	X	H	

InformationBase			
stored-messages	G	H	
inlog	X	X	
outlog	X	X	
Range			
sequence-number-range	X	H	
from	X	H	
to	X	H	
creation-time-range	X	H	
from	X	H	
to	X	H	
Selector			
child-entries	X	H	
range	X	H	Range
filter	X	H	Filter
limit	X	H	
override	X	H	

8.17.5 Message Store General Attribute Support

General Attributes	MS Support	Functional Group(s)
child-sequence-numbers	M	
content	M	
content-confidentiality-algorithm-identifier	X	
content-correlator	H	
content-integrity-check	X	
content-length	H	
content-returned	H	
content-type	M	
conversion-with-loss-prohibited	H	
converted-eits	H	
creation-time	M	
delivered-eits	H	
delivery-flags	H	
dl-expansion-history	H	
entry-status	M	
entry-type	M	
intended-recipient-name	H	
message-delivery-envelope	M	
message-delivery-identifier	H	
message-delivery-time	H	
message-origin-authentication-check	X	
message-security-label	X	
message-submission-time	H	
message-token	X	
original-eits	H	
originator-certificate	X	
originator-name	H	
other-recipient-names	H	
parent-sequence-number	M	
per-recipient-report-delivery-fields	M	
priority	H	
proof-of-delivery-request	X	
redirection-history	H	
report-delivery-envelope	M	
reporting-dl-name	X	
reporting-mta-certificate	X	
report-origin-authentication-check	X	
security-classification	X	
sequence-number	M	
subject-submission-identifier	M	
this-recipient-name	H	

Note: Enhanced MS support for optional Functional Groups is for further study. Attributes which are relevant to these areas are currently specified as Unsupported.

8.17.6 Message Store IPM Attribute Support

IPM Attribute	MS Support
Summary Attributes:	
ipm-entry-type	H
ipm-synopsis	H
Heading Attributes:	
authorizing-users	H
auto-forwarded	H
blind-copy-recipients	H
copy-recipients	H
expiry-time	H
heading	M
importance	H
incomplete-copy	X
languages	H
nrn-requestors	H
obsoleted-ipms	H
originator	H
primary-recipients	H
related-ipms	H
replied-to-ipm	H
reply-recipients	H
reply-requestors	H
reply-time	H
rn-requestors	H
sensitivity	H
subject	H
this-ipm	M
Body Attributes:	
bilaterally-defined-body-parts	X
body	M
encrypted-body-parts	X
encrypted-data	X
encrypted-parameters	X
extended-body-part-types	X
g3-facsimile-body-parts	X
g3-facsimile-data	X
g3-facsimile-parameters	X
g4-class1-body-parts	X
ia5-text-body-parts	H
ia5-text-data	X
ia5-text-parameters	X
message-body-parts	H
message-data	X

message-parameters	X
mixed-mode-body-parts	X
nationally-defined-body-parts	X
teletex-body-parts	X
teletex-data	X
teletex-parameters	X
videotex-body-parts	X
videotex-data	X
videotex-parameters	X
voice-body-parts	X
voice-data	X
voice-parameters	X

Notification Attributes:

acknowledgment-mode	H
auto-forward-comment	H
conversion-eits	H
discard-reason	H
ipm-preferred-recipient	H
ipn-originator	H
non-receipt-reason	H
receipt-time	H
returned-ipm	X
subject-ipm	M
suppl-receipt-info	X

8.18 APPENDIX B: INTERPRETATION OF ELEMENTS OF SERVICE

8.19 APPENDIX C: RECOMMENDED PRACTICES

It is not necessary to follow the recommended practices when claiming conformance to this Agreement.

8.19.1 EDI

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 current 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 2, Edition 3, June 1989.

10. ISO FILE TRANSFER, ACCESS AND MANAGEMENT PHASE 3

Editor's Note: The "NBS" designation remains in effect for document types, abstract syntaxes, and constraint sets defined in all FTAM agreements up to 1/1/89. After 1/1/89, any new functionality references the "NIST" designation. This is to reflect the change in identifying organization from "NBS" to "NIST".

10.1 INTRODUCTION

This section contains Implementors Agreements based on ISO 8571 File Transfer, Access and Management. These Agreements define enhancements to the Stable FTAM Implementation Agreements for OSI Protocols, Version 1, Edition 1, December 1987 (FTAM Phase 2 Agreements, NBS 500-150), including all their subsequent Errata changes as specified in Version 2, Edition 3 (NIST Special Publication 500-162).

Therefore it is assumed that the reader is familiar both with the contents of the base standard ISO 8571 and its underlying layers, and also with the above-mentioned NIST FTAM Phase 2 specifications.

Phase 2 Agreements define six Implementation Profiles which are T1, T2, T3, A1, A2, and M1. In order to avoid ambiguity when referring to these Implementation Profiles the above designations will apply only to Phase 2 functionality, references to Phase 3 enhanced Implementation Profiles will be by the addition of a'.3', i.e. T1.3, T2.3, T3.3, A1.3, A2.3, and M1.3.

10.2 SCOPE AND FIELD OF APPLICATION

These Phase 3 Agreements specify additional functionality to the FTAM Phase 2 Agreements. These additional functions include:

- Further specifications of document types,
- Specification for Restart Data Transfer and Recovery functional units,
- Specification of FADU Locking functional unit, and
- More details on Access Control and Concurrency Control.

All Phase 2 systems are upward compatible to a Phase 3 system and can therefore interwork with it, if the additional functions are negotiated out (e.g. use of Recovery) or not used for the interconnection (e.g. additional features for document types).

10.3 STATUS

These FTAM Phase 3 Agreements are at working paper status, reflecting the results from the FTAM SIG Meeting, June 13-15, 1989. They will become stable by September 1989.

10.4 ERRATA

10.5 CONFORMANCE

In addition to the specific requirements specified in the following subsections, conformance to this Phase 3 specification requires

- conformance to ISO 8571
- conformance to Phase 2 FTAM

10.5.1 Conformance for Access Profiles

The access Profiles A1.3 and A2.3 do not include the requirement for transferring files using the File Transfer service class.

10.6 ASSUMPTIONS

FTAM Phase 3 Agreements specify additional functionality to the Implementation Profiles T1, T2, T3, A1, A2, and M1 as defined in the FTAM Phase 2 Agreements. So all definitions and requirements for these Implementation Profiles apply also to the Phase 3 Agreements.

10.7 FILESTORE AGREEMENTS

10.7.1 Document Types

In addition to the Phase 2 Document Type Agreements the document types FTAM-4 (see ISO 8571-2, Annex-B) and NBS-10, NBS-11, NBS-12 (see Appendix C) are defined for optional support.

Table 10.1 gives the support levels for all document types with respect to the Implementation Profiles.

For FTAM-1, FTAM-2, FTAM-3 and FTAM-4 the supported parameter values for <universal class number> and <string significance> respectively are listed. Other values are outside the scope of these Agreements. No restriction or minimum requirement is

Table 10.1 Implementation Profiles and Document Types
(a) FTAM-1 Through FTAM-4

Implementation Profile (Note 1)	Document Type	Universal Class Number (Notes 1, 3, 4, 5)	String Significance
T1.3, T2.3, T3.3, A1.3, A2.3	FTAM-1	Graphic String (25)	'variable' 'fixed'
		VisibleString (26)	'variable' 'fixed'
		GeneralString (27)	'not-significant'
		IA5String (22)	'not-significant'
T2.3, T3.3, A1.3, A2.3	FTAM-2	GraphicString (25)	'not-significant'
		VisibleString (26)	'not-significant'
		[GeneralString (27)]	'not-significant'
		[IA5String (22)]	'not-significant'
T1.3, T2.3, T.3.3, A1.3, A2.3	FTAM-3	-	'not-significant'
[T2.3], [T3.3], [A1.3], [A2.3]	FTAM-4	-	'not-significant'

Table 10.1 Implementation Profiles and Document Types
(b) NBS-6 Through NBS-11

Implementation Profile (Note 1)	Document Type	Universal Class Number	String Significance
[T2.3], T3.3, [A1.3], A2.3	NBS-6		
[T2.3], T3.3, [A1.3], A2.3	NBS-7		
[T2.3], T3.3 [A1.3], A2.3	NBS-8		
[T1.3], [T2.3], [T3.3]	NBS-9		
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-10		
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-11		

Table 10.1 Implementation Profiles and Document Types
(c) NBS-12

Implementation Profile (Note 1)	Document Type	Universal Class Number	Character-Set Escape Sequences as defined for Reg. Numbers C0 G0 G1	String-Significance
[T2.3], [T3.3] [A1.3], [A2.3]	NBS-12 See Note 6	IA5String [22]	(parameter absent)	'variable' 'fixed'
		GraphicString[25]	(parameter absent)	'variable' 'fixed'
		GraphicString[25]	- 6 100	'variable' 'fixed'
		VisibleString[26]	(parameter absent)	'variable' 'fixed'
		GeneralString[27]	(parameter absent)	'variable' 'fixed'
		GeneralString[27]	1 6 100	'variable' 'fixed'

Notes:

1. Brackets around a Profile designator or a parameter value indicate that the respective document type or parameter value is optionally supported in this Implementation Profile.

2. The support level for document types in Implementation Profile M1.3 depends on the T- or A-Implementation Profile, in conjunction with which M1.3 is implemented.

3. The support for IA5 String is the ISO 646, IRV G0 character set and the ISO 646, IRV C0 set.

4. The minimum level of support for Graphic String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 sets, and ISO 646, IRV C0 character set.

5. The minimum level of support for General String is the ISO 646, IRV G0 character set and the 8859-1 G0 and G1 sets, and ISO 646, IRV C0 character set.

6. See below:

   If the Character-Set parameter is absent, the following defaults apply:

Universal-Class-Number	Default Registration Numbers		
	C0	G0	G1
IA5String [22]	1	2	-
GraphicString [25]	-	2	-
VisibleString [26]	-	2	-
GeneralString [27]	1	2	-

Character-Sets and Escape Sequences:

Registration Number	Content	Escape Sequence
1	C0 set of ISO 646	ESC 2/1 4/0
2	ISO 646, IRV	-
6	ISO 646, USA Version-X 3.4 - 1968 (Left-hand part of ISO 8859-1)	ESC 2/8 4/2
100	Right-hand part of Latin Alphabet No 1 ISO 8859-1, ECMA-94	ESC 2/13 4/1

In addition to the Phase 2 FADU Identity Agreements the following is specified:

> For the document type NBS-11 used in conjunction with the Transfer service class or the Transfer and Management service class, the support of the FADU identities of 'current', 'next', 'previous' and 'end' is outside the scope of these Agreements.

10.7.3 Access Control Attribute

The location field of access control element is optionally supported. It is the implementor's choice which combinations of fields in an access control element are supported. The ACE combination should be stated in the PICS.

10.8 PROTOCOL AGREEMENTS

10.8.1 Functional Units

For FTAM Phase 3 implementations Recovery and Restart Data Transfer are optionally supported.

FADU locking is optionally supported for Implementation Profiles A1.3 and A2.3.

When the M1.3 Implementation Profile is implemented, the Enhanced-file-management functional unit may be combined with any of the service classes 'transfer' and 'access' as defined for the Implementation Profiles T1.3, T2.3, A1.3, if the corresponding Profiles are implemented.

10.8.2 Implementation Information Parameter

In addition to the Agreements as specified for FTAM Phase 2, Section 9.12 (NIST SP 500-162), the following value is defined

NBS-Phase 3.

10.8.3 F-Check

In order to maximize interoperability, implementations of FTAM service providers should not restrict the amount of data transmitted between successive F-CHECK requests to a single quantity. Variations in the amount of data transmitted between checkpoints may be required to accommodate differences in real end systems supporting FTAM Virtual Filestores and/or in the communications media underlying FTAM associations. It is required that all FTAM implementations are able to receive at least one PSDU between checkpoints.

10.8.4 Error Recovery

Procedures for Class I, II and III errors are defined and supported for FTAM Phase 3 implementations. It is the implementor's choice whether to handle class I errors using F-RESTART PDUs or whether to use the class II error procedure.

10.8.4.1 Docket Handling

When a class III error occurs, the length of time a docket is maintained is determined by the local system. Recovery from a class III error is only possible as long as both end systems maintain the docket.

It is also a local decision how many dockets can be maintained simultaneously.

10.8.4.2 Parameters for Error Recovery

- The semantics of the <FTAM quality of service> parameter is as defined in ISO 8571, including the local knowledge of FERPM.

- No minimum requirement for the <checkpoint window> parameter or the checkpoint size is defined.

- For the <recovery mode> parameter of F-OPEN, the values 'none' and 'at-start-of-file' are supported. The value 'at-any-active-checkpoint' is optionally supported. If recovery mode 'at-start-of-file' is negotiated, no F-CHECK shall be issued. When recovering at the start of the file, the <recovery point> value of 0 shall be used.

Note: This Agreement is because of a deficiency of the standard. All other behaviors would lead to unpredictable results, because text and state tables in 8571-4 are ambiguous.

- It is required that Responders implementing the Restart-data-transfer or the Recovery functional unit must be able to negotiate <recovery mode> parameter to a value other than 'none'.

- For the <diagnostic> parameter of F-CANCEL/F-U-ABORT/F-P-ABORT the term <suggested delay> shall be supported if the Recovery or Restart-data-transfer functional units are implemented. The Basic FERPM should wait at least the amount of time as given by the <suggested delay> term before attempting to recover.

10.8.5 Concurrency Control

10.8.5.1 Concurrency Control to whole file

The <concurrency control> parameters of F-SELECT, F-CREATE and F-OPEN with or without the <access control> attribute of Security Group are supported for Initiators and optionally supported for Responders.

If supported by a Responder, details of their possible usage is a local matter and shall be specified in the PICS.

Default values for concurrency control are as specified for FTAM Phase 2 Agreements.

No minimum requirement is defined for <concurrency control> parameter values.

For a first accessor either the specified concurrency locks or the default values are assigned. For a subsequent accessor the access to a file is granted only if this concurrency control requirement, as specified in this concurrency control parameter or given by the default values, can be met. Otherwise the subsequent request shall be rejected.

10.8.5.2 FADU Locking

FADU locking functional unit and the respective <FADU lock> parameters are optionally supported for the Implementation Profiles A1.3 and A2.3.

It is understood that ISO 8571-4 Clause 18.4 also applies to FADU locks; that means that as long as a docket is maintained, FADU locks locking any FADUs recorded in that docket should be maintained.

10.8.6 Create Password

The <create password> parameter for an implementation acting as an Initiator is supported. This parameter is optionally supported for an implementation acting as a Responder.

10.9 Range of Values for Integer-Type Parameter

In addition to the parameters specified for FTAM Phase 2 under the same heading, the parameters

F-RECOVER request

```
            bulk-transfer-number
    NBS-AS3
        NBS-Node-Name
            starting-fadu
            fadu-count
```

may be encoded so that the length of its contents octets is no more than eight octets.

A P P E N D I C E S

APPENDIX A: PROFILES REQUIREMENTS LIST FOR NIST FTAM PHASE 3

APPENDIX B: FTAM PHASE 2/PHASE 3 COMPATIBILITY

APPENDIX C: DOCUMENT TYPES

APPENDIX D; CONSTRAINT SETS

APPENDIX E; ABSTRACT SYNTAXES

10.10 APPENDIX A

PROFILES REQUIREMENTS LIST FOR NIST FTAM PHASE 3

A.0 Introduction

This appendix to NIST FTAM Phase 3 Agreements defines a Profile Requirements List (PRL) for the Implementation Profiles

- T1.3 - Simple File Transfer
- T2.3 - Positional File Transfer
- A1.3 - Simple File Access
- M1.3 - Management

This appendix specifies the constraints and characteristics of NIST OIW FTAM Phase 3 on what shall or may appear in the supplier columns of an FTAM Phase 3 PICS. This appendix is completely based on ISO DIS 8571-5. It uses only a selection of the tables from ISO DIS 8571-5 which are necessary for the specification of the FTAM Phase 3 status, and retains their numbering, in order to facilitate for a supplier to fill in the respective PICS Proforma.

This appendix is a summary of all definitions of FTAM Phase 3 as they appear in the Stable Implementation Agreements for OSI Protocols, Version 2 Edition 1, Dec 1988, NIST Special Publication 500-162 (in the following referenced as 'Phase 2') and in chapter 10 of this document (in the following referenced as 'Phase 3').

A.0.1 Conformance requirement of Base Standards

The D-column of sections A.1 to A.13 specifies the conformance requirement of the base standards ISO 8571, as written in ISO 8571-5. The definitions apply as defined in ISO 8571-5 clause 8.1 :

- m - mandatory support
- o - optional support
- f - full support of attributes
- p - partial support of attributes
- - - not applicable

A single value in the D-column applies to the Initiator role of a system as well as to the Responder role. If two values are specified in the D-column separated by a space, they apply to the Initiator role and to the Responder role, respectively.

A.0.2 Conformance requirement of Profiles

The Conformance requirement of the Implementation Profiles is specified in the 'Profiles' column/columns in sections A.1 to A.13. The following convention is applied for this purpose :

- o a 'PROFILES' column is valid for all Profiles T1.3, T2.3, A1.3 and M1.3
- o if different conformance requirements apply to different Profiles, separate columns are included in the tables each bearing the corresponding Profile name as its heading,
  or separate tables for these Profiles are used

- o a single value in these columns applies to the Initiator as well as to the Responder role of an implementation
- o if two values are specified in a column separated by a space, they apply to the Initiator role and to the Responder role, respectively.

For the conformance requirement of NIST FTAM Phase 3 the following abbreviations are used.

supported; y :
This is a mandatory or optional feature in the base standard. Its syntax and semantics shall be implemented as specified in the base standard or in FTAM Phase 3 by all implementations claiming conformance to FTAM Phase 3.

However, it is not a requirement that the feature shall be used in all instances of communication, unless mandated by the base standard or stated otherwise in FTAM Phase 3.

For fully supported attributes, this implies that at least the minimum range of attribute values, as defined in ISO 8571-2, shall be supported unless stated otherwise in FTAM Phase 3.

Also for features which are optional in the base standard, conformant implementations shall be able to interwork with other implementations not supporting this feature.

The support of a feature can be conditional, depending on the support of a class of features to which it belongs, e.g. an attribute in an attribute group, a parameter in a PDU, a PDU in a functional unit.

optionally supported; o :
It is left to the implementation as to whether this feature is supported or not.

If an attribute group with a support level of 'o' is chosen to be supported, then all the attributes in this group that are classified as 'y' shall be supported.

The support for PDUs is determined by the negotiation of functional units when the connection is established.

If a parameter is optionally supported, then the syntax shall be supported, but it is left to each implementation whether the semantics are supported or not.

When receiving an optional parameter which is not subject of negotiation and is not supported by the Receiver, the Receiver shall at least inform the Sender by informative diagnostic and interworking shall not be disrupted.

conditionally supported; c :
This feature shall be supported under the conditions specified in FTAM Phase 3.

excluded; n :
This feature is excluded in FTAM Phase 3. The implementor's answer in the PICS shall always be 'no'.

outside the scope; / :
This feature is outside the scope of FTAM Phase 3 and will therefore not be subject of a Phase 3 conformance test. However the syntax of all parameters of supported PDUs shall be supported, even if the semantics are not (i.e. the Receiver shall be able to decode the PDU).

not applicable; - :
This feature is not defined in the context where it is mentioned, e.g. a parameter which is not part of the respective PDU. The occurrence of 'not applicable' features is mainly due to the format of the tables in the Phase 3 Profiles Requirements List.

Section one

A.1 (void)

A.2 (void)

Section two: General ISO 8571 Detail

A.3 ISO 8571 Protocol versions

1	FTAM protocol version number(s)	**One**

A.4 ISO 8571 Addenda

1	ISO 8571-1	--
2	ISO 8571-2	--
3	ISO 8571-3	--
4	ISO 8571-4	--
5	ISO 8571-5	--

A.5 Defect report numbers and amendments

1	ISO 8571-1	--
2	ISO 8571-2	--
3	ISO 8571-3	--
4	ISO 8571-4	--
5	ISO 8571-5	--

A.6 Global statement of conformance

1	**Are all mandatory features of ISO 8571 required?**	**yes**

A.7 Initiator / Responder capability

	ROLES	D	PROFILES
1	Sender	o	o
2	Receiver	o	o

NOTE - See section 9.18.1

A.8 Application Context Name details

1	**ISO 8571-4 defines a value for a simple transfer mechanism. Other values are outside the scope of FTAM Phase 3 (see 9.5(9)).**

Section three : Syntax Detail

A.9 Abstract syntaxes

	Object Descriptor	Object Identifier	D	T1.3	T2.3	A1.3	M1.3
1	**FTAM PCI**	{iso standard 8571 abstract-syntax (2) ftam-pci (1) }	m	y	y	y	y
2	**FTAM FADU**	{iso standard 8571 abstract-syntax (2) ftam-fadu (2) }	o	/	y	y	/
3		{joint-iso-ccitt association-control (2) abstract-syntax (1) apdus(0) version1 (1) }	m	y	y	y	y
4	**FTAM unstructured text abstract syntax**	{iso standard 8571 abstract-syntax (2) unstructured-text (3) }	o	y	y	y	-
5	**FTAM unstructured binary abstract syntax**	{iso standard 8571 abstract-syntax (2) unstructured-binary (4) }	o	y	y	y	-
6	**NBS file directory entry abstract syntax**	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-as2 (2) }	-	c	c	/	-
7	**NBS abstract syntax AS1**	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-as1 (1) }	-	/	c	c	-
8	**NBS random access node name abstract syntax**	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-node-name (3) }	-	/	c see 10.9	c	-
9	**NBS random binary access file abstract syntax**	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-random-binary (4) }	-	/	c	c	-
10	**NBS simple text abstract syntax**	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-simple-text (5) }	-	/	c	c	-

NOTES

1 The abstract syntaxes which are supported in the Implementation Profile M1.3 depend on the T-or A-Profile in conjunction with which M1.3 is implemented.

2 For the conditionally supported abstract syntaxes see section A.13.

3 ISO 8571 requires the presence of the transfer syntax derived from the "Basic Encoding of a single ASN.1 type "{joint-iso-ccitt asn1 (1) basic-encoding (1)} encoding rules for transfer of the "FTAM PCI" and the "FTAM FADU" abstract syntaxes. Implementation detail of this transfer syntax, and other transfer syntaxes supported, is specified in the PICS of ISO 8823.

Section four : Virtual Filestore Detail

A.10 Virtual filestore

This clause details the conformance to the file model, file attribute support and to file structure support.

A.10.1 File model

	FILE MODEL	D	PROFILES
1	Hierarchical	o	y
2	Other models		/

A.10.2 Attributes

A.10.2.1 Attribute groups

The level of support within each group is stated in A.10.2.2.

	ATTRIBUTE GROUP NAME	D	PROFILES
1	Kernel	m	y
2	Storage	o	o
3	Security	o	o
4	Private	o	/

A.10.2.2 Attribute values

	KERNEL GROUP (INITIATOR)	D	PROFILES full	RANGE OF VALUES
1	Filename	f	y	see A.10.2.3
2	Permitted Actions	f	y	
3	Contents Type	f	y	see A.12.7

NOTE - An initiator may not partially support attributes

	KERNEL GROUP (RESPONDER)	D	PROFILES full	RANGE OF VALUES
4	Filename	f	y	see A.10.2.3
5	Permitted Actions	f	y	
6	Contents Type	f	y	see A.12.7

	STORAGE GROUP (INITIATOR)	D	PROFILES full	RANGE OF VALUES
7	Storage account	f	y	
8	Date and time of creation	f	y	
9	Date and time of last modification	f	y	
10	Date and time of last read access	f	y	
11	Date and time of last attribute modification	f	y	
12	Identity of creator	f	y	
13	Identity of last modifier	f	y	
14	Identity of last reader	f	y	
15	Identity of last attribute modifier	f	y	
16	File availability	f	y	
17	Filesize	f	y	see 9.17.9
18	Future filesize	f	y	see 9.17.9

NOTE - An initiator may not partially support attributes

	STORAGE GROUP (RESPONDER)	D	PROFILES full	partial	RANGE OF VALUES
19	Storage account	p	o	o	
20	Date and time of creation	p	o	o	
21	Date and time of last modification	p	o	o	
22	Date and time of last read access	p	o	o	
23	Date and time of last attribute modification	p	o	o	
24	Identity of creator	p	o	o	
25	Identity of last modifier	p	o	o	
26	Identity of last reader	p	o	o	
27	Identity of last attribute modifier	p	o	o	
28	File availability	p	y	n	
29	Filesize	p	y	n	see 9.17.9
30	Future filesize	p	o	o	see 9.17.9

	SECURITY GROUP (INITIATOR)	D	PROFILES full	RANGE OF VALUES
31	Access control	f	y	see A.12.2
32	Legal qualifications	f	y	

NOTE - An initiator may not partially support attributes

	SECURITY GROUP (RESPONDER)	D	PROFILES full	partial	RANGE OF VALUES
33	Access control	p	y	n	see A.12.2, 9.9.2
34	Legal qualifications	p	o	o	

A.10.2.3 Filename detail

See section 9.9.1

A.10.3 File structures

A.10.3.1 Constraint sets

	CONSTRAINT SET NAME	D	T1.3	T2.3	A1.3	M1.3
1	Unstructured	o	y	y	y	-
2	Sequential Flat	o	/	y	y	-
3	Ordered flat	o	/	o	o	-
4	Ordered flat with unique names	o	/	o	o	-
5	Ordered hierarchical	o	/	/	/	-
6	General hierarchical	o	/	/	/	-
7	General hierarchical with unique names	o	/	/	/	-
8	NBS ordered flat	-	/	o	o	-
9	NBS random access	-	/	o	o	-

A.10.3.2 File and filestore actions

A.10.3.2.1 Filestore Actions

Support for filestore actions is dependent upon the functional units implemented (see A.12.4 and A.12.5)

A.10.3.2.2 File Actions

	Responder	CONSTRAINT SET unstructured	
	ACTION	D	T1.3
1	Locate	---------	
2	Read	o	o
3	Insert	---------	
4	Replace	o	o
5	Extend	o	o
6	Erase	o	/

	Responder	CONSTRAINT SET											
		unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACTION	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3	D	T2.3
7	Locate	---------		o	/	o	/	o	/	-	/	-	/
8	Read	o	o	o	o	o	o	o	o	-	o	-	o
9	Insert	---------		o	o	o	o	o	o	-	o	-	o
10	Replace	o	o	---------		o	o	o	o	-	o	-	o
11	Extend	o	o	---------		o	o	o	o	---------		---------	
12	Erase	o	/	o	/	o	/	o	/	-	/	-	/

	Responder	CONSTRAINT SET unstructured		sequential flat		ordered flat		ordered flat with unique names		NBS ordered flat		NBS random access	
	ACTION	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3	D	A1.3
13	Locate	——		o	o	o	o	o	o	-	o	-	o
14	Read	o	o	o	o	o	o	o	o	-	o	-	o
15	Insert	——		o	o	o	o	o	o	-	o	-	o
16	Replace	o	o	——		o	o	o	o	-	o	-	o
17	Extend	o	o	——		o	o	o	o	——		——	
18	Erase	o	o	o	o	o	o	o	o	-	o	-	o

NOTE - File actions are not defined in Implementation Profile M1.3

A.10.3.2.3 Access contexts supported

	Responder	CONSTRAINT SET unstructured	
	ACCESS CONTEXT	D	T1.3
1	US	——	
2	UA	o	y
3	FS	——	
4	FL	——	
5	FA	——	
6	HN	——	
7	HA	——	

	Responder / ACCESS / CONTEXT	CONSTRAINT SET: unstructured D	unstructured T2.3	sequential flat D	sequential flat T2.3	ordered flat D	ordered flat T2.3	ordered flat with unique names D	ordered flat with unique names T2.3	NBS ordered flat D	NBS ordered flat T2.3	NBS random access D	NBS random access T2.3
8	US	---		---		---		---		---		---	
9	UA	o	y	o	y	o	y	o	y	-	y	-	y
10	FS	---		---		---		---		---		---	
11	FL	---		---		---		---		---		---	
12	FA	---		o	y	o	y	o	y	-	y	---	
13	HN	---		---		---		---		---		---	
14	HA	---		---		o	o	o	o	-	o	---	

	Responder / ACCESS / CONTEXT	CONSTRAINT SET: unstructured D	unstructured A1.3	sequential flat D	sequential flat A1.3	ordered flat D	ordered flat A1.3	ordered flat with unique names D	ordered flat with unique names A1.3	NBS ordered flat D	NBS ordered flat A1.3	NBS random access D	NBS random access A1.3
15	US	---		---		---		---		---		---	
16	UA	o	y	o	y	o	y	o	y	-	y	-	y
17	FS	---		---		---		---		---		---	
18	FL	---		---		---		---		---		---	
19	FA	---		o	y	o	y	o	y	-	y	---	
20	HN	---		---		---		---		---		---	
21	HA	---		---		o	y	o	y	-	y	---	

NOTE - The supported access contexts for Impementation Profile M1.3 are defined in the T- or A-Profile in conjunction with which M1.3 is implemented.

A.10.4 Additional information

(Void)

A.10.5 Override

	Responder override	D	PROFILES
1	Create failure	o	y
2	Select old file	o	y
3	Delete and recreate with old attributes	o	o
4	Delete and create with new attributes	o	y

NOTE - The specification of the role of initator is given in section five (file protocol detail).

Section five : File Protocol Detail

A.11 File protocol

See sections 9.5(1) - (3), 9.17

NOTES

1) In order to keep the protocol tables compact some forward references have been introduced to clauses which expand upon the detail of field support.

2) The FTAM protocol will require a number of optional lower layer services to be available (eg Application Entity Titles in ACSE). This requirement is outside the scope of this ISPICS Requirements List.

A.11.1 GraphicString support

(Void)

A.11.2 FTAM regime establishment

	F-INITIALIZE FIELD NAME	D		PROFILES		RANGE OF VALUES
1	State result	-	m	-	y	all values defined in ISO 8571
2	Action result	-	m	-	y	all values defined in ISO 8571
3	Protocol version	m	m	y	y	see Section 2
4	Implementation information	o	o	o	o	see A.12.1
5	Presentation context management	m	m	y	y	see note 1, 9.17.10
6	Service class	m	m	y	y	see A.12.4
7	Functional units	m	m	y	y	see A.12.5
8	Attribute groups	m	m	y	y	see A.10.2
9	Shared ASE information	o	o	/	/	see 9.5(8)
10	FTAM Quality of Service	m	m	y	y	see A.12.8
11	Contents type list	o	o	y	y	see A.12.7.1, 9.18.4
12	Initiator identity	o	-	y	-	see 9.16.1, 9.18.4
13	Account	o	-	o	-	see 9.18.4
14	Filestore password	o	-	y	-	see 9.16.1
15	Diagnostic	-	o	-	y	see A.12.6, 9.13
16	Checkpoint window	m	m	y	y	see note 2, 10.8.4.2

NOTES
1) The values available for the presentation context management field depend upon the functional units implemented in ISO 8823.
2) Checkpoint window field is indicated as mandatory in accordancqwith ISO 8571-4. The field is defaulted to the value 1.

A.11.3 FTAM regime termination (orderly)

	F-TERMINATE FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Shared ASE information	o o	/ /	see 9.5 (8)
2	Charging	- o	- o	see A.12.10

A.11.4 FTAM regime termination (abrupt) by service user

	F-U-ABORT FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Action result	m	y	all values defined in ISO 8571
2	Diagnostic	o	y	see A.12.6, 9.13, 10.8.4.2

A.11.5 FTAM regime termination (abrupt) by service provider

	F-P-ABORT FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Action result	m	y	all values defined in ISO 8571
2	Diagnostic	o	y	see A.12.6, 9.13, 10.8.4.2

A.11.6 File selection

	F-SELECT FIELD NAME	D	PROFILES	RANGE OF VALUES
1	State result	- m	- y	all values defined in ISO 8571
2	Action result	- m	- y	all values defined in ISO 8571
3	Attributes	m m	y y	see A.10.2, 9.17.9
4	Requested access	m -	y -	see A.12.16
5	Access passwords	o -	c -	see 9.16.2
6	Concurrency control	o -	y -	see A.12.13, 10.8.5.1
7	Shared ASE information	o o	/ /	see 9.5(8)
8	Account	o -	o -	see 9.18.4
9	Diagnostic	- o	- y	see A.12.6, 9.13

A.11.7 File deselection

	F-DESELECT FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Action result	- m	- y	all values defined in ISO 8571
2	Charging	- o	- o	see A.12.10
3	Shared ASE information	o o	/ /	see 9.5(8)
4	Diagnostic	- o	- y	see A.12.6, 9.13

A.11.8 File creation

	F-CREATE FIELD NAME	D	PROFILES	RANGE OF VALUES
1	State result	- m	- y	all values defined in ISO 8571
2	Action result	- m	- y	all values defined in ISO 8571
3	Override	m -	y -	see A.12.15
4	Initial attributes	m m	y y	see A.10.2, 9.10.2.2, 9.17.9
5	Create password	o -	y -	see 9.16.2, 10.8.6
6	Requested access	m -	y -	see A.12.16
7	Access password	o -	c -	see 9.16.2
8	Concurrency control	o -	y -	see A.12.13, 10.8.5.1
9	Shared ASE information	o o	/ /	see 9.5(8)
10	Account	o -	o -	see 9.18.4
11	Diagnostic	- o	- y	see A.12.6, 9.13

A.11.9 File deletion

	F-DELETE FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Action result	- m	- y	all values defined in ISO 8571
2	Shared ASE information	o o	/ /	
3	Charging	- o	- o	see A.12.10
4	Diagnostic	- o	- y	see A.12.6, 9.13

A.11.10 Read attributes

	F-READ-ATTRIB FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Action result	- m	- y	all values defined in ISO 8571
2	Attribute names	m -	y -	
3	Attributes	- o	- y	see A.10.2, 9.17.9
4	Diagnostic	- o	- y	see A12.6, 9.13

A.11.11 Change attributes

	F-CHANGE-ATTRIB FIELD NAME	D	T1.3, T2.3, A1.3,	M1.3	RANGE OF VALUES
1	Action result	- m	/	- y	all values defined in ISO 8571
2	Attributes	m o	/	y y	see A.10.2, 9.17.9
3	Diagnostic	- o	/	- y	see A.12.6, 9.13

A.11.12 File open

	F-OPEN FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	State result	- m	- y	/	all values defined in ISO 8571
2	Action result	- m	- y	/	all values defined in ISO 8571
3	Processing mode	m -	y -	/	see A.12.17
4	Contents type	m m	y y	/	see A.12.7.2
5	Concurrency control	o o	y o	/	see A.12.13, 10.8.5.1
6	Shared ASE information	o o	/ /	/	see 9.5(8)
7	Enable FADU locking	m -	y -	/	'false' for T1.3 and T2.3
8	Activity identifier	o -	o -	/	
9	Diagnostic	- o	- y	/	see A.12.6, 9.13
10	Recovery mode	m m	y y	/	see A. 12.18
11	Remove contexts	o -	/ -	/	
12	Define contexts	o -	/ -	/	
13	Presentation action	- m	- y	/	see notes

NOTES

1) The values available for the presentation action field depend upon the functional units implemented in ISO 8823.

2) Presentation action field is indicated as mandatory in accordance with ISO 8571-4. The field is defaulted to no action.

A.11.13 File close

	F-CLOSE FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	Action result	m	y	/	all values defined in ISO 8571
2	Shared ASE information	o	/	/	see 9.5(8)
3	Diagnostic	o	y	/	see A.12.6, 9.13

A.11.14 Beginning of grouping

	F-BEGIN-GROUP FIELD NAME	D	PROFILES	RANGE OF VALUES
1	Threshold	m -	y -	

A.11.15 End of grouping

The F-END-GROUP PDU carries no fields

A.11.16 Regime recovery

See section 10.8.4

	F-RECOVER FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	State result	- m	- y	/	all values defined in ISO 8571
2	Action result	- m	- y	/	all values defined in ISO 8571
3	Activity identifier	m -	y -	/	
4	Bulk transfer number	m -	y -	/	see 10.9
5	Requested access	m -	y -	/	see A.12.16
6	Access passwords	o -	c -	/	see 9.16.2
7	Contents type	- m	- y	/	see A.12.7.2
8	Recovery point	m m	y y	/	
9	Diagnostic	- o	- y	/	see A.12.6, 9.13
10	Remove contexts	o -	/ -	/	see notes
11	Define contexts	o -	/ -	/	see notes
12	Presentation action	- m	- y	/	see notes

NOTES

1) The values available for the presentation action field depend upon the functional units implemented in ISO 8823.

2) Presentation action field is indicated as mandatory in accordance with ISO 8571-4. The field is defaulted to no action.

A.11.17 Locate file access data unit

	F-LOCATE FIELD NAME	D	T1.3, T2.3	A1.3	M1.3	
1	Action result	- m	/	- y	/	all values defined in ISO 8571
2	FADU identity	m o	/	y o	/	see 9.17.9
3	FADU lock	o -	/	o -	/	see A.12.14
4	Diagnostic	- o	/	- y	/	see A.12.6, 9.13

A.11.18 Erase file access data unit

	F-ERASE FIELD NAME	D	T1.3, T2.3	A1.3	M1.3	
1	Action result	- m	/	- y	/	all values defined in ISO 8571
2	FADU identity	m -	/	y -	/	see 9.17.9
3	Diagnostic	- o	/	- y	/	see A.12.6, 9.13

A.11.19 Read bulk data

	F-READ FIELD NAME	D	T1.3, T2.3	A1.3	M1.3	RANGE OF VALUES
1	FADU identity	m -	y -	y -	/	see 9.17.9
2	Access context	m -	y -	y -	/	see A.10.3.2.3
3	FADU lock	o -	/ -	o -	/	

A.11.20 Write bulk data

	F-WRITE FIELD NAME	D	T1.3, T2.3	A1.3	M1.3	RANGE OF VALUES
1	FADU operation	m -	y -	y -	/	
2	FADU identity	m -	y -	y -	/	see 9.17.9
3	FADU Lock	o -	/ -	o -	/	

A.11.21 End of data transfer

	F-DATA-END FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	Action result	m	y	/	all values defined in ISO 8571
2	Diagnostic	o	y	/	see A.12.6, 9.13

A.11.22 End of transfer

	F-TRANSFER-END FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	Action result	- m	- y	/	all values defined in ISO 8571
2	Shared ASE information	o o	/ /	/	see 9.5(8)
3	Diagnostic	- o	- y	/	see A.12.6, 9.13

A.11.23 Cancel data transfer

See section 9.11

	F-CANCEL FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	Action result	m	y	/	all values defined in ISO 8571
2	Shared ASE information	o	/	/	see 9.5(8)
3	Diagnostic	o	y	/	see A.12.6, 9.13, 10.8.4.2

A.11.23.1 F-CANCEL mapping

See sections 9.11, 9.17.10

A.11.24 Restart data transfer

	F-RESTART FIELD NAME	D	T1.3, T2.3, A1.3	M1.3	RANGE OF VALUES
1	Checkpoint identifier	m	y	/	

A.12 Expanded field detail

This clause identifies further field detail to expand on that given in A.10 and A.11.

A.12.1 Implementation information detail

See sections 9.5(6), 9.12, 10.8.2

A.12.2 Access control detail

See sections 9.9.2, 10.7.3

	Access control element terms	D	PROFILES	RANGE OF VALUES
1	Action list	m	y y	
2	Concurrency access	o	o o	see A.12.3.3
3	Identity	o	o o	
4	Passwords	o	c o	see A.12.3.6
5	Location	o	o o	

A.12.3 Access control element detail

A.12.3.1 Action list detail (initiator)

(Void)

A.12.3.2 Action list detail (responder)

(Void)

A.12.3.3 Concurrency access term

If the concurrency access term is supported in the access control element the following details of the concurrency control shall be available with each action.

	RESPONDER Action	not required D	not required T1.3	shared D	shared T1.3	exclusive D	exclusive T1.3	no access D	no access T1.3
1	Read	o	o	o	o	o	o	o	o
2	Insert	o	/	o	/	o	/	o	/
3	Replace	o	o	o	o	o	o	o	o
4	Extend	o	o	o	o	o	o	o	o
5	Erase	o	/	o	/	o	/	o	/
6	Read attributes	o	o	o	o	o	o	o	o
7	Change attributes	o	/	o	/	o	/	o	/
8	Delete file	o	o	o	o	o	o	o	o

NOTE - no equivalent table exists for the initiator

	RESPONDER Action	not required D	not required T2.3	shared D	shared T2.3	exclusive D	exclusive T2.3	no access D	no access T2.3
9	Read	o	o	o	o	o	o	o	o
10	Insert	o	o	o	o	o	o	o	o
11	Replace	o	o	o	o	o	o	o	o
12	Extend	o	o	o	o	o	o	o	o
13	Erase	o	/	o	/	o	/	o	/
14	Read attributes	o	o	o	o	o	o	o	o
15	Change attributes	o	/	o	/	o	/	o	/
16	Delete file	o	o	o	o	o	o	o	o

	RESPONDER Action	not required D	not required A1.3	shared D	shared A1.3	exclusive D	exclusive A1.3	no access D	no access A1.3
17	Read	o	o	o	o	o	o	o	o
18	Insert	o	o	o	o	o	o	o	o
19	Replace	o	o	o	o	o	o	o	o
20	Extend	o	o	o	o	o	o	o	o
21	Erase	o	o	o	o	o	o	o	o
22	Read attributes	o	o	o	o	o	o	o	o
23	Change attributes	o	/	o	/	o	/	o	/
24	Delete file	o	o	o	o	o	o	o	o

NOTE - no equivalent table exists for the initiator

RESPONDER Action	not required D	not required M1.3	shared D	shared M1.3	exclusive D	exclusive M1.3	no access D	no access M1.3
Read	o	/	o	/	o	/	o	/
Insert	o	/	o	/	o	/	o	/
Replace	o	/	o	/	o	/	o	/
Extend	o	/	o	/	o	/	o	/
Erase	o	/	o	/	o	/	o	/
Read attributes	o	o	o	o	o	o	o	o
Change attributes	o	o	o	o	o	o	o	o
Delete file	o	o	o	o	o	o	o	o

NOTE - no equivalent table exists for the initiator

A.12.3.4 Identity term

(Void)

A.12.3.5 Access passwords - general detail

See section 9.16.3

A.12.3.6 Passwords term

Responder	D	PROFILES
OctetString	o	o
GraphicString	o	o

A.12.3.7 Location term

(Void)

A.12.3.7.1 Application Entity Titles detail

See section 9.5(7)

A.12.3.8 Access control element combinations

	Responder			D	PROFILES
1	Identity	Password	Location	o	o
2	Identity	Password		o	o
3	Identity		Location	o	o
4		Password	Location	o	o
5	Identity			o	o
6		Password		o	o
7			Location	o	o

NOTE - Implementation of access control without any of the above combinations is valid.

A.12.4 Service class field detail

See table 9.7 and section 10.5.1

		D	T1.3, T2.3	A1.3	M1.3
1	Transfer class	o	y	/	/
2	Access class	o	/	y	/
3	Management class	o	/	/	y
4	Transfer and management class	o	o	/	/
5	Unconstrained class	o	/	/	/

NOTE - the initiator is only permitted to specify those combinations defined in ISO 8571-3

A.12.5 Functional unit field detail

See table 9.7 and section 10.8.1

	T1.3, T2.3	SERVICE CLASSES			
		Transfer		Transfer Management	
	FUNCTIONAL UNITS	D	T1.3, T2.3	D	T1.3, T2.3
1	Kernel	m	y	m	y
2	Read (see note 2)	*	o	*	o
3	Write (see note 2)	*	o	*	o
4	File Access	----------		----------	
5	Limited File Management	o	o	m	y
6	Enhanced File Management	o	/	o	/
7	Grouping	m	y	m	y
8	FADU Locking	----------		-	--
9	Recovery	o	o	o	o
10	Restart	o	o	o	o

NOTES

1. the recovery and the restart functional units are only available at the internal file service interface and should only be explicitly referenced in the protocol.

2. the * indicates that either or both of the read and write functional units shall be implemented in the particular service class

	A1.3	SERVICE CLASSES Access		
	FUNCTIONAL UNITS	**D**	**A1.3**	
11	Kernel	m	**y**	
12	Read	m	**y**	
13	Write	m	**y**	
14	File Access	m	**y**	
15	Limited File Management	o	**o**	
16	Enhanced File Management	o	/	
17	Grouping	o	**o**	
18	FADU Locking	o	**o**	see 10.8.5.2
19	Recovery	o	**o**	
20	Restart	o	**o**	

	M1.3	SERVICE CLASSES Management		
	FUNCTIONAL UNITS	**D**	**M1.3**	
21	Kernel	m	**y**	
22	Read	/	/	
23	Write	/	/	
24	File Access	----------		
25	Limited File Management	m	**y**	
26	Enhanced File Management	o	**y**	see 10.8..1
27	Grouping	m	**y**	
28	FADU Locking	----------		
29	Recovery	----------		
30	Restart	----------		

A.12.6 Diagnostic field detail

		D	T1.3, T2.3, A1.3	M1.3	
1	Diagnostic type	m	y	y	
2	Error identifier	m	y	y	
3	Error observer	m	y	y	
4	Error source	m	y	y	
5	Suggested delay	o	c	/	see 10.8.4.2
6	Further details	o	y	y	

For values of the 'further details' term only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required (see section 9.13).

A.12.7 Contents type detail

A.12.7.1 Contents type list parameter

See section 9.10.2.1

		D	PROFILES	Maximum number of elements
1	document type specifications	o	o y	
2	abstract syntax specifications	o	o y	

A.12.7.2 Contents type parameter

See section 9.10.2.3

		D	PROFILES	
1	document type specifications	o	y	see 9.9.1
2	abstract syntax / constraint set pair specifications	o	/	

NOTE - The detail of document types supported is contained in section A.13.

A.12.8 FTAM Quality of service details

See section 10.8.4.2

A.12.9 Details of shared ASE information

(Void)

A.12.10 Details of charging

See section 9.5(8), 9.18.4

	Charging Responder	D	PROFILES
1	Resource identifier term	m	y
2	Charging unit term	m	y
3	Charging value term	m	y

A.12.11 General Password Detail

(Void)

A.12.12 Responder Access passwords

	Responder	D	T1.3 OctetString GraphicString	T2.3 OctetString GraphicString	A1.3 OctetString GraphicString	M1.3 OctetString GraphicString
1	Read-password	o	o	o	o	/
2	Insert-password	o	/	o	o	/
3	Replace-password	o	o	o	o	/
4	Extend-password	o	o	o	o	/
5	Erase-password	o	/	/	o	/
6	Read-attribute password	o	o	o	o	o
7	Change-attribute password	o	/	/	/	o
8	Delete-password	o	o	o	o	o

NOTE - See A.12.3 for initiator support of this feature.

A.12.13 Concurrency control

A.12.13.1 Supported values

See section 10.8.5.1

	T1.3	not required		shared		exclusive		no access	
	Action	D	T1.3	D	T1.3	D	T1.3	D	T1.3
1	Read	o	o	o	o	o	o	o	o
2	Insert	o	/	o	/	o	/	o	/
3	Replace	o	o	o	o	o	o	o	o
4	Extend	o	o	o	o	o	o	o	o
5	Erase	o	/	o	/	o	/	o	/
6	Read attrib	o	o	o	o	o	o	o	o
7	Change attrib	o	/	o	/	o	/	o	/
8	Delete file	o	o	o	o	o	o	o	o

	T2.3	not required		shared		exclusive		no access	
	Action	D	T2.3	D	T2.3	D	T2.3	D	T2.3
9	Read	o	o	o	o	o	o	o	o
10	Insert	o	o	o	o	o	o	o	o
11	Replace	o	o	o	o	o	o	o	o
12	Extend	o	o	o	o	o	o	o	o
13	Erase	o	/	o	/	o	/	o	/
14	Read attrib	o	o	o	o	o	o	o	o
25	Change attrib	o	/	o	/	o	/	o	/
16	Delete file	o	o	o	o	o	o	o	o

	A1.3	not required		shared		exclusive		no access	
	Action	**D**	**A1.3**	**D**	**A1.3**	**D**	**A1.3**	**D**	**A1.3**
17	Read	o	o	o	o	o	o	o	o
18	Insert	o	o	o	o	o	o	o	o
19	Replace	o	o	o	o	o	o	o	o
20	Extend	o	o	o	o	o	o	o	o
21	Erase	o	o	o	o	o	o	o	o
22	Read attrib	o	o	o	o	o	o	o	o
23	Change attrib	o	/	o	/	o	/	o	/
24	Delete file	o	o	o	o	o	o	o	o

	M1.3	not required		shared		exclusive		no access	
	Action	**D**	**M1.3**	**D**	**M1.3**	**D**	**M1.3**	**D**	**M1.3**
25	Read	o	/	o	/	o	/	o	/
26	Insert	o	/	o	/	o	/	o	/
27	Replace	o	/	o	/	o	/	o	/
28	Extend	o	/	o	/	o	/	o	/
29	Erase	o	/	o	/	o	/	o	/
30	Read attrib	o	o	o	o	o	o	o	o
31	Change attrib	o	o	o	o	o	o	o	o
32	Delete file	o	o	o	o	o	o	o	o

A.12.13.2 Responder Default values

See sections 9.14, 10.8.5.1

A.12.14 FADU Locking

A1..3	FADU Locking Support Values							
	not required		shared		exclusive		no access	
	D	A1.3	D	A1.3	D	A1.3	D	A1.3
1 Read	o	o	o	o	o	o	o	o
2 Insert	o	o	o	o	o	o	o	o
3 Replace	o	o	o	o	o	o	o	o
4 Extend	o	o	o	o	o	o	o	o
5 Erase	o	o	o	o	o	o	o	o

A.12.15 Initiator Override

Initiator override	D	PROFILES
1 Create failure	o	o
2 Select old file	o	o
3 Delete and recreate with old attributes	o	o
4 Delete and create with new attributes	o	o

NOTE - The specification of the role of responder is given in the filestore section

A.12.16 Requested Access

See section 9.15

Action	D	T1.3	T2.3	A1.3	M1.3
1 Read	o	o	o	o	/
2 Insert	o	/	o	o	/
3 Replace	o	o	o	o	/
4 Extend	o	o	o	o	/
5 Erase	o	n	n	o	/
6 Read attribute	o	o	o	o	y
7 Change attribute	o	/	/	/	y
8 Delete file	o	o	o	o	y

A.12.17 Processing mode

	Processing mode	D	T1.3	T2.3	A1.3	M1.3
1	Read	o	o	o	o	/
2	Insert	o	/	o	o	/
3	Replace	o	o	o	o	/
4	Extend	o	o	o	o	/
5	Erase	o	n	n	o	/

A.12.18 Recovery mode

See section 10.8.4.2

	Recovery mode	D	T1.3, T2.3, A1.3	M1.3
1	None	o	y	/
2	At start of file	o	y	/
3	Any active checkpoint	o	o	/

Section six : Document Types

A.13 Document types

See section 10.7.1

Conformance to document types is given at two levels. The following table indicates which document types have some level of support. The detail of that level of support is stated in the following sections.

	Entry number	FTAM-1	D	T1.3	T2.3	A1.3	M1.3
1	Object descriptor	ISO FTAM unstructured text	o	y	y	y	/
	Object identifier	{iso standard 8571 document-type (5) unstructured-text (1)}					

	Entry number	FTAM-2	D	T1.3	T2.3	A1.3	M1.3
2	Object descriptor	ISO FTAM sequential text	o	/	y	y	/
	Object identifier	{iso standard 8571 document-type (5) sequential-text (2)}					

	Entry number	FTAM-3	D	T1.3	T2.3	A1.3	M1.3
3	Object descriptor	ISO FTAM unstructured binary	o	y	y	y	/
	Object identifier	{iso standard 8571 document-type (5) unstructured-binary (3)}					

	Entry number	FTAM-4	D	T1.3	T2.3	A1.3	M1.3
4	Object descriptor	ISO FTAM sequential binary	o	/	o	o	/
	Object identifier	{iso standard 8571 document-type (5) sequential-binary (4)}					

	Entry number	NBS-6	D	T1.3	T2.3	A1.3	M1.3
5	Object descriptor	NBS-6 FTAM sequential file	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) sequential (6) }					

	Entry number	NBS-7	D	T1.3	T2.3	A1.3	M1.3
6	Object descriptor	NBS-7 FTAM random access file	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) random-file (7) }					

	Entry number	NBS-8	D	T1.3	T2.3	A1.3	M1.3
7	Object descriptor	NBS-8 FTAM indexed file	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) indexed-file (8) }					

	Entry number	NBS-9	D	T1.3	T2.3	A1.3	M1.3
8	Object descriptor	NBS-9 FTAM file directory file	-	o	o	/	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) file-directory (9) }			see 9.18.3		

	Entry number	NBS-10	D	T1.3	T2.3	A1.3	M1.3
9	Object descriptor	NBS-10 FTAM random binary access file	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) random-binary (10) }			see 9.10		

	Entry number	NBS-11	D	T1.3	T2.3	A1.3	M1.3
10	Object descriptor	NBS-11 FTAM indexed file with unique keys	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) indexed-file-with-unique-keys (11) }					

	Entry number	NBS-12	D	T1.3	T2.3	A1.3	M1.3
11	Object descriptor	NBS-12 NBS FTAM simple text file	-	/	o	o	/
	Object identifier	{iso identified-organization icd (9999) organization-code (1) document-type (5) simple-text-file (12) }					

A.13.1 Constraint sets and FADU identities for document types

For the constraint set/FADU identity tables in section A.13.1 the following notation is used:

m	mandatory	in the constraint set definition
o	optional	in the constraint set definition
y	supported	(shall be implemented by implementations claiming conformance to FTAM Phase 3. The actions with which the identity can be used, are given in the constraint set definition)
/	not supported	(outside the scope of this ISP)
-	not applicable	(not defined in the constraint set definition)
n	excluded	(disallowed in the document type definition or in FTAM Phase 3)

Implementation Profile T1.3.

FADU Identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node-Name Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM-1	-	-	y	-	-	-	-	-	-
FTAM-3	-	-	y	-	-	-	-	-	-
NBS-9	-	-	y	-	-	-	-	-	-

Implementation Profile T2.3 (see sections 9.10, 10.7.2)

FADU Identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node-Name Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM - 1	-	-	y	-	-	-	-	-	-
FTAM - 3	-	-	y	-	-	-	-	-	-
NBS-9	-	-	y	-	-	-	-	-	-
FTAM sequential flat constraint set	o	o	o	o	o	o	o	-	o
FTAM-2	y	y	/	/	/	/	/	-	/
FTAM-4	y	y	/	/	/	/	/	-	/
NBS-6	y	y	/	n	n	/	n	-	n
NBS-12	y	y	n	n	n	n	n	-	n
FTAM ordered flat constraint set	o	o	o	o	o	o	o	o	o
NBS-8	y	/	/	/	/	/	/	y	/
FTAM ordered flat constraint set with unique names	o	o	-	-	o	o	o	o	o
NBS-11	y	/	-	-	/	/	/	y	/
NBS ordered flat constraint set	o	o	o	o	o	o	o	-	o
NBS-7	y	y	y	y	/	/	/	-	y
NBS random access constraint set	o	o	-	-	-	-	-	o	o
NBS-10	y	y	-	-	-	-	-	y	y

Implementation Profile A1.3 (see section 9.10)

FADU Identity / Constraint Set	Begin	End	First	Last	Current	Next	Previous	Node-Name Node Seq	Node Number
FTAM unstructured constraint set	-	-	m	-	-	-	-	-	-
FTAM-1	-	-	y	-	-	-	-	-	-
FTAM-3	-	-	y	-	-	-	-	-	-
NBS-9	-	-	y	-	-	-	-	-	-
FTAM sequential flat constraint set	o	o	o	o	o	o	o	-	o
FTAM-2	y	y	y	/	/	y	/	-	/
FTAM-4	y	y	y	/	/	y	/	-	/
NBS-6	y	y	y	n	n	y	n	-	n
NBS-12	y	y	y	n	n	y	n	-	n
FTAM ordered flat constraint set	o	o	o	o	o	o	o	o	o
NBS-8	y	y	/	/	y	y	y	y	/
FTAM ordered flat constraint set with unique names	o	o	-	-	o	o	o	o	o
NBS-11	y	y	-	-	y	y	y	y	/
NBS ordered flat constraint set	o	o	o	o	o	o	o	-	o
NBS-7	y	y	y	y	y	y	y	-	y
NBS random access constraint set	o	o	-	-	-	-	-	o	o
NBS-10	y	y	-	-	-	-	-	y	y

A.13.2 Parameter details for document types

A.13.2.1 FTAM-1 (See section 10.7.1)

A.13.2.1.1 Universal class number parameter (See section 9.10.1)

		D	T1.3, T2.3, A1.3	
1	Universal class number parameter supported	o	y	
2	PrintableString - Universal class 19	o	/	
3	TeletexString - Universal class 20	o	/	
4	VideotexString - Universal class 21	o	/	
5	IA5String - Universal class 22	o	y	see 9.10.1.1-2
6	GraphicString - Universal class 25	o	y	see A13.2.1.4
7	VisibleString - Universal class 26	o	y	
8	GeneralString - Universal class 27	o	y	see A.13.2.1.5

A.13.2.1.2 String length parameter

		D	T1.3, T2.3, A1.3
1	Maximum string length parameter supported	o	y
2	Are unbounded string lengths supported?	o	y

A.13.2.1.3 String significance parameter

		D	T1.3, T2.3, A1.3
1	String significance parameter supported	o	y
2	Variable length strings supported	o	y
3	Fixed length strings supported	o	y
4	Not significant strings supported	o	y

A.13.2.1.4 G sets supported

G sets which are supported in FTAM-1 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3**

A.13.2.1.5 G and C sets supported

G and C sets which are supported in FTAM-1 GeneralString

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets and ISO 646 IRV (C0) control character set is required** **(see 9.10.1-3)**

A.13.2.2 FTAM-2 (see section 10.7.1)

A.13.2.2.1 Universal class number parameter (see section 9.10.1)

		D	T2.3, A1.3	
1	Universal class number parameter supported	o	**y**	
2	PrintableString - Universal class 19	o	/	
3	TeletexString - Universal class 20	o	/	
4	VideotexString - Universal class 21	o	/	
5	IA5String - Universal class 22	o	**o**	see 9.10.1.1-2
6	GraphicString - Universal class 25	o	**y**	see A.13.2.2.4
7	VisibleString - Universal class 26	o	**y**	
8	GeneralString - Universal class 27	o	**o**	see A.13.2.2.5

A.13.2.2.2 String length parameter

		D	T2.3, A1.3
1	Maximum string length parameter supported	o	**y**
2	Are unbounded string lengths supported ?	o	**y**

A.13.2.2.3 String significance parameter

		D	T2.3, A1.3
1	String significance parameter supported	o	y
2	variable length strings supported	o	/
3	Fixed length strings supported	o	/
4	Not significant strings supported	o	y

A.13.2.2.4 G sets supported

G sets which are supported in FTAM-2 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3**

A.13.2.2.5 G and C sets supported

G and C sets which are supported in FTAM-2 GeneralString

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets and ISO 646 IRV (C0) control character set is required.** **see 9.10.1.1-3**

A.13.2.3 FTAM-3

A.13.2.3.1 String length parameter (see section 10.7.1)

		D	T1.3, T2.3, A1.3
1	Maximum string length parameter supported	o	y
2	Are unbounded string lengths supported?	o	y

A.13.2.3.2 String significance parameter

		D	T1.3, T2.3, A1.3
1	String significance parameter supported	o	y
2	Variable length strings supported	o	/
3	Fixed length strings supported	o	/
4	Not significant strings supported	o	y

A.13.2.4 FTAM-4 (see section 10.7.1)

A.13.2.4.1 String length parameter

		D	T2.3, A1.3
1	Maximum string length parameter supported	o	y
2	Are unbounded string lengths supported ?	o	y

A.13.2.4.2 String significance parameter

		D	T2.3, A1.3
1	String significance parameter supported	o	y
2	Variable length strings supported	o	/
3	Fixed length strings supported	o	/
4	Not significant strings supported	o	y

A.13.2.5 NBS-6

See tables 9.2, 9.3

A.13.2.5.1 Parameter0

				D	T2.3, A1.3
1	Parameter0 supported			-	y
2	Universal-time	-	Universal class 23	-	y
3	Generalized-time	-	Universal class 24	-	y
4	boolean	-	Universal class 1	-	y
5	null	-	Universal class 5	-	y

A.13.2.5.2 Parameter1 (see section 9.10.1)

				D	T2.3, A1.3
1	Parameter1 supported			-	y
2	integer	-	Universal class 2	-	y
3	bit	-	Universal class 3	-	y
4	IA5	-	Universal class 22	-	y
5	GraphicString	-	Universal class 25	-	y
6	GeneralString	-	Universal class 27	-	y
7	OctetString	-	Universal class 4	-	y

A.13.2.5.3 Parameter2

		D	T2.3, A1.3
1	Parameter2 supported	-	o

A.13.2.6 NBS-7

See tables 9.2, 9.3

A.13.2.6.1 Parameter0

				D	T2.3, A1.3
1	Parameter0 supported			-	y
2	Universal-time	-	Universal class 23	-	y
3	Generalized-time	-	Universal class 24	-	y
4	boolean	-	Universal class 1	-	y
5	null	-	Universal class 5	-	y

A.13.2.6.2 Parameter1 (see section 9.10.1)

				D	T2.3, A1.3
1	Parameter1 supported			-	y
2	integer	-	Universal class 2	-	y
3	bit	-	Universal class 3	-	y
4	IA5	-	Universal class 22	-	y
5	GraphicString	-	Universal class 25	-	y
6	GeneralString	-	Universal class 27	-	y
7	OctetString	-	Universal class 4	-	y

A.13.2.6.3 Parameter2

		D	T2.3, A1.3
1	Parameter2 supported	-	o

A.13.2.7 NBS-8

See tables 9.2, 9.3

A.13.2.7.1 Parameter0

				Data Types		Key Type	
				D	T2.3, A1.3	D	T2.3, A1.3
1	Parameter0 supported			-	y	-	y
2	Universal-time	-	Universal class 23	-	y	-	y
3	Generalized-time	-	Universal class 24	-	y	-	y
4	boolean	-	Universal class 1	-	y	-	y
5	null	-	Universal class 5	-	y	-	y

A.13.2.7.2 Parameter1 (see section 9.10.1)

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter1 supported			-	y	-	y
2	integer	-	Universal class 2	-	y	-	y
3	bit	-	Universal class 3	-	y	-	y
4	IA5	-	Universal class 22	-	y	-	y
5	GraphicString	-	Universal class 25	-	y	-	y
6	GeneralString	-	Universal class 27	-	y	-	y
7	OctetString	-	Universal class 4	-	y	-	y

A.13.2.7.3 Parameter2

		Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter2 supported	-	o	-	o

A.13.2.8 NBS-11

See tables 9.2, 9.3

A.13.2.8.1 Parameter0

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter0 supported			-	y	-	y
2	Universal-time	-	Universal class 23	-	y	-	y
3	Generalized-time	-	Universal class 24	-	y	-	y
4	boolean	-	Universal class 1	-	y	-	y
5	null	-	Universal class 5	-	y	-	y

A.13.2.8.2 Parameter1 (see section 9.10.1)

				Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter1 supported			-	y	-	y
2	integer	-	Universal class 2	-	y	-	y
3	bit	-	Universal class 3	-	y	-	y
4	IA5	-	Universal class 22	-	y	-	y
5	GraphicString	-	Universal class 25	-	y	-	y
6	GeneralString	-	Universal class 27	-	y	-	y
7	OctetString	-	Universal class 4	-	y	-	y

A.13.2.8.3 Parameter2

		Data Types D	Data Types T2.3, A1.3	Key Type D	Key Type T2.3, A1.3
1	Parameter2 supported	-	o	-	o

A.13.2.9 NBS-12 (see section 10.7.1)

A.13.2.9.1 Universal class number parameter (see section 9.10.1)

				D	T2.3, A1.3	
1	Universal class number parameter supported			-	y	
2	PrintableString	-	Universal class 19	-	/	
3	TeletexString	-	Universal class 20	-	/	
4	VideotexString	-	Universal class 21	-	/	
5	IA5String	-	Universal class 22	-	y	
6	GraphicString	-	Universal class 25	-	y	see A.13.2.9.5
7	VisibleString	-	Universal class 26	-	y	
8	GeneralString	-	Universal class 27	-	y	see A.13.2.9.6

A.13.2.9.2 String length parameter

		D	T2.3, A1.3
1	Maximum string length parameter supported	-	y

A.13.2.9.3 String significance parameter

		D	T2.3, A1.3
1	String significance parameter supported	-	y
2	Variable length strings supported	-	y
3	Fixed length strings supported	-	y

A.13.2.9.4 Character set parameter

		D	T2.3, A1.3
1	Character set parameter supported	-	y

A.13.2.9.5 G sets supported

G sets which are supported in NBS-12 GraphicString.

1	**For values of GraphicString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character sets is required.** **(see 9.10.1.1, 9.10.1.3)**

A.13.2.9.6 G and C sets supported

G and C sets which are supported in NBS-12 GeneralString.

1	**For values of GeneralString only the support of character strings of the ISO 646, IRV (G0) and ISO 8859-1 (G0 and G1) character set and ISO 646 IRV (C0) control character sets is required.** **(see 9.10.1.1-3)**

- END OF PHASE 3 PROFILES REQUIREMENTS LIST -

10.11 APPENDIX B: FTAM PHASE 2 / PHASE 3 COMPATIBILITY

This appendix summarizes the functions and features which are defined for FTAM Phase 3 in addition to the FTAM Phase 2 specifications. It also states the degree of possible interworking and the backward compatibility.

Additional Requirements in FTAM Phase 3	Backward Compatibility to FTAM Phase 2
FTAM-1: GraphicString, VisibleString FTAM-2: VisibleString concurrency-control parameter for Initiator create-password parameter for Initiator	(for further study)

Additional Optional Features in FTAM Phase 3	Backward Compatibility to FTAM Phase 2
FTAM-2: GeneralString, IA5String	
FTAM-4	(for further study)
NBS-8 in T2.3, A1.3	
NBS-10	
NBS-11	
NBS-12	
Recovery functional unit	
Restart-data-transfer functional unit	
FADU-locking functional unit and FADU-lock parameters in A1.3, A2.3	
concurrency-control parameters for Responder	
create-password parameter for Responder	
location-field of access-control element	
Enhanced-file-management functional unit in conjunction with transfer or access service class	
suggested-delay term of diagnostic parameter supported conditionally on Recovery or Restart-data-transfer functional units	
Profiles A1.3, A2.3 do not require transfer service class	
no minimum requirement for maximum-string-length parameters for document types	(for further study)

10.12 APPENDIX C: DOCUMENT TYPES

NBS-10 Random Binary Access Document Type

1. Entry Number: NBS-10

2. Information objects

Table 10.2 Information objects in NBS-10

document type name	{iso identified-organization icd(9999) organization-code(1) document type(5) random-binary(10)} "NBS-10 random binary access file"
abstract syntax names: a) name of asname1 b) name of asname2 c) name of asname3	{iso identified-organization icd(9999) organization-code(1) abstract- syntax(2) nbs-random-binary(4)} "NBS random binary access file abstract syntax" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU" {iso identified-organization icd(9999) organization-code(1) abstract- syntax(2) nbs-node-name(3)} "NBS random access node name abstract syntax"
transfer syntax names:	{joint-iso-ccitt asn1(1) basic-encoding (1)} "Basic encoding of a single ASN.1 type"
file model	{iso standard 8571 file-model (3) hierarchical (1)} "FTAM hierarchical file model"
constraint set	{iso identified-organization icd(9999) organization-code(1) constraint-set(4) nbs-random-access(2)} "NBS random access constraint set"

File contents:

```
Datatype1 ::= a single octet

Datatype2 ::= Node-Name
 --The type to be used for Node-Name is defined in
 --ISO 8571-FADU
 --The only Choice for Node-Name is user-coded

Datatype3 ::= NBS-Node-Name
 --As defined by the NBS Node Name Abstract Syntax
```

3. Scope and field of application

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

4. References

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

5. Definitions

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

6. Abbreviations

FTAM File Transfer, Access and Management

7. Document semantics

The document consists of zero, one or more file access data units each of which consists of one data element. The data element is made up of one octet. The order of these elements is significant. The semantics of the data elements is not specified by this document type.

The document structure takes the form allowed by the FTAM hierarchical file model as constrained by the NBS random access constraint set. The definition for FTAM hierarchical file model appears in 8571-2.

There are no size or length limitations imposed by this definition.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a series of octets.

9. Definition of transfer

9.1. Datatype definition

The presentation data value used for transfer is an ASN.1 OCTET STRING.

Datatype 2 is used to specify the FADU-Identity of "single-name" in the FTAM PDUs specifying FADU-Identity, where "single-name" is defined as an EXTERNAL. The EXTERNAL is defined as Node-Name in the FTAM FADU abstract syntax. The use of Datatype2 is defined in "NBS random access constraint set".

Datatype3 specifies the "user-coded" form of the Node-Name in the FTAM FADU abstract syntax, where "user-coded" is defined as an EXTERNAL. That EXTERNAL is defined by Datatype3. The use of Datatype3 is defined in "NBS random access constraint set".

9.2 Presentation data values

The document is transmitted as a series of presentation data values. Each presentation data value shall consist of the "data" from one or more FADUs concatenated together. The result is one value of the ASN.1 data type OCTET STRING. The "fadu_count" field supplied in the Node-Name specifies the number of FADUs to transfer during a Read operation. The requested FADUs may be transferred as one or more presentation data values.

All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in Table 10.2.

Note: Specific carrier standards may impose additional constraints on the presentation context to be used, when the above permits a choice.

Boundaries between P-DATA primitives and between presentation data values are chosen locally by the sending entity at the time of transmission. The boundaries are not preserved when the file is stored and they carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

9.3 Sequence of presentation data values

The sequence of presentation data values is the same as the sequence of Data Units within the file.

10. Transfer syntax

An implementation supporting these document types shall support the transfer syntax generation rules named in Table 10.2 for all presentation data values transferred.

Implementations may optionally support other transfer syntaxes.

11. ASE specific specifications

11.1 Simplification and relaxation

The document type NBS-10 may be simplified to the document type FTAM-3. The resultant document contains the same sequence of data values as would result from accessing the file as an NBS-10 file.

11.2 The READ operation

A READ operation may be applied to a range of FADUs via the FADU Identity of "NodeSeq". The "starting-fadu" part of the node name specifies the node number of the first FADU; the "fadu-count" specifies the number of consecutive FADUs to be transferred.

A READ operation applied to a range of FADUs that spans beyond the end of file is valid. All available data in the range is transferred. An informative diagnostic (5005) is returned on the F-Data-End Request indicating that the end of file was reached and a portion of the request was satisfied.

11.3 The REPLACE operation

When the REPLACE operation is applied to the root FADU of an NBS-10 document, the transferred data shall be any NBS-10 document.

The REPLACE operation applied to a FADU identity of "node number" is used to replace a series of FADUs, starting at the specified position in the file, by the new FADUs being transferred. The number of replaced FADUs is determined by the number of transferred FADUs.

If the replacement spans beyond the end of the existing file, then the additional FADUs are inserted at the end of the file.

11.4 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred data shall be a series of FADUs which would be generated by reading any NBS-10 document type in access context UA.

NBS-11 Indexed Sequential File With Unique Keys

1. **Entry Number: NBS-11**
2. **Information objects**

Table 10.3 Information Objects in NBS-11

document type name	{iso identified-organization icd (9999) organization-code (1) document type (5) indexed-file-with-unique-keys (11)} "NBS-11 FTAM indexed file with unique keys"
abstract syntax names: a) name for asname1 b) name for asname2	{iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-as1 (1)} "NBS abstract syntax AS1" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1) } "Basic Encoding of a single ASN.1 type"

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

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

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

             --  Parameter0, Parameter1, Parameter2, as defined for the
             --  document types NBS-6, NBS-7, NBS-8

  KeyPosition::= INTEGER
```

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

```
file contents:
              Datatype1 ::= PrimType -- as defined in Annex 9 A, Part 3
                                     of NIST SP 500-162
              Datatype2 ::= CHOICE   { Node-Descriptor-Data-Element,
                                       Enter-Subtree-Data-Element }
                                       Exit-Subtree-Data-Element }
```

3. **Scope and field of application**

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

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

4. **References**

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

5. **Definitions**

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

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

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

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the FTAM ordered flat constraint set with unique names (see Table 10.3). These definitions appear in ISO 8571-2.

The following additional requirements are specified for the use of the ordered flat constraint set with unique names:

- The FADU identity 'node number' is not required for conformant implementations
- The identities 'next' and 'previous' are allowed for all FADUs

Each data element is a data type from the set of primitive data types defined in Appendix 9A, Part 3 of NIST 500-162. Each data unit contains the same data element types in the same order as all other data units. These types and their respective maximum lengths are defined by the <DataTypes> parameter.

The string-length field of Parameter 1 specifies the length of the value in octets for the INTEGER, BIT STRING and OCTET STRING types. For character-type data elements, the string-length indicates the actual number of characters from the specified character set, not including any escape sequences or overhead from the character encoding.

For floating point numbers, finite form, length-1 and length-2 specify the length in bits of mantissa and exponent, respectively. The length-1 and length-2 values are irrelevant for the other choices of floating point numbers.

Each data unit in the file has a key associated with it. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in Appendix 9A, Part 3 of NIST 500-162.

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

The primitive data types and minimum size ranges of each unit which an implementation must accept as a key value are given in the following Table 10.4.

Table 10.4 Datatypes for keys

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

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

8. Abstract syntactic structure

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

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in Table 10.3, where the PrimType in the datatype is given by the NBS-AS1 definition; or

b) Datatype2 defined in Table 10.3, which is the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO8571-FADU.

9.2 Presentation data values

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

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

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

Notes:

1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

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

9.3 Sequence of presentation data values

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

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 10.2 for all presentation data values transferred. Implementation may optionally support other named transfer syntaxes.

11. ASE specific specifications for FTAM

11.1 Simplification and relaxation

11.1.1 Structural simplification

This simplification loses information.

The document type NBS-11 may be accessed as a document type FTAM-3 (allowed only when reading the file) by specifying document type FTAM-3 in the <contents type> parameter in <F-OPEN request>, and limiting access context to UA on F-READ.

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

A document of type NBS-11 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 with appropriate data type parameters in the <contents type> parameter on the <F-OPEN request>. The traversal order of the FADUs must be maintained.

Note: The traversal order is as reading the file as NBS-11 in key order.

A document of type NBS-11 may be accessed as a document of type NBS-8 (allowed only when reading the file) by specifying document type NBS-8 in the <contents type> parameter in the <F-OPEN REQUEST>.

11.2 Access context selection

A document of type NBS-11 may be accessed in any one of the access contexts defined in the FTAM ordered flat constraint set with unique names. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the <INSERT> operation is applied the transferred material shall be the series of FADU which would be generated by reading any NBS-11 document with the same parameter values in access context FA.

A transferred FADU whose name duplicates that of an already existing FADU will cause the <INSERT> operation to fail. The failure shall be signalled by issuing an F-CANCEL Request with a corresponding diagnostic.

11.4 The EXTEND operation

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

11.5 The REPLACE operation

When the <REPLACE> operation is applied with FADU Identity 'begin', a transferred FADU whose name duplicates that of a previously transferred FADU will cause the <REPLACE> operation to fail. The failure shall be signalled by issuing an F-CANCEL Request with a corresponding diagnostic.

NBS-12 Simple Text File Document Type

1. Entry Number: NBS-12

2. Information objects

Table 10.5 Information objects in NBS-12

document type name	{iso identified-organization icd (9999) organization-code (1) document-type (5) simple-text-file (12)} "NBS-12 FTAM simple text file"
abstract syntax names: a) name for asname1 b) name for asname2	 {iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-simple-text (5)} "NBS simple text abstract syntax" {iso standard 8571 abstract-syntax(2) ftam-fadu (2)} "FTAM FADU"
transfer syntax names:	{joint-iso-ccitt asn1 (1) basic-encoding (1)} "Basic Encoding of a single ASN.1 type"

Parameter Syntax

```
PARAMETERS ::= SEQUENCE{

    universal-class-number [0] IMPLICIT INTEGER,
    maximum-string-length  [1] IMPLICIT INTEGER,
    string-significance    [2] IMPLICIT INTEGER {variable (0), fixed (1)},
    character-set          [3] IMPLICIT OctetString OPTIONAL}
```

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

File contents

```
            Datatype1 ::= NBS Text
                                  --as defined in the NBS Simple Text
                                  --Abstract Syntax registration entry


            Datatype2 ::= Node-Descriptor-Data-Element
```

3. **Scope and field of application**

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

4. **References**

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

ISO 8824, Information Processing Systems-Open Systems Interconnection-Specification of Abstract Syntax Notation 1 (ASN.1).

ISO 8825, Information Processing Systems-Open Systems Interconnection-Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

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

5. **Definitions**

This definition makes use of the terms data element, data unit and file access data unit as defined in ISO 8571-1. In addition, it makes use of the terms character string, graphics character, and format effector as defined in document type registration entry "FTAM-2" in ISO 8571-2.

6. **Abbreviations**

FTAM File Transfer, Access and Management

7. **Document semantics**

This document consists of zero, one or more file access data units, each of which consists of one character string. The order of each of these elements is significant. The semantics of the character strings is not specified by this document type.

The document structure takes any of the forms allowed by the FTAM hierarchical file model as constrained by the sequential flat constraint set. These definitions appear in ISO 8571-2. As additional constraints FADU identity will be limited to the following values:

a) 'begin' and 'end' when using the Transfer or Transfer and Management service classes.

b) 'begin', 'end', 'first', and 'next' when using the Access service class.

Each character string consists of characters from the character set defined by the ASN.1 (ISO 8824) character set type whose universal class number is given by the "universal-class-number" parameter and by the escape sequences contained in the optional "character-set" parameter. If the character set type allows explicit escape sequences, the "character-set" parameter, if present, contains escape sequences which designate and invoke specific character sets. If the "character-set" parameter is not present, character sets are assumed to be designated and invoked as specified in Table 2 in ISO 8825. Character strings shall not contain escape sequences.

There are no size or length limitations imposed by this definition, except those specified here. Each character string is of a length determined by the number of characters given by the "maximum-string-length" parameter.

Note: The length restriction refers to the number of characters from the applicable character set, not to the number of octets in the encoding, nor to the line length in any rendition of the document, where these are different.

The exact significance of the character strings is determined by the "string-significance" parameter. If its value is "variable", the length of the character strings is less than or equal to the length given. If the value is "fixed", the length of each character string is exactly equal to the length given.

If the document is interpreted on a character imaging device (outside the scope of ISO 8571), the interpretation depends on the character set in use.

a) If the character set contains format effectors, they shall be interpreted as defined in ISO 6429; end of string and end of file access data unit are given no formatting significance, and do not contribute to the document semantics;

b) If the character set does not contain format effectors, the end of each character string is interpreted as implying carriage return and line feed formatting actions in any rendition. The end of file access data unit is given no formatting significance beyond that attached to the end of the string in it.

8. Abstract syntactic structure

The abstract syntactic structure of the document is a hierarchically structured file as defined in the ASN.1 modules ISO8571-FADU and ISO 8571-CONTENTS in ISO 8571, in which each of the file contents data elements has the abstract syntactic structure of "NBS Simple Text."

9. Definition of transfer

9.1 Datatype definitions

The file consists of data values which are of either

a) Datatype1 defined in Table 10.5, the ASN.1 datatype declared as "NBS-Text" in the NBS Simple Text Abstract Syntax definition. The choice in "NBS-Text" is determined by the universal-class-number parameter; or

b) Datatype2 defined in Table 10.5, the ASN.1 datatype declared as "Data-Element" in the ASN.1 module ISO 8571-FADU.

9.2 Presentation data values

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

a) one value of the ASN.1 datatype "Datatype1", carrying one of the character strings of the document. Each character shall be transmitted using one of the character sets identified by the universal-class-number parameter. All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname1" declared in Table 10.5, or

b) one value of the ASN.1 datatype "Datatype2". All values are transmitted in the same (but any) presentation context established to support the abstract syntax name "asname2" declared in Table 10.5.

Notes: 1. Specific carrier standards may impose additional constraints on the presentation context to be used, where the above permits a choice.

2. Any document type defined in this entry either makes no use of Datatype2, or starts with a Datatype2 transmission.

Boundaries between P-DATA primitives are chosen locally by the sending entity at the time of transmission, and carry no semantics of the document type. Receivers which support this document type shall accept a document with any of the permitted transfer options.

9.3 Sequence of presentation data values

The sequence of presentation data values of type (a) and the sequence of presentation data values of types (a) and (b) is the same as the sequence of character strings within a Data Unit, and Data Units in the hierarchical structure, when flattened

according to the definition of the hierarchical file model in ISO 8571-2.

10. Transfer syntax

An implementation supporting this document type shall support the transfer syntax generation rules named in Table 10.5 for all presentation data values transferred.

11. ASE specific specifications

11.1 Simplification and relaxation

11.1.1 Simplification to FTAM-1

This simplification loses information.

The document type NBS-12 may be accessed as a document type FTAM-1. The resultant document contains the same sequence of data values as would result from accessing the structured text file in access context UA. That is, only the presentation data values in the abstract syntax "asname1" are present. If the "character-set" parameter was present before the simplification, its contents will be added to the beginning of each string.

Note: The boundary between file access data units remains a boundary between strings, but any special significance given to it is lost.

11.1.2 Relaxation to FTAM-2

The document type NBS-12 may be relaxed to the document type FTAM-2. If the "character-set" parameter was present before the relaxation, its contents will be added to the beginning of each string.

11.1.3 Character set relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to a different document of type NBS-12 with

- a different "universal-class-number" parameter value,
- a different "character-set" parameter value,
- different values for both of these parameters,

- a different "universal-class-number" parameter value and no "character-set" parameter value, or
- no "character-set" parameter value,

if the resultant document type permits all characters from the original document type. If this relaxation involves including format effectors and none were present before the simplification, the characters "carriage return" and "line-feed" shall be added to the end of each string.

Note: If the characters "carriage return" and "line feed" are not part of the format effectors, the formatting action may be represented by "newline", or some other implementation specific choice if there is no representation of "newline" defined.

11.1.4 String length relaxation

This operation loses explicit information in the document type identification.

A document of type NBS-12 may be relaxed to another document type NBS-12 with a larger "maximum-string-length" parameter.

11.2 Access context selection

A document of type NBS-12 may be accessed in any one of the access contexts defined in the sequential flat constraint set. The presentation data units transferred in each case are those derived from the structuring elements defined for that access context in ISO 8571-2.

11.3 The INSERT operation

When the INSERT operation is applied at the end of file, the transferred material shall be the series of FADUs which would be generated by reading any NBS-12 document type with the same parameter values in access context FA.

10.13 APPENDIX D: CONSTRAINT SETS

NBS Random Access Constraint Set

Table 10.6 - Basic Constraints in the NBS Random Access Constraint Set

Constraint set descriptor	NBS Random Access
Constraint set identifier	{iso identified-organization icd(9999) organization-code(1) constraint-set(4) nbs-random-access(2)}
Node names	All names shall be of the same type; the type of the names and an ordering of the names shall be defined when reference is made to the constraint set.
File access actions	Locate, Read, Insert, Erase, Replace
Qualified actions	None
Available access context	UA
Creation state	Root node without an associate data unit
Location after open	Root node
Beginning of file	Root node
End of file	No node selected
Read whole file	Read in access context UA with FADU-Identity of"begin"
Write whole file	Transfer a series of leaf FADUs which would be generated by reading the whole file in access context UA; Perform the transfer with an FADU Identity of "end" and a file access action of "insert", or with an FADU Identity of "begin" and an action of "replace", or with an FADU Identity of "node-number" and an action of "replace". Here "node number" identifies the first FADU in the preorder traversal sequence.

Table 10.7 - Identity Constraints in the NBS Random Access Constraint Set

Action	Begin	End	NodeSeq	Node Number
Locate				leaf
Read	whole		leaf	
Insert		leaf		
Erase	whole			leaf
Replace	whole			leaf

Note: NodeSeq = A sequence of node names with a single member

1. Field of application

The NBS Random Access constraint set applies to files which are structured into a sequence of individual FADUs and to which access may be made randomly by NodeSeq. The structuring of the file into individual FADUs is determined by the NodeName.

2. Basic constraints

The basic constraints in the NBS Random Access constraint set are given in Table 10.6.

3. Structural constraints

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

4. Action constraints

Insert: the insert action is allowed only at the end of the file, with FADU-Identity of "end"; the new node is inserted following all existing nodes in the file. The location following the insert is "end".

Erase: the erase action is allowed at the root node to empty the file, with FADU-Identity of "begin". The result is a solitary root node without an associated data unit. Erase with the FADU-Identity of "node number" means truncation of the file.

Replace whole file: the FADU-Identity is "begin" and the complete series of new FADU contents is sent.

Replace new leaves: the FADU-Identity is "node number" and the number of FADUs being replaced is given by the number of FADUs sent.

5. Identity constraints

The FADU-Identity associated with the file action shall be one of the identities: begin, end, Node Number and NodeSeq. The actions with which these identities can be used are given in Table 10.7.

10.14 APPENDIX E: ABSTRACT SYNTAXES

NBS Node Name Abstract Syntax

Abstract Syntax Name
{ iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-node-name (3) }

"NBS random access node name abstract syntax"

This is an abstract syntax for the user-coded Node-Name in the FTAM FADU abstract syntax.

```
NBS-AS3 DEFINITIONS::=

BEGIN

    NBS-Node-Name::== SEQUENCE

        {   starting-fadu [0] IMPLICIT INTEGER,
            fadu-count [1] IMPLICIT INTEGER }
                --a "fadu-count" of 0 specifies the
--range of FADUs
                --beginning at "starting-fadu" and      - -
ending at "end of file"

END
```

For this abstract syntax the following transfer syntax will be used.

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

NBS Random Binary Access File Abstract Syntax

Abstract Syntax Name
{ iso identified-organization icd (9999) organization-code (1) abstract-syntax (2) nbs-random-binary (4) }

"NBS random binary access file abstract syntax"

This is an abstract syntax for the transfer of the file contents for NBS Random binary files.

```
NBS-AS4 DEFINITIONS::=

BEGIN
    NBS-Random Binary ::== OCTET STRING
        --contains one or more presentation data values
        --concatenated together.
        --Each presentation data value is defined as
        --Datatype1 in Table 10.2.
```

```
        END
```

For this abstract syntax the following transfer syntax will be used.

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

NBS Simple Text Abstract Syntax

```
Abstract Syntax Name
        {iso identified-organization icd (9999) organization-code(1)
        abstract-syntax (2) nbs-simple-text(5) }
        "NBS simple text abstract syntax"

        NBS-AS5 DEFINITIONS::==

        BEGIN

        NBS-Text::= CHOICE {

                        IA5String,--Universal Class 22
                        GraphicString,--Universal Class 25
                        VisibleString,--Universal Class 26
                        GeneralString--Universal Class 27}

        END
```

For this abstract syntax, the following transfer syntax will be used:

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

11. DIRECTORIES

11.1 INTRODUCTION

Refer to Section 11.1 of Stable Agreements Version 2 Edition 3.

11.2 SCOPE AND FIELD OF APPLICATION

Refer to Section 11.2 of Stable Agreements Version 2 Edition 3.

11.3 STATUS

This version completed June, 1989.

11.4 USE OF DIRECTORIES

11.4.1 Introduction

(See Stable Document for current information.)

11.4.2 MHS

(TBD)

11.4.3 FTAM

(TBD)

11.5 DIRECTORY ASEs, APPLICATION CONTEXTS, AND PORTS

Refer to Section 11.5 of Stable Agreements Version 2 Edition 3.

11.6 SCHEMAS

Refer to Section 11.6 of Stable Agreements, Version 2, Edition 3.

11.6.1 Support of Structure and Naming Rules

Refer to Section 11.6.1 of Stable Agreements, Version 2, Edition 3.

11.6.2 Support of Object Classes and Subclasses

DSAs shall be able to support storage and use of the object classes below, as defined in the Directory Documents, Part 7.

The following object classes are mandated by the the standard:

Top Alias DSA

The following object classes are expected to be generally useful in the creation of the upper portion of the DIT:

Country	Organization
Locality	Organizational Unit

The following object classes are expected to be generally useful in the creation of DIT leaf entries:

Alias	Group of Names
Application Process	Organizational Person
Application Entity	Organizational Role
DSA	Residential Person
Device	

The DSAs shall be be able to support all superclasses of the supported object classes (e.g. Top, Person).

Use of an object class in this profile or the standard (or a subclass derived from one or more of these object classes) is recommended wherever the semantics is appropriate for the application. The derviation of a new object class as an immediate subclass of Top should be avoided. For example, to represent printers in the Directory, one can derive a subclass of Device.

An entry of a particular object class may contain any optional attribute listed for it in ISO 9594; and a conformant DSA must be able to support all these optional attributes.

In addition, a DSA may permit any locally-registered attribute, or a subset of these, by invoking the local extension facilities permitted by unregistered object classes (viz. ISO/IEC/9594-2) clause 9.4.1 a) and Note).

11.6.2.1 Strong Authentication Profile

The following object classes are expected to be generally useful for applications to support strong authentication:

Strong Authentication User
Certification Authority

11.6.3 Support of Attribute Types

Refer to Section 11.6.3 of Stable Agreements, Version 2, Edition 3.

DSAs must support the encoding, decoding, and matching of all the attributes in the Naming Prefixes of every naming context they hold (ref ISO 9594-4 para 9). These attribute may include attributes that are not permitted to appear in entries in those naming contexts.

11.6.4 Support of Attribute Syntaxes

Refer to Section 11.6.4 of Stable Agreements, Version 2, Edition 3.

11.6.5 Naming Contexts

The root of a naming context must not be an alias entry.

11.7 CLASSIFICATION OF SUPPORT FOR ATTRIBUTE TYPES

Refer to Section 11.7 of Stable Agreements, Version 2, Edition 3.

11.8 INTRODUCTION TO PRAGMATIC CONSTRAINTS

Refer to Section 11.8 of Stable Agreements Version 2 Edition 3.

11.9 GENERAL CONSTRAINTS

Refer to Section 11.9 of Stable Agreements Version 2 Edition 3.

11.10 CONSTRAINTS ON OPERATIONS

Refer to Section 11.10 of Stable Agreements Version 2 Edition 3.

11.11 CONSTRAINTS ON ATTRIBUTE TYPES

Refer to Section 11.11 of Stable Agreements Version 2 Edition 3.

11.11.1 Attribute Values

Integer Values

DSAs shall be required to "pass through" encoded integer attribute values of arbitrary length (e.g. when chaining a Directory operation). No Directory component (i.e. DUA or DSA) shall be deemed non-conformant if it encodes integer attribute values of arbitrary length.

Components of the Directory are required to support (for storage and processing), as a minimum, integer attribute values encoded in 4 octets.

11.12 CONFORMANCE

Refer to Section 11.12 of Stable Agreements, Version 2, Edition 3.

11.13 DISTRIBUTED OPERATIONS

Refer to Section 11.13 of Stable Agreements, Version 2, Edition 3.

11.13.1 Referrals and Chaining

Refer to Section 11.13.1 of Stable Agreements, Version 2, Editon 3.

11.13.2 Trace Information

A **Traceinformation** value carries forward a record of the DSAs which have been involved in the performance of an operation. It is used to detect the existence of, or avoid, loops which might arise from inconsistent knowledge or from the presence of alias loops in the DIT.

Each DSA which is propagating an operation to another, adds a new item to the trace information. If the propagation of a Search operation involves the creation of a new Search (cf. IS 9594-4 18.7.2.2.2), the trace information must not be re-set, but the full trace information for the overall Search operation to the point where the new Search was generated must be included in the new Search.

11.14 UNDERLYING SERVICES

Refer to Section 11.14 of Stable Agreements Version 2 Edition 3.

11.15 ACCESS CONTROL

Refer to Section 11.15 of Stable Agreements Version 2 Edition 3.

11.16 TEST CONSIDERATIONS

Refer to Section 11.16 of Stable Agreements Version 2 Edition 3.

11.17 ERRORS

Refer to Section 11.17 of Stable Agreements Version 2, Edition 3.

11.18 DSA CHARACTERISTICS

(TBD)

11.19 APPENDIX A: MAINTENANCE OF ATTRIBUTE SYNTAXES

11.19.1 Introduction

Please refer to Appendix A from Stable Agreements Version 2 Edition 3.

11.19.2 General Rules

For description of general rule information, refer to the aligned Section 11.19.2 of the Stable Implementation Agreements, Version 2, Edition 3.

The following rule is proposed to simplify the handling of attributes:

1) The T.61 string type shall be further constrained to contain no characters other than defined graphic characters and spaces. Character set restrictions shall be specified in Table 11.1.

Table 11.1: Charater Set Restrictions Upper 4 bits of encoding (hex)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	X	X					X		X	X	X		X	X		
1	X	X							X	X				X		
2	X	X							X	X				X		
3	X	X							X	X				X		
4	X	X							X	X				X		
5	X	X							X	X				X	X	
6	X	X							X	X				X		
7	X	X							X	X				X		
8	X	X							X	X				X		
9	X	X							X	X	X	X		X		
A	X	X							X	X	X	X		X		
B	X	X						X	X	X				X		
C	X	X				X			X	X	X			X		
D	X	X						X	X	X	X			X		
E	X	X				X		X	X	X	X			X		
F	X	X						X	X	X	X			X		X

Notes: 1. Row headings give the lower 4 bits of the encoding in hexadecimal.

2. Entries marked X are illegal T.61 encodings.

Prohibition of the use of and support of recursive distinguished names is for further study.

11.19.3 Checking Algorithms

Please refer to Appendix A from Stable Agreements Version 2 Edition 3.

11.19.4 Matching Algorithms

Please refer to Appendix A from Stable Agreements Version 2 Edition 3.

11.20 APPENDIX B: GLOSSARY

Please refer to Appendix B from Stable Agreements Version 2 Edition 3.

11.21 APPENDIX C: REQUIREMENTS FOR DISTRIBUTED OPERATIONS

Please refer to Appendix C from Stable Agreements Version 2 Edition 3.

11.22 APPENDIX D: REGISTRATION AND USAGE OF OBJECT CLASSES

11.22.1 Introduction

This tutorial material is included because the SIG felt that it was useful clarification (of the Directory documents) to Implementors on matters that could not be deferred. However, implementors should be advised that the material is the subject of change/enhancement in the tandards and lies in an area of substantial instability.

The objective of the tutorial is to clarify how structure rules need to be related to object classes (whether or not a DSA polices structure rules), and the way in which DSAs can administrate entries in relation to the Object Classes which they support.

11.22.2 Primary and Secondary Object Classes

Object classes specify the nature and properties of entries, in terms of the attributes which they must (or may) possess, and also in terms of their possible positions in the DIT and the names that they may have.

Primary object classes define the nature and role of objects, and therefore of the corresponding Directory entries. A Primary object class will normally be associated with a structure rule. Thus, "Country", "Device", "Person" are Primary (although "Person" does not possess a structure rule).

Secondary object classes, by contrast, only qualify Primary object classes, by adding new mandatory or optional attributes. A Secondary Object Class will never be associated with a structure rule. "MHS-User", "Top", "Alias" are Secondary.

The "multiple inheritance" provisions of the Directory Documents enables any particular object (and associated entry) to be defined by zero or more Secondary Object Classes, and by one and just one Primary Object Class. (The rule specifying that there must be just one Primary object class prevents ambiguity in the source of the structure rules.)

Define an Object Class Component as that new information which a particular Object Class adds to the Object Classes of which it is

a subset. The Object Class macro is what defines the Object Class Component.

Then, the following rules apply to the derivation of new Object Classes, in accordance with the Directory Documents.

A. Recursive Object Class definitions are forbidden (e.g. an object class may not have itself as a superset).

B. A new Primary Object Class can be derived by the use of superclasses comprising any set of Object Classes if its own Object Class Component defines any structure rules for the Object Class. This allows the derivation of a completely new class of object class, while making use of existing object class definitions.

C. A new Primary Object Class can also be derived by the use ofsuperclasses comprising a single Primary Object Class, and zero, one or more Secondary Object Classes, by inheriting the structure rules associated with the Primary Object Class. This allows the derivation of a related Object Class, and forbids the ambiguity in derivation ofstructure rules that would arise from having more than one Primary superclass.

D. Unregistered Object Classes (i.e. those to which no distinct object identifier is allocated) must always be Primary Object Classes derived in accordance with rule C. That is, the unregistered Object Class Component must not contain structure rules of its own. This prevents the use of unregistered Object Classes which do not obey the structure rules associated with other objects which share the same set of Object Class attribute values.

E. Secondary Object Classes can be derived by the use of superclasses comprising any set of Secondary Object Classes - there can be no structure rules associated with Secondary object Classes.

F. Entries may only be created with an Object Class which is Primary and possesses structure rules. This says that all entries must have structure rules.

11.22.3 Locally Registered Object Classes

A particular DSA is not required to support all Object Classes. It may contain a registry of the object classes which it does support.

The rules above enable the registry to be defined in terms of the locally registered Primary Object Classes which it supports. Each of these can be defined in terms of the single object identifier which represents that Object Class. (Of course, any entry defined with this Object Class contains an attributes whose values include not only the corresponding object identifier, but also the identifiers associated with each of the Object Class's superclasses.)

Associated with each locally registered Primary Object Class could be a list of secondary Object Classes which may be permitted to be used in association with this Primary Object Class. When a new entry is created, its Object Class attributes can then be analysed to determine:

Whether the entry's Object Class attribute is compatible with local registration

The Primary Object Class to which it conforms

The structure rules to which it must conform

The Secondary Object Classes (if any) to which it must conform. Given this analysis, the name and attributes of the entry can be analysed to determine its compatibility with the local registry of Primary Object Classes.

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 2, Edition 3.

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 2, Edition 3, June 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

The X.3 profile will support functionality similar to the CCITT recommendations and could be used to implement an X.3 to ISO-VT gateway.

The Page profile is 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 is complete and was stabilized in December, 1988. See Stable Agreements.

14.3.3 Status of Phase II

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

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 NIST REGISTERED CONTROL OBJECTS

See Stable Agreements.

14.8 NIST 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.

14.8.4 Scroll Profile

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

14.8.4.1 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 unidirectional, forward only.

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".

14.8.4.2 Association Requirements

14.8.4.2.1 Functional Units

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

14.8.4.2.2 Mode

This profile operates in A-mode.

14.8.4.3 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 = "higher only",
            y-absolute = "no",
            y-window = 0
        },

    erasure-capability = "yes",

    *( repertoire-capability is implied by the number of
    occurences 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 = 0
        },
    erasure capability = "yes",

    *( repertoire-capability is implied by the number of
    occurences 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".

14.8.4.4 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.

14.8.4.5 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.

14.8.4.6 Profile Notes

14.8.4.6.1 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).

14.8.4.6.2 Informative Notes

1. This profile models a scrolling device with scrolling only in the forward direction. The display pointer may not 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 terminateion 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.

14.8.4.7 Specific Conformance Requirements

None.

14.8.5 X3 Profile

NIST VTE-Profile X3-1989 (r1, r2, r3, r4, r5)

14.8.5.1 Introduction

This profile provides support for CCITT X.3 PAD compatible operation.

The purpose of this profile is two-fold:

- to provide a transitional environment for applications that assume the availability of X.3 parameters with which to control the behavior of the terminal-system.
- to facilitate a gateway function between ISO-VTP and X.3.

14.8.5.2 Association Requirements

14.8.5.2.1 Functional Units

The Structured CO Functional Unit is mandatory.

The Urgent Data Functional Unit is optional.

14.8.5.2.2 Mode

This is an A-mode profile.

14.8.5.3 Profile Body

```
Display-objects  =
{
    {
    display-object-name = D1,
    DO-access           = profile-argument-r1,
    dimensions      = "one",
        x-dimension =
        {
            x-bound     = "unbounded",
            x-addressing    = "not-permitted",
            x-absolute  = "no",
            x-window        = 0
        },
    repertoire-assignment   = <ESC> 2/5 2/15 4/2
                        *( VTS Transparent Set )*
    },
```

```
    {
    display-object-name = D2,
    DO-access           = opposite of profile-argument-r1,
    dimensions      = "one",
        x-dimension =
        {
            x-bound     = "unbounded",
            x-addressing    = "not-permitted",
            x-absolute  = "no",
            x-window        = 0
        },
    repertoire-assignment   = <ESC> 2/5 2/15 4/2
                        *( VTS Transparent Set )*
    },
},


Control-objects =
{
    { *( PAD -
    Each element of the PAD CO represents a CCITT PAD
    parameter.  The CO-element-id of each element has been
    chosen so that it would be same value as the CCITT PAD
    parameter number that it represents.  The PAD CO is
    used both to set CCITT PAD parameter-equivalent values
    and to reply to an update to the READ CO.  The access
    rights are modified as follows: initially, the access
    rights are assigned the opposite of profile-argument-
    r1, but are changed to the value of profile-argument-r1
    when the READ CO is received, thus enabling the reply
    to the read to be satisfied.  Once the reply is made,
    the access rights revert to the opposite of profile-
    argument-r1. )*
    CO-name     = PAD,
    CO-structure    = 22,
    CO-access       = "NSAC",
    CO-priority = "normal",
    CO-trigger  = "not-selected",
    {   *( X.3 parameter 1 -- PAD recall )*
        CO-element-id   = 1,
        CO-category = "transparent",
        CO-size     = 8 },
    {   *( X.3 parameter 2 -- PAD echo )*
        CO-element-id   = 2,
        CO-category = "boolean",
        CO-size     = 1 },
    {   *( X.3 parameter 3 -- Data Forwarding Character )*
        CO-element-id   = 3,
        CO-category = "boolean",
        CO-size     = 7 },
```

```
{   *( X.3 parameter 4 -- Idle Timer Delay )*
    CO-element-id   = 4,
    CO-category = "integer",
    CO-size     = 255 },
{   *( X.3 parameter 5 -- Ancillary Device Control )*
    CO-element-id   = 5,
    CO-category = "boolean",
    CO-size     = 1 },
{   *( X.3 parameter 6 -- Control of PAD Signals )*
    CO-element-id   = 6,
    CO-category = "transparent",
    CO-category = 4 },
{   *( X.3 parameter 7 -- PAD on receipt of Break )*
    CO-element-id   = 7,
    CO-category = "boolean",
    CO-size     = 5 },
{   *( X.3 parameter 8 -- Discard Output )*
    CO-element-id   = 8,
    CO-category = "boolean",
    CO-size     = 1 },
{   *( X.3 parameter 9 -- Padding After <CR> )*
    CO-element-id   = 9,
    CO-category = "integer",
    CO-size     = 7 },
{   *( X.3 parameter 10 -- Line Folding )*
    CO-element-id   = 10,
    CO-category = "integer",
    CO-size     = 255 },
{   *( X.3 parameter 11 -- Device Speed )*
    CO-element-id   = 11,
    CO-category = "symbolic",
    CO-category = 19 },
{   *(X.3 parameter 12 -- Flow Control by Device )*
    CO-element-id   = 12,
    CO-category = "boolean",
    CO-size     = 1 },
{   *( X.3 parameter 13 -- Insert <LF> after <CR> )*
    CO-element-id   = 13,
    CO-category = "boolean",
    CO-size     = 3 },
{   *( X.3 parameter 14 -- Linefeed Padding )*
    CO-element-id   = 14,
    CO-category = "integer",
    CO-size     = 7 },
{   *( X.3 parameter 15 -- Editing )*
    CO-element-id   = 15,
    CO-category = "boolean",
    CO-size     = 1 },
```

```
{   *( X.3 parameter 16 -- Character Delete )*
    CO-element-id   = 16,
    CO-category = "character",
    CO-repertoire-assignment    *( any from C0 )*
                = "void", "void", <ESC> 2/1 4/0,
    CO-size     = 1 },
{   *( X.3 parameter 17 -- Line Delete )*
    CO-element-id   = 17,
    CO-category = "character",
    CO-repertoire-assignment *( any from C0 )*
                = "void", "void", <ESC> 2/1 4/0,
    CO-size     = 1 },
{   *( X.3 parameter 18 -- Line Display )*
    CO-element-id   = 18,
    CO-category = "character",
    CO-repertoire-assignment *( any from C0 )*
                = "void", "void", <ESC> 2/1 4/0,
    CO-size     = 1 },
{   *( X.3 parameter 19 -- Editing Service Signals )*
    CO-element-id   = 19,
    CO-category = "transparent",
    CO-size     = 8 },
{   *( X.3 parameter 20 -- Echo Mask )*
    CO-element-id   = 20,
    CO-category = "boolean",
    CO-size     = 8 },
{   *( X.3 parameter 21 -- Parity Treatment )*
    CO-element-id   = 21,
    CO-category = "boolean",
    CO-size     = 2 },
{   *( X.3 parameter 22 -- Page Wait )*
    CO-element-id   = 22,
    CO-category = "integer",
    CO-size     = 256 }
},

{ *( READ -
Each boolean of the READ CO represents an element-id of
the PAD CO with the same identifying value.  The READ
CO is used to request the current values of PAD CO,
which may have been changed by some local agent.  See
the description of the PAD CO for how the update to
this CO modifies the access to the PAD CO. )*
CO-name     = READ,
CO-structure    = 1,
CO-access       = opposite of profile-argument-r1,
CO-priority = "normal",
CO-trigger  = "not-selected",
CO-category = "boolean",
CO-size     = 22
},
```

```
( *( Break Out-of-Band -
receipt of this control object represents "indication
of break"; use is applicable when boolean 1 of element-
id 7 in PAD CO has the value "true". )*
CO-name      = BO,
CO-structure     = 1,
CO-access        = profile-argument-r1,
CO-priority = "urgent",
CO-trigger  = "not-selected",
CO-category = "symbolic",
CO-size      = 1
),

( *( Break In-Band -
receipt of this control object represents "indication
of break"; use is applicable when boolean 3 of element-
id 7 in PAD CO has the value "true". )*
CO-name      = BI,
CO-structure     = 1,
CO-access        = profile-argument-r1,
CO-priority = "normal",
CO-trigger  = "selected",
CO-category = "symbolic",
CO-size      = 1
),
```

```
{ *( CUD -
This CO is used to optionally convey Call User Data
which is normally carried in the CCITT PAD call.  The
CO is not updateable, but may be given initial content
value during association establishment.  The CO is
parametric, with two elements, one representing the
protocol identifier field, and the other representing
the call data field containing user data. )*
CO-name      = CUD,
CO-structure     = 2,
CO-access        = "no-access",
{    *( Protocol Identifier )*
     CO-category = "character",
     CO-repertoire-assignment *( VTS Transparent Set )*
                 = <ESC> 2/5 2/15 4/2,
     CO-size     = 4 },
{    *( User Data )*
     CO-category = "character",
     CO-repertoire-assignment *(VTS Transparent Set )*
                 = <ESC> 2/5 2/15 4/2,
     CO-size     = 12 }
},

{ *( DTE -
This CO is used to optionally indicate the calling and
called DTE addresses which are normally available in a
true CCITT PAD environment.  They may not be updated,
but may be given initial content values during the
association establishment. )*
CO-name      = DTE,
CO-structure     = 2,
CO-access        = "no-access",
{    *( Calling DTE address )*
     CO-element-id   = 1,
     CO-category = "character",
     CO-repertoire-assignment *(VTS Transparent Set )*
                 = <ESC> 2/5 2/15 4/2,
     CO-size     = 15 },
{    *( Called DTE address )*
     CO-element-id   = 2,
     CO-category = "character",
     CO-repertoire-assignment *(VTS Transparent Set )*
                 = <ESC> 2/5 2/15 4/2,
     CO-size     = 15 }
},
```

```
        { *( FAC -
        This CO is used to optionally indicate the CCITT
        facilities which are normally negotiable during the
        establishment of a PAD virtual circuit.  The
        negotiation takes place in the VT association
        establishment, where the initiator may propose the
        initial content value, and the acceptor may return
        other values. )*
        CO-name      = FAC,
        CO-structure     = 1,
        CO-access        = "no-access",
        CO-category = "character",
        CO-repertoire-assignment *(VTS Transparent Set )*
                     = <ESC> 2/5 2/15 4/2,
        CO-size      =  127
        },
    },

    Device-objects *(double occurence)* =
    {
        {
        device-name = DEVICE-1,
        device-default-CO-access = profile-argument-r1,
        device-default-CO-priority = "normal",
        device-default-CO-trigger = "not-selected",
        device-default-CO-initial-value = 1."true",
        device-minimum-X-array-length = 1, *(no constraint)*
        device-control-object = { BI, BO, PAD },
        device-display-object = D1
        *(termination parameters are controlled explicitly
        through the values assigned to the COs P3 and P4 )*
        },
        {
        device-name = DEVICE-2,
        device-default-CO-access =
                          opposite of profile-argument-r1,
        device-default-CO-priority = "normal",
        device-default-CO-trigger = "not-selected",
        device-default-CO-initial-value = 1."true",
        device-minimum-X-array-length = 1, *(no constraint)*
        device-control-object = { READ, PAD },
        device-display-object = D2
        }
    },
    Type of delivery control = "simple-delivery-control".
```

14.8.5.4 Profile Arguments

r1 - is mandatory, and is used to establish the access rules for the display objects and several of the control objects. This argument takes one of the values "WACI" or "WACA". It is identified by the identifier for DO-access for display object D1.

r2 - is optional, and is used to set the intial content values to the elements of the DTE CO.

r3 - is optional, and is used to set the initial content value of the FAC CO.

r4 - is optional, and is used to set the initial content value of the CUD CO.

14.8.5.5 Profile Notes

14.8.5.5.1 Definitive Notes

1. The value assigned to element 1 of PAD CO selects the character used to return control to the terminal-system. The valid values and associated meanings are:

value	meaning
0	not-permitted
1	1/0 character (DLE)
32-126	graphic character

2. The value assigned to element 2 of PAD CO determines whether or not characters are echoed at the terminal-system. When the value of this boolean is "true", then the characters are echoed at the terminal-system.

3. The values assigned to element 3 of PAD CO control the forwarding of characters from the terminal-system to the application-system based on the character value. The defined booleans and associated meanings are:

boolean	meaning
1	alphanumeric (A-Z, a-z, 0-9)
2	character 0/13 (CR)
3	characters 1/11 (ESC), 0/7 (BEL), 0/5 (ENQ), 0/6 (ACK)
4	characters 7/15 (DEL), 1/8 (CAN), 1/2 (DC2)
5	characters 0/3 (ETX), 0/4 (EOT)
6	characters 0/9 (HT), 0/10 (LF), 0/11 (VT), 0/12 (FF)
7	all others in column 0 and 1 not already included above

4. The value assigned to element 4 of PAD CO controls the forwarding of characters from the terminal-system to the application-system based on the duration of idle time elapsed between consecutive characters received by the terminal-system from the device. The valid values include any non-negative integer 0-255; a value between 1 and 255 indicates the time-out in twentieths of a second; a value of 0 means that a time-out is not a forwarding condition.
5. The value assigned to element 5 of PAD CO determines whether the XON/XOFF flow-control characters (1/1 and 1/3) are available for use by

the terminal-system. When the value of this element is "true", then the flow-control characters are available, and the terminal-system may use them to indicate to the device its readiness to accept characters from it.

6. The value assigned to element 6 of PAD CO determines whether the terminal-system issues messages, called PAD service signals, to the device during the association. The specific service signals are not a part of this profile definition, only the control of their issue.

7. The values assigned to element 7 of PAD CO determine the behavior at the terminal-system when a Break is received from the device. The defined booleans and associated meanings are:

boolean	meaning
1	update BO CO
2	release the association
3	update BI CO
4	return control to terminal-system
5	discard data from application-system

When all booleans have the value "false", there is no action at the terminal-system when a Break is received

A useful combination of booleans with value "true" is (1,3,5). When a Break is received, the terminal-system updates both the BO CO and the BI CO and discards all display-object updates from the application-system until it receives an update to the PAD CO for element 8. The result is that the data path has been cleared in both directions. Notice that this is non-destructive of control object updates.

8. The value assigned to element 8 of PAD CO determines whether or not the terminal-system discards data from the application-system. This element works with element 7 to acknowledge the receipt of the Break and resume normal processing of display-object updates. The only valid value of this boolean in an update is "false".

9. The value assigned to element 9 of PAD CO indicates the number of padding characters to be generated by the terminal-system to the device following a carriage return character. The valid values are integers in the range 0-7.

10. The value assigned to element 10 of PAD CO indicates the number of graphic characters sent to the device after which the terminal-system will insert a carriage return. The valid values are integers in the range 0-255, where a value of 0 means that this function is not performed.

11. The value assigned to element 11 of PAD CO indicates the bit-transmission speed of the device. This element may only appear in an update sent to the application-system in response to and update of the READ CO when boolean 11 has the value "true".

12. The value assigned to element 12 of PAD CO determines whether the XON/XOFF flow-control characters (1/1 and 1/3) are available for use by the device. When the value of this element is "true", then the flow-control characters are available, and the device may use them to indicate to the terminal-system its readiness to accept characters from it.

13. The values assigned to element 13 of PAD CO determine under which situations a linefeed is inserted following a carriage return character. The valid values and associated meanings are:

boolean	meaning
1	insert linefeed after carriage return sent to device
2	insert linefeed after carriage return received from device
3	insert linefeed after carriage return echoed to the device

14. The values assigned to element 14 of PAD CO determine the number of padding characters generated by the terminal-system to the device following a linefeed character. The valid values are any number in the range 0-7.

15. The value assigned to element 15 of PAD CO determines whether or not the terminal-system performs data-editing. When this CO has value "true", the values of the elements 3 and 4 of the PAD CO are ignored.

16. The value assigned to element 16 of PAD CO determines which character is used in editing the line to signify the function "delete character". The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true".

17. The value assigned to element 17 of PAD CO determines which character is used in editing to signify the function "delete line". The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true".

18. The value assigned to element 18 of PAD CO determines which character is used in editing to signify the function "display line". The valid values are the IA5 characters, decimal value 0-127. Only applicable if the value of element 15 of PAD CO is "true".

19. The value assigned to element 19 of PAD CO determines whether the terminal-system provides for editing of PAD service signals. The valid values and meanings are as follows:

value	meaning
0	no editing
1	editing as for a paper device
2	editing as for a glass device
8	editing using one editing character
32-126	editing using one editing character

20. The values assigned to element 19 of PAD CO determines which characters are NOT to be echoed to the device by the terminal-system. If no bits are set, then all characters are to be echoed, assuming that element 2 has the value "true". The defined booleans and associated meanings are:

boolean	meaning
1	Do not echo 0/13 (CR)
2	Do not echo 0/10 (LF)
3	Do not echo 0/11 (VT), 0/9 (HT) 0/12 (FF)
4	Do not echo 0/7 (BEL) or 0/8 (BS)
5	Do not echo 1/11 (ESC) or 0/5 (ENQ)
6	Do not echo 0/6 (ACK), 1/5 (NAK), 0/2 (STX), 0/1 (SOH), 0/4 (EOT), 1/7 (ETB) or 0/3 (ETX)
7	Do not echo the editing characters defined by elements 16, 17 and 18 of the PAD CO
8	Do not echo 7/15 (DEL) or any of the other characters belonging to CO or C1 which are not already mentioned above

21. The value assigned to element 21 of PAD CO determines the treatment of parity on the characters received from and sent to the device from the terminal-system. The defined booleans and associated meanings are:

boolean	meaning
1	parity is checked on characters received from the device
2	parity is generated on characters sent to the device

22. The value assigned to element 22 of PAD CO determines the number of linefeeds that the terminal-system may send to the device before it must wait for input from the device request it to continue displaying characters. The range of valid values is 0-255, where a value of 0 indicates that the terminal-system need never wait.

23. The text operation is the only operation allowed on the display objects.

24. The content values of the two elements of the DTE CO convey the calling and called DTE addresses as a sequence of up to 15 decimal digits, where each digit is represented in a binary form and is encoded in an octet.

25. The content values of the FAC CO are encoded using the same encoding as is specified in Recommendation X.28.

26. The content values of the CUD CO are encoded using the same encoding as is specified in Recommendation X.28.

14.8.5.5.2 Informative Notes

1. Users of this profile should refer to CCITT Recommendations X.3, X.28 and X.29 for the original model for this profile.

2. The following values for the elements of the PAD CO are taken from the CCITT Simple standard profile and may prove useful:

element-id	Value
1	1 - possible to return control to the terminal-system using 0/1 (DLE)
2	1."true" - echo performed at the terminal-system
3	1."false", 2."true", 3."true", - 4."true", 5."true", 6."true", 7."true" - forward on receipt of any character in C0 and C1
4	0 - no time-out used for forwarding condition
5	1."true" - terminal-system use XON/XOFF to flow-control the device
6	1."true" - service signals are sent
7	2."true", all others "false" - release the association when a Break is received from the device
8	1."false" - deliver data to device
9	0 - do not pad after CR

10	0 - do not fold the line
11	- read-only
12	1."true" - device use XON/XOFF to flow-control the terminal-system
13	0 - do not insert LF after CR
14	0 - do not pad after LF
15	1."false" - do not edit data
16	7/15 (DEL) - character delete
17	1/8 (CAN) - line delete
18	1/2 (DC2) - line display
19	1 - edit as for paper
20	0 - echo all characters
21	0 - no parity checking or generation
22	0 - no page wait

3. The following values for the elements of the PAD CO are taken from the CCITT Transparent standard profile and may prove useful.

element-id	Value
1	0 - control may not be returned to the terminal-system
2	1."false" - the terminal-system does not perform character echo
3	all booleans "false" - no forwarding on character value
4	20 - forward on time-out of 1 second

5	1."false" - terminal-system may not flow-control device
6	1."false" - service signals are never sent
7	2."true", all others "false" - release the association
8	1."false" - deliver data to device
9	0 - no pad after CR
10	0 - no line folding
11	- read-only
12	1."false" - device may not flow-control terminal-system
13	0 - no LF insert after CR
14	0 - no pad after LF
15	1."false" - no editing data
16	7/15 (DEL) - character delete
17	1/8 (CAN) - line delete
18	1/2 (DC2) - line display
19	1 - edit as for paper
20	0 - echo all characters
21	0 - no parity checking or generation
22	0 - no page wait

14.8.5.6 Specific Conformance Requirements

None.

14.9 APPENDIX A

See Stable Agreements.

14.10 APPENDIX B - CLARIFICATIONS

14.10.1 Defaults

When a profile argument is not present in either the offer or value list, the default for the corresponding VTE parameter is specified by ISO 9040 or the argument description in the profile.

15. TRANSACTION PROCESSING

Editor's Note: This section is a placeholder for future Transaction Processing (TP) Agreements. The TP Special Interest Group is newly formed and held its first regular meeting in March, 1989. Any new text from this group will be inserted here.

16. OFFICE DOCUMENT ARCHITECTURE

Editor's Note: For current Stable ODA Agreements, consult the aligned section of the Stable Implementation Agreements Document, Version 2, Edition 3, June 1989.

17. FUTURE OFFICE DOCUMENT ARCHITECTURE (ODA)

Editor's Note: This section will contain the new text relating to Office Document Architecture (ODA) Agreements.

18. NETWORK MANAGEMENT

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.

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

Document	Target Dates 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 Attributes	12/88	4/90	4/91
	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.

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.

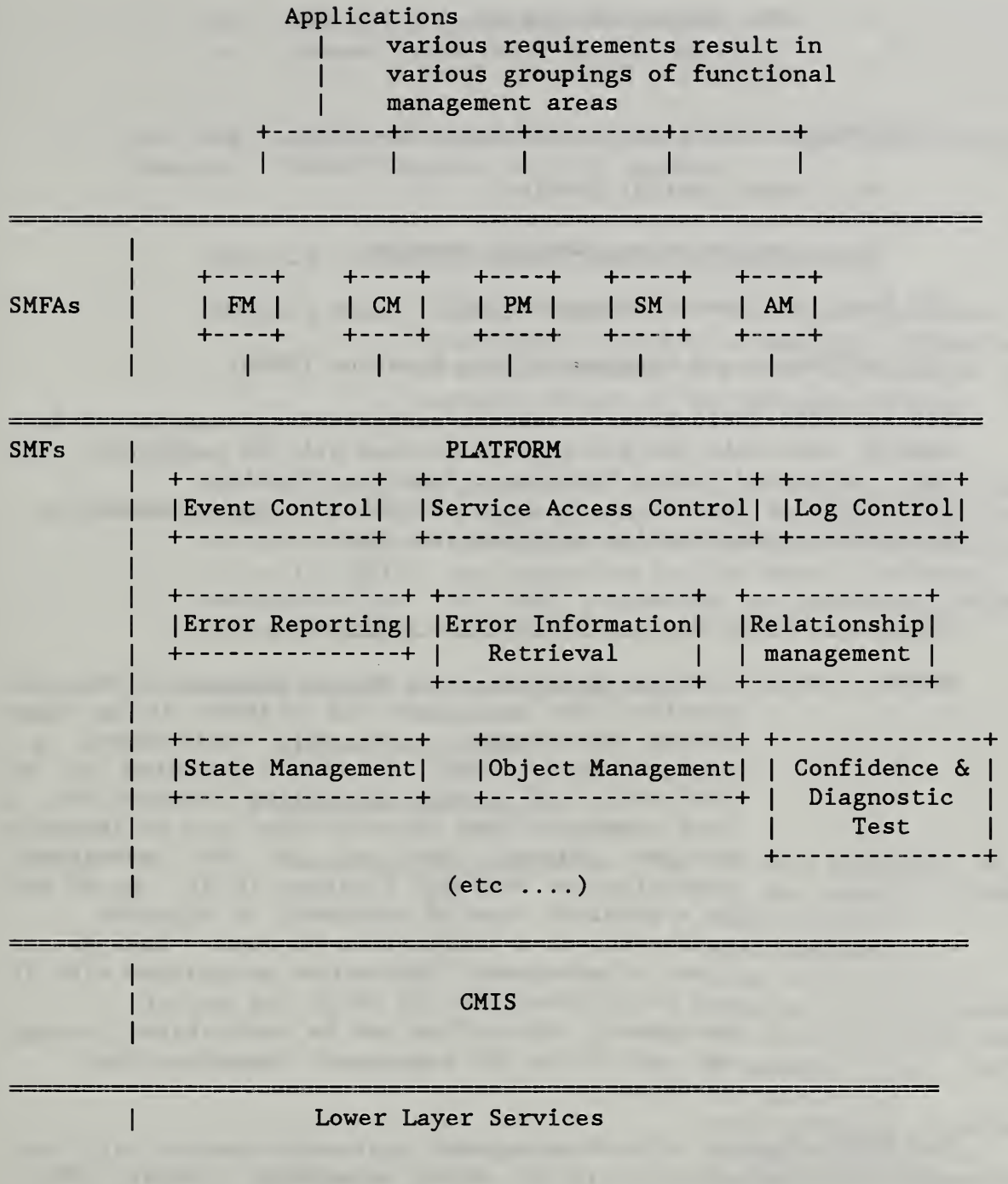
Editor's Note: Tutorial Material. ISO has partitioned network management into five Specific Management Functional Areas (SMFAs) as a convenience for developing requirements particular to configuration management (CM), fault management (FM), performance management (PM), security management (SM), and accounting 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.



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> Refer to Section 18.6
<managed Object Instance> (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>	Refer to Section 18.6 (Management Communications) of this chapter for agreements pertaining to these parameters.
<managedObjectInstance>	

<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.

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> Refer to Section 18.6
<managedObjectInstance> (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

o 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> Refer to Section 18.6
<managedObjectInstance> (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> Refer to Section 18.6
<managedObjectInstance> (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 Agreements

18.5.3.1 Relationship Management Model:

18.5.3.2 Relationship Management using the INDIRECT MODEL:

18.5.3.2.1 Relationship creation Agreements:

18.5.3.2.2 Relationship deletion Agreements:

18.5.3.2.3 Relationship changing Agreements:

18.5.3.2.4 Relationship listing Agreements:

18.5.3.2.5 Related object listing Agreements:

18.5.3.2.6 Relationship creation reporting Service Agreements:

18.5.3.2.7 Relationship deletion reporting Service Agreements:

18.5.3.2.8 Relationship change reporting Service Agreements:

18.5.3.3 Relationship Management using the DIRECT MODEL:

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 Rtrieval Service.

For the NIST 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]):

- Initiate event reporting service
- Terminate event reporting service
- Suspend event reporting service
- 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:

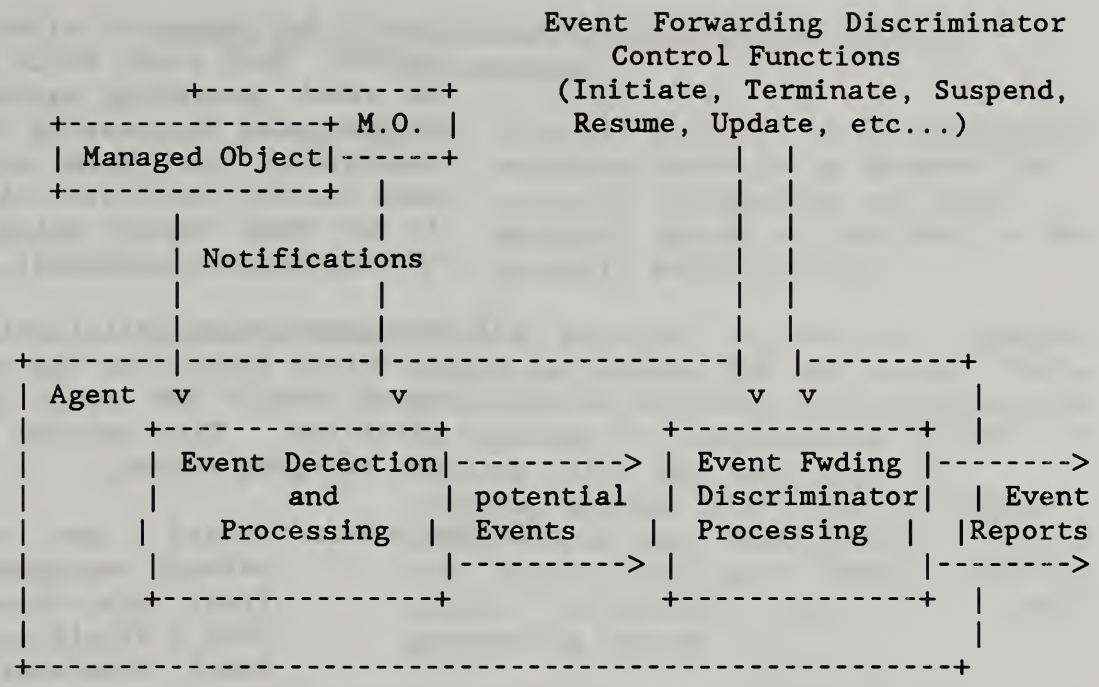
(1) There is (at least) one managed object capable of generating notifications.

(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).



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> Refer to Section 18.6
<managedObjectInstance> (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> Refer to Section 18.6
<managedObjectInstance> (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> Refer to Section 18.6
<managedObjectInstance> (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.

18.6.1 Association Policies

Editor's Note: Define the problem space, and why associations help. Consider that we are trying to simplify the job of building a managed system at the cost of added complexity in the managing system. Consider

also that we are trying to provide some guarantees to managing systems so that they will not interfere with each other - hence we define a controlling association so that there is mutual exclusion for the duration of the particular association.

18.6.1.1 Types of Association

Editor's Note: Define the different types of association, such as monitoring, controlling, etc. These are usually directional and consequently one then defines a monitoring manager and a monitored agent, and so on.

18.6.1.2 Functional Units

Editor's Note: Define the different Functional Units and how they may be combined to identify each endpoint of an association of one of the types previously defined.

18.6.1.3 Functional Unit Negotiation

Editor's Note: Indicate how the association requestor and association responder negotiate to get to a common agreement as to the nature of the particular association. For example, while the requestor may wish to have a controlling association, the responder may not be able to permit it due to an existing controlling association which includes some of the same managed objects. The responder may choose to permit either a controlling association with a reduced scope of MOs, or it may permit a monitoring association with the same MOs. The requestor needs to decide if the negotiated terms are acceptable; if not, the requestor will need to tear down the association.

18.6.1.4 Span of an Association

Editor's Note: Need to indicate the span of an association, notably which managed objects are involved, and over what time period an association is normally expected to exist. In the case of the former, we might choose to involve all MOs under the control of the managed system, or only a subset (perhaps defined by pointing to a place in the containment tree and indicating the scope of MOs, much like Scoping with CMIS/CMIP). For the latter, we might indicate that associations are maintained for as long as needed, but no longer; that might mean for the duration of a "user session" at a terminal, or might mean forever for an event stream. Also need to note that security, in terms of access control, applies to the association.

18.6.1.5 Other Aspects of Associations

Editor's Note: Need to define what happens when an operation is attempted which is not one of those permitted by the association type as agreed at association negotiation time. Need to define what happens when an operation is attempted on a managed object that is not within the span of the association as agreed at asssociation negotiation time. Need to define what happens if Multiple Object Selection is used and the scoped and filtered objects include some objects outside the span of the association - should there be an implicit filter that excludes objects not included in the span of the association? Define other error processing as appropriate. Define any other miscellaneous topics that relate to associations here.

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 theoperations 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 four 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.

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

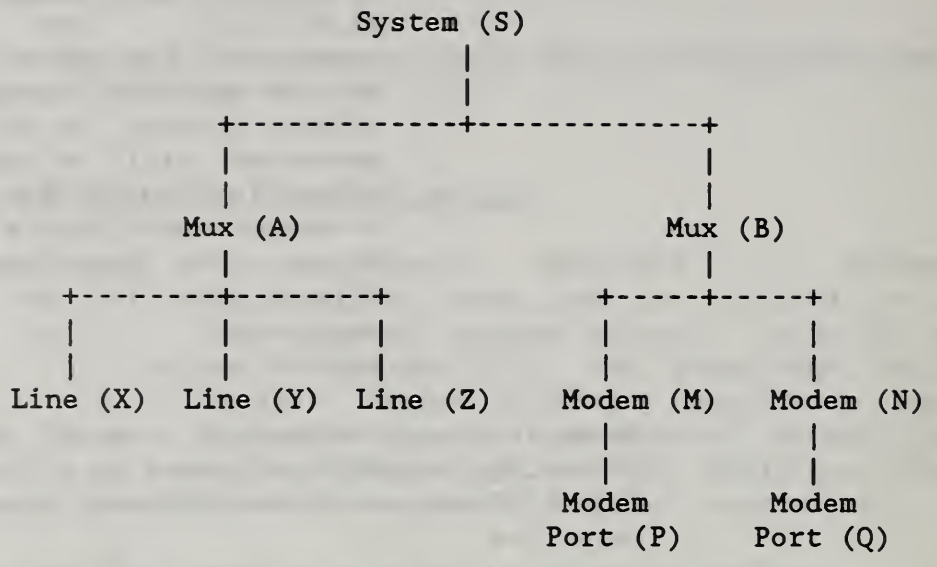
Editor's Note: Tutorial material: Scoping is used to define the scope of managed objects to which a particular management operation will apply (subject also to any

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
- 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):



If the Base Managed Object Class is System and the Base Managed Object is (S):

- If Base Object Only is chosen, then System (S) is the selected object
- If 1st Level Subordinate is chosen, then Mux (A) and Mux (B) are the selected objects
- If 2nd Level Subordinate is chosen, then Line (X) Line (Y), Line (Z), Modem (M) and Modem (N) are the selected objects
- If 3rd Level Subordinate is chosen, then Modem Port (P) and Modem Port (Q) are the selected objects
- If 4th Level Subordinate is chosen, there are no objects that satisfy the criteria.
- 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 'NoSuchObjectInstance' is returned.

Editor's Note: The error 'InvalidScope' will be used instead of 'NoSuchObjectInstance' when and if defined by the standards.

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 operationon all selected objects successfully or else must not perform the operation on any of the objects.

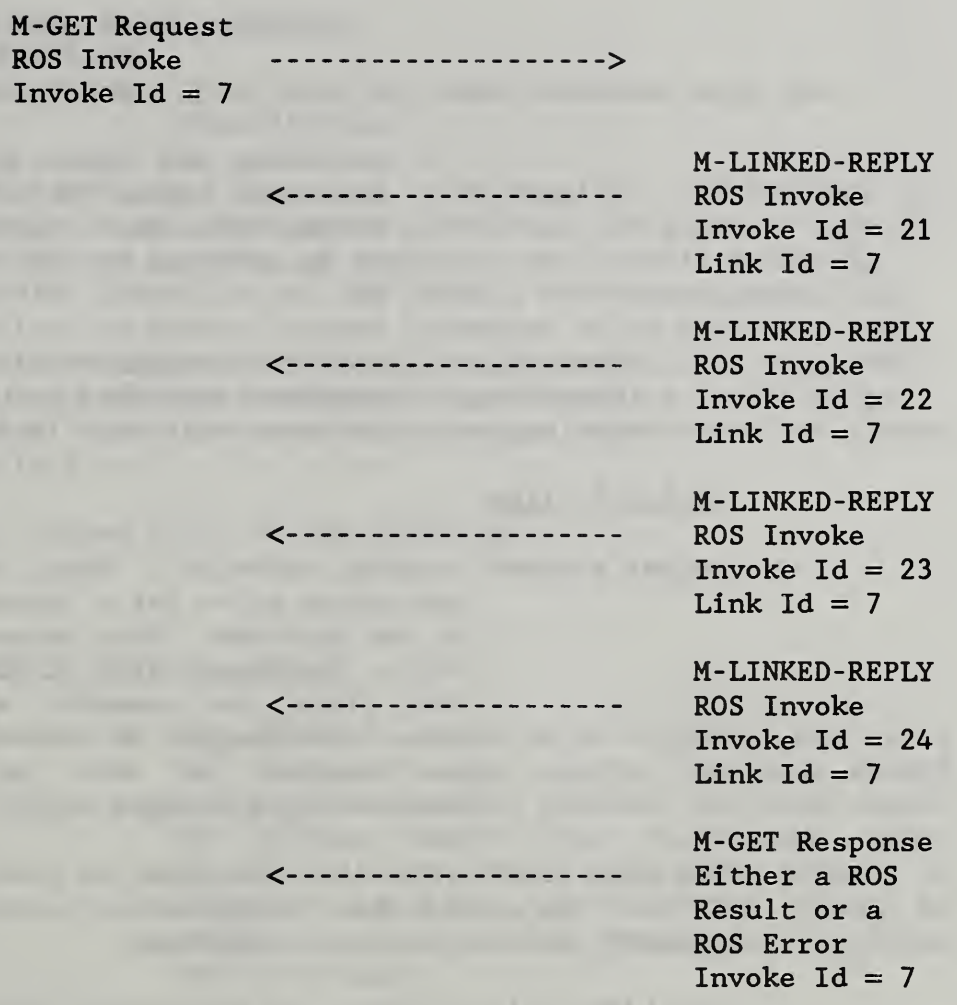
In order to support interoperability between managing systems and managed systems, these Phase 1 network management IAs define that the default synchronization (i.e., BestEffort) must be supported by all conforming systems.

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]):



Editor's Note: What gets returned in the last response? CMIS and CMIP differ on this. If the response contains no attribute list ([CMIS] Section 8.3.1.2.8 for example), then what is in Managed Object Class and Managed Object Instance, etc?
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.

18.6.2.3 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 networkmanagement IAs restrict the Access Control parameter to be permitted only in an association initialization. Use of thisfield in other PDUs for individual management operations is outside the scope of these IAs and conformant implementations may ignore thisfield.

(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 definedin 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 outsidethe scope of these IAs and conformant implementations mayignore 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 granularity of time stamps is defined to be at least as fine as 1ms. Accordingly, the managed system must be able to resolve time to a precision of 1ms.

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-Initialize

The following agreements and clarifications, pertinent to Section 8.1.1 of the base standard [CMIS] and Section 6.1 of the base standard [CMIP] and regarding the M-INITIALIZE service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

M-INITIALIZE Request Parameters:

<Functional Units> (See Section 18.6.1)

<User Information> (See Section 18.6.1)

Editor's Note: Need to define what, if anything, is allowed in this parameter.

<Access Control>

Editor's Note: Need to define the permissible contents of this field.

M-INITIALIZE Response Parameters:

<Functional Units> (See Section 18.6.1.3)

<User Information> (See Section 18.6.1)

Editor's Note: Need to define what, if anything, is allowed in this parameter.

18.6.3.2.2 M-Terminate

The following agreements and clarifications, pertinent to Section 8.1.2 of the base standard [CMIS] and Section 6.9 of the base standard [CMIP] and regarding the M-TERMINATE service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

M-TERMINATE Request Parameters:

<User Information> (See Section 18.6.1)

Editor's Note: Need to define what, if anything, is allowed in this parameter.

M-TERMINATE Response Parameters:

<User Information> See Section 18.6.1)

Editor's Note: Need to define what, if anything, is allowed in this parameter.

18.6.3.2.3 M-Abort

The following agreements and clarifications, pertinent to Section 8.1.3 of the base standard [CMIS] and Section 6.10 of the base standard [CMIP] and regarding the M-ABORT service and protocol, are included within these Phase 1 network management IAs. All parameters are mandatory, except where noted below.

M-ABORT Request Parameters:

<M-ABORT source> (See Section 18.6.1)

<User Information> (See Section 18.6.1)

Editor's Note: Need to define what, if anything, is allowed in this parameter.

18.6.3.2.4 M-EventReport

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>

18.6.3.2.5 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>

Editor's Note: The response parameters may need additional changes if the standards alter the way in which the final response to a multiple reply case is handled.

18.6.3.2.6 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>

18.6.3.2.6.1 Add, Remove and Set to Default

PDAD2 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 (PDAD2) Parameters:

<Modification List>

M-SET Response (PDAD2) Parameters:

<Attribute List>

18.6.3.2.7 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>	

18.6.3.2.8 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.

Editor's Note: New Issue: When a new instance of a managed object is created, there are no agreements w.r.t. association policy that indicate the association over which the object may be managed.

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>

18.6.3.2.9 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 (though this error has yet to be added to CMIS/CMIP).

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 ?

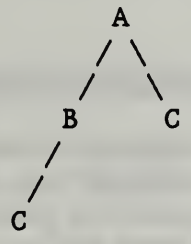
c) Is the distinguishing attribute of a subclass the same or different from distinguishing attribute of its superclass? If the superclass and its subclass have the same distinguishing attribute, there could be ambiguities in situations where instances of both the

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 :



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 relationship instances 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

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

20.3 STATUS

20.4 ERRATA

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 2, Edition 3, June 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 89-4140

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
AUGUST 1989

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 time 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: OSINET: 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
373

15. PRICE
$32.95

ELECTRONIC FORM

☆ U.S. GOVERNMENT PRINTING OFFICE: 1989— 242-200/05054