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

Output from the June 1994 NIST Workshop for Implementors of OSI

SIG Chair: **Chris Bonatti, Booz • Allen & Hamilton**
SIG Editor: **Chris Bonatti, Booz • Allen & Hamilton**

Foreword

The text in this chapter specifies the North American requirements for use of the MHS ISPs. It also specifies any additional requirements and Recommended Practices that are beyond the scope of the ISPs.

Table of Contents

Part 8 Message Handling Systems 1

0 Introduction 1

1 Scope 2

2 References 2
2.1 CCITT 2
2.2 ISO 3

3 Status 4

4 Taxonomy and Functional Groups 4
4.1 AMH1 4
4.2 AMH2 6
4.3 AMH3 8

5 Conformance 9

6 Common Messaging 12
6.1 Introduction 12
6.2 Elements of Service 12
6.3 MTS Transfer Protocol (P1) 12
6.4 MTS Access Protocol (P3) 12
6.5 MS Transfer Protocol (P7) 13
6.6 Pragmatic Constraints 14
6.6.1 MTS – APDU Size 14
6.6.2 Number of Recipient Names 14
6.7 1988/84 Interworking Considerations 14

7 MHS Management 16

8 IPM Service 16
8.1 Introduction 16

9 EDI Messaging Service 17

Annex A (normative)

Naming, Addressing and Routing 18
A.1 ORAddress Attribute List Equivalence Rules 18
A.2 MHS Use of Directory 18
A.2.1 Introduction 18
A.2.2 Functional Configuration 19
A.2.3 Functionality 19

A.2.4 Naming and Attributes 20
A.2.5 Directory Services 21
A.2.6 OIW Application Specific Attributes and Attribute Sets 21
A.2.7 OIW Application Specific Object Classes 23
A.2.8 Structure Rules 23
A.2.8.1 MHS Distribution List 23
A.2.8.2 MHS User 23
A.2.9 Use of Capabilities Information 23

Annex B (normative)

IPM Body Part Support 24

Annex C (normative)

Object Identifiers 26
C.1 X.400 SIG Object Identifiers 26
C.2 Content Types 26
C.3 Body Part Types 27
C.4 Security Classes 27

Annex D (informative)

Interpretation of Elements of Service 28

Annex E (informative)

Recommended Practices 29
E.1 Printable String 29
E.2 Rendition of IA5Text 30
E.3 EDI Use of MHS 31
E.3.1 P0 Recommended Practice 31
E.3.1.1 P0 to P(edi) Conversion 31
E.3.1.2 P(edi) to P0 Conversion 32
E.3.2 P2 Recommended Practice 33
E.3.2.1 Conversion from IPMS to EDIMS (P2 to P(edi)) 33
E.3.2.2 Conversion from EDIMS to IPMS (P(edi) to P2) 33
E.4 ODA Transfer 34
E.5 Use of Externally Defined Body Part 35
E.5.1 General 35
E.5.2 Use of Equivalents of Basic Body Part Types 36
E.5.3 Use of General Text Body Part Type 36
E.5.4 Use of File Transfer Body Part Type 36
E.5.4.1 Encoding of General Identifier 36
E.5.4.2 Encoding of Contents Type 36
E.5.4.3 Encoding of Application Specific Information 36
E.5.4.4 EITs for the File Transfer Body Part 37
E.5.5 Use of Other Extended Body Part Types 37
E.5.6 Obtaining Object Identifiers 38

E.6 Privacy Enhanced Mail Body Part 38
E.7 Selection of OR Name Attributes 39
E.7.1 Teletex Name Attributes 39
E.8 Use of the Teletex Body Part 39
E.9 Provision of Security Class S0A Using Asymmetric Algorithms 40
E.9.1 Protocol Elements 40
E.9.2 Algorithm Selection 41
E.9.3 Certificate Management 42
E.9.4 Other Issues 42

Annex F (informative)

Bibliography 43
F.1 ANSI 43
F.2 Internet 43
F.3 Other References 43

Annex G (informative)

Defense Message Handling Profiles 44
G.1 Introduction 44

Annex H (informative)

Management Domains 45
H.1 Management Domain Names 45
H.2 Use of ADMD Names 47
H.3 Uniqueness of MTS Identifiers Within a Management Domain 48

List of Figures

Figure 1 - Combinations of AMH1n Profiles 5
Figure 2 - Combinations of AMH2n Profiles 7
Figure 4 - 1988 MHS Physical Configurations 9
Figure 5 - 1988 to 1984 Mapping 15
Figure 6 - 1984 to 1988 Mapping 16
Figure A1 - Example of Unregistered Object Class Definition 20
Figure B1 - Privately-Defined Body Parts 25
Figure C1 - Definition of the *mhsig* Object Identifier 26
Figure C2 - Defintion of the X.400 SIG Object Identifier Categories. 26
Figure C3 - Definition of the External Body Part Object Identifiers 27
Figure E1 - ASCII to PrintableString Algorithm 30
Figure E2 - PrintableString to ASCII Algorithm 30
Figure E3 - Externally Defined Body Part Definition 35
Figure E4 - Definition of the Privacy Enhanced Mail Body Part Type 39
Figure H1 - Management Domain Name Construction 45
Figure H2 - Name Construction by Subauthorities 46
Figure H3 - Prefix 47

List of Tables

Table 1 - MHS Configurations 10
Table 2 - Deltas to Clause A.1.2 of ISP 10611-3 12
Table 3 - Deltas to Table A.1.2.4 of ISP 10611-4 13
Table 4 - Deltas to Table A.1.2.4 of ISP 10611-5 13
Table 5 - Deltas to Table A.1.3.1 of ISP 10611-5 13
Table 6 - Deltas to Table A.1.11 of ISP 10611-5 13
Table A1 - Directory Service Support Requirements 21
Table E1 - Printable String to ASCII Mapping 29
Table E2 - Interpretation of Format Effector Combinations 30

Part 8 Message Handling Systems

0 Introduction

This is an Implementation Agreement developed by the Implementor's Workshop sponsored by the National Institute of Standards and Technology to promote the useful exchange of data between devices manufactured by different vendors. This Agreement is based on, and employs protocols developed in accord with, the OSI Reference Model. It provides detailed guidance for the implementor and eliminates ambiguities in interpretations.

This is an Implementation Agreement for Message Handling Systems (MHS) based on the CCITT X.400 (1988) series of Recommendations, the similar (but not identical) ISO MOTIS standard, and Recommendations F.435 and X.435 (1991) (see References). These Recommendations and Standards are referred to as the *base standards*. The term "MHS" is used to refer to both sources where a distinction is unnecessary. Similarly, "1984" and "1988" are often used to distinguish between the CCITT X.400 (1984) series of Recommendations and the later sources.

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

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

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

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

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

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

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

However, it is considered that the OIW Stable Implementation Agreements for CCITT 1984 X.400-based Message Handling Systems (part 7) should not be withdrawn at this stage. It is anticipated that X.400 (1984) implementations will continue to provide a viable alternative for applications that do **not** require the additional 1988 functionality for some time.

1 Scope

This Agreement specifies the requirements for MHS implementations based on the 1988 MHS standards.

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

a) Management Domain (MD) to MD;

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

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

d) MS to Remote UA.

MHS protocols other than the Message Transfer Protocol (P1), the Message Transfer System Access Protocol (P3), the Interpersonal Messaging Protocol P22 (i.e., P2 encoded as integer 22), the Message Store Access Protocol (P7), and the EDI Messaging Protocol (P35) are beyond the scope of this Agreement. Issues arising from the use of other protocols are outside the scope of this document.

2 References

2.1 CCITT

Application Layer - MHS

CCITT Recommendation X.400 (1988), *Message Handling, System and Service Overview*.

CCITT Recommendation X.402 (1988), *Message Handling Systems, Overall Architecture*.

CCITT Recommendation X.407 (1988), *Message Handling Systems, Abstract Service Definition Conventions*.

CCITT Recommendation X.411 (1988), *Message Handling Systems, Message Transfer System: Abstract Service Definition and Procedures*.

CCITT Recommendation X.413 (1988), *Message Handling Systems, Message Store: Abstract Service Definition*.

CCITT Recommendation X.419 (1988), *Message Handling Systems, Protocol Specifications*.

CCITT Recommendation X.420 (1988), *Message Handling Systems, Interpersonal Messaging System*.

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

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

Specifications.

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

CCITT MHS Implementors Guide, Version 8.

2.2 ISO

Application Layer - MHS

ISO 10021-1 *Information Processing Systems - Text Communication - MOTIS - System and Service Overview*.

ISO 10021-2 *Information Processing Systems - Text Communication - MOTIS - Overall Architecture*.

ISO 10021-3 *Information Processing Systems - Text Communication - MOTIS - Abstract Service Definition Conventions*.

ISO 10021-4 *Information Processing Systems - Text Communication - MOTIS - Message Transfer System: Abstract Service Definition and Procedures*.

ISO 10021-5 *Information Processing Systems - Text Communication - MOTIS - Message Store: Abstract Service Definition*.

ISO 10021-6 *Information Processing Systems - Text Communication - MOTIS - Protocol Specifications*.

ISO 10021-7 *Information Processing Systems - Text Communication - MOTIS - Interpersonal Messaging System*.

OIW SIA Chapter 29 - Working Draft ISP 10611 *Information Processing Systems - International Standardized Profiles AMH1n - Message Handling Systems - Common Messaging*.

OIW SIA Chapter 30 - Working Draft ISP 12062 *Information Processing Systems - International Standardized Profiles AMH2n - Message Handling Systems - Interpersonal Messaging*.

3 Status

This version of the *Implementation Agreements for Message Handling Systems (MHS)* is under development. It is based on the CCITT X.400 (1988) Recommendations and ISO MOTIS (10021, parts 1-7) standards, as amended by the *MHS Implementors Guide*, version 8, as well as ISPs AMH1n and AMH2n (with deltas defined in this document).

4 Taxonomy and Functional Groups

The 1988 MHS standards cover a wide and diverse range of functional areas, not all of which would be relevant to every implementation. The Implementors Agreements describe the services in terms of profiles and divide some of the functionality into the concept of optional Functional Groups. Although the profiles have been developed in open workshops and were reasonably mature there have been some differences between the OIW profiles and those developed by EWOS/ETSI. It has therefore, in the interest of international harmonization, been the intention all along to replace the OIW agreements with pointers to the International Standardized Profiles for MHS once these became stable.

At this point these agreements include the ISPs by reference and include any differences that are required in the North American market in the form of deltas to the ISPs.

The AMH ISPs were developed under the management of the MHS ISP Special Group (MISG). The MISG was formed in early 1991 as a joint workshop initiative, comprising delegations from the MHS groups of the three regional workshops, OIW, EWOS/ETSI, and AOW. It has provided a forum for developing and agreeing the MHS ISP taxonomy, resolving key issues and carrying out initial review of revised ISP drafts. All MISG decisions have been subject to ratification by the full meetings of the workshop MHS groups, which have also carried out detailed review of the ISP drafts.

The AMH set of profiles, so far consists of three multipart profiles.

AMH1 covers Common Messaging - i.e., those aspects of the MHS base standards which are independent of a particular content type.

AMH2 covers the Interpersonal Messaging content type.

AMH3 covers the EDI Messaging content type.

4.1 AMH1

The AMH1n set of profiles is applicable to end systems operating in an Open Systems Interconnection (OSI) environment which form part of a distributed Message Handling Systems (MHS) environment as specified in ISO/IEC 10021 (MOTIS) and the equivalent CCITT X.400 Recommendations. The AMH1n profiles each specify a particular combination of OSI standards which collectively provide one of the MHS services as realized by an MHS protocol:

- AMH11 - Message Transfer (P1 protocol) - between message transfer agents (MTAs)

- AMH12 - Message Transfer System (MTS) Access (P3 protocol) - between a remote user agent (UA) and an MTA, and between a remote message store (MS) and an MTA.

- AMH13 - Message Store (MS) Access (P7 protocol) - between a remote UA and an MS

Profile AMH11 is further subdivided into:

- AMH111 - requiring support of a "normal mode" OSI protocol infrastructure [as required by ISO/IEC 10021 (MOTIS)]

- AMH112 - requiring support of an "X.410 mode" OSI protocol infrastructure [as required by the CCITT X.400 (1984) Recommendations]

An MTA which conforms to profile AMH11 may conform to AMH111, or to AMH112, or both.

Each AMH1n profile specifies the conformance requirements for all relevant MHS functional objects (i.e., MTA, UA, MS). Two or more AMH1n profiles can be combined to establish the conformance requirements for the various physical configurations that may be achieved within the scope of the MHS base standards as illustrated in the following diagram.

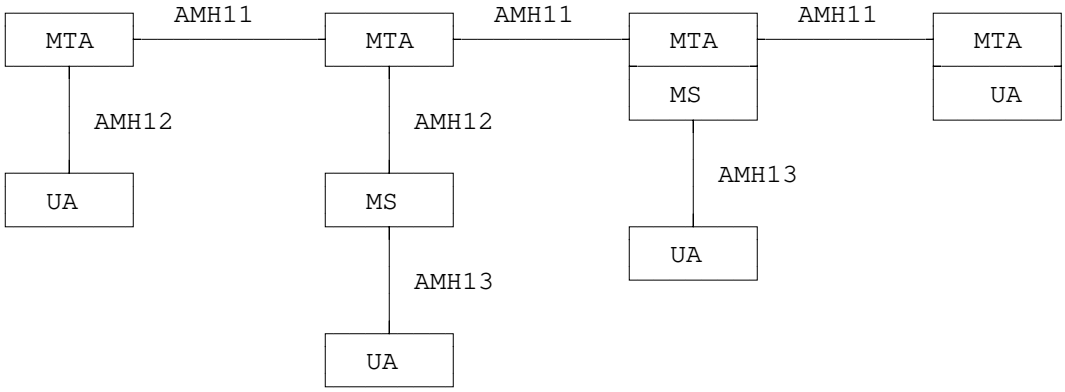

Figure 1 - Combinations of AMH1n Profiles

The AMH1n set of profiles is specified as a multipart ISP consisting of the following parts:

Part 1: MHS service support.

A common text part which provides functional description and specification of MHS service support and associated functionality as covered by the AMH1n set of profiles. It identifies what service support and associated functionality can be supported by each type of MHS component, divided into basic requirements (i.e., required to be supported by all implementations) and zero or more optional functional groups (discrete sets of related functionality which are not required to be supported by all implementations). Such specifications are in many cases applicable to more than one MHS protocol or are otherwise concerned with component functionality which although it can be verified via protocol, is not just related to protocol support. The specification in this part is therefore designed for reference by the following parts (which specify conformance requirements by protocol for each MHS component) and is additional to the protocol-specific requirements specified in those parts. Thus, although this part contains normative requirements, there is no

separate conformance to this part (i.e., it is not identified in the MHS taxonomy) since such requirements are only significant when referenced in the context of a particular protocol profile.

Part 2: Specification of ROSE, RTSE, ACSE, Presentation and Session protocols for use by MHS.

A common text part which provides specification of the underlying protocol infrastructure requirements to support the various MHS application contexts. This is achieved as far as possible by reference to the Common Upper Layer Requirements (CULR): Basic connection oriented requirements ISP 11188-1, plus specification of any further requirements which are either MHS-specific or otherwise not covered by Part 1 of the CULR ISP (ROSE, RTSE).

Part 3: AMH11 - Message Transfer (P1).

This part covers message transfer between MTAs using the P1 Message transfer Protocol. It specifies P1 support in terms of basic requirements and optional functional groups and defines conformance requirements for an MTA which supports transfer with respect to support of P1 and associated functionality (by reference to the common specifications in part 1).

Part 4: AMH12 - MTS Access (P3).

This part covers access to an MTS using the P3 MTS Access Protocol. It specifies P3 support in terms of basic requirements and optional functional groups and defines conformance requirements for an MTA which supports remote access, and for a remote MTS-user (i.e., UA or MS) with respect to support of P3 and associated functionality (by reference to the common specifications in part 1).

Part 5: AMH13 - MS Access (P7).

This part covers access to an MS using the P7 MS Access Protocol It specifies P7 support in terms of basic requirements and optional functional groups and defines the conformance requirements for an MS which supports remote access, and for a remote MS-user (i.e., UA), with respect to support of P7 and associated functionality (by reference to the common specifications in part 1).

4.2 AMH2

The AMH2n set of profiles is applicable to end systems operating in an Open Systems Interconnection (OSI) environment which form part of a distributed Message Handling Systems (MHS) environment and which provide an interpersonal messaging service.

The AMH21 profile specifies the Interpersonal Messaging (IPM) content (P2 "protocol") which is carried end-to-end (i.e., UA-to-UA) by the MHS protocols (i.e., P1, P3, and P7).

The remaining AMH2n profiles cover the other aspects of an IPM MHS environment, specifying additional requirements to those specified in the AMH1n Common Messaging set of profiles as appropriate to support an IPM service:

- AMH22 - IPM Requirements for Message Transfer (P1) - any additional MTA capabilities related

to message transfer which are specific to support of an IPM environment (i.e., additional to the requirements of AMH11)

- AMH23 - IPM Requirements for MTS Access (P3) - any additional MTA and MTS-user capabilities related to MTS access which are specific to support of an IPM environment (i.e., additional to the requirements of AMH12)

- AMH24 - IPM Requirements for MS Access (P7) - any additional MS and MS-user capabilities related to MS access which are specific to support of an IPM environment (i.e., additional to the requirements of AMH13)

Each AMH2n profile specifies the conformance requirements for all relevant MHS functional objects (i.e., MTA, UA, MS). Two or more AMH2n profiles can be combined to establish the conformance requirements for the various physical configurations that may be achieved within the scope of the MHS base standards as illustrated in the following diagram.

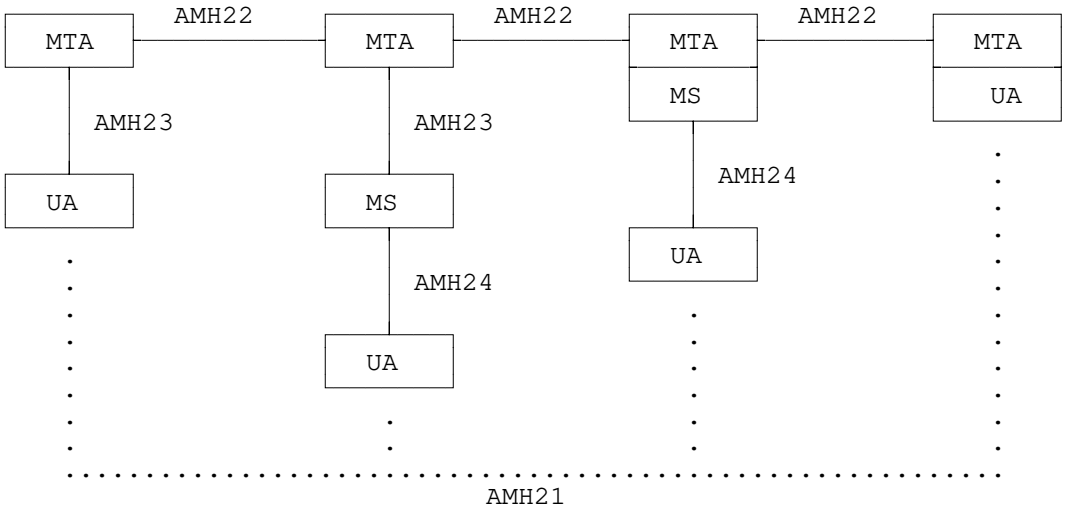

Figure 2 - Combinations of AMH2n Profiles

The AMH2n set of profiles is specified as a multipart ISP consisting of the following parts:

Part 1: IPM MHS service support.

A common text part which provides functional description and specification of IPM-specific MHS service support and associated functionality as covered by the AMH2n set of profiles. It identifies what additional service support and functionality can be supported by each type of MHS component in an IPM environment (i.e., also covering the services supported by an IPMS UA, plus any additional MTA and MS aspects such as IPM body part conversion), divided into basic requirements and zero or more optional functional groups (see AMH1n). Such specifications are in many cases applicable to more than one MHS protocol or are otherwise concerned with component functionality which, although it can be verified via protocol, is not just related to protocol support. The specification in this part is therefore designed for reference by the following parts (which specify conformance requirements by protocol for each MHS component) and is additional to the protocol-specific requirements specified in those parts. Thus, although this part contains

normative requirements, there is no separate conformance to this part (i.e., it is not identified in the MHS taxonomy) since such requirements are only significant when referenced in the context of a particular protocol profile.

Part 2: IPM Content.

This part covers IPMS UA functionality. It specifies support of the IPM content 'protocol' in terms of basic requirements and optional functional groups and defines conformance requirements for an IPMS UA with respect to support of IPM content and associated functionality (by reference to the common IPM specifications in part 1).

Part 3: AMH22 - IPM requirements for Message Transfer (P1).

This part covers message transfer between MTAs using the P1 Message Transfer Protocol to support an IPM environment. It specifies any additional P1 support to that specified in AMH1n and defines conformance requirements for an MTA which supports IPM transfer with respect to support of P1 and associated functionality (requiring conformance to AMH11 and by reference to the common IPM specifications in part 1).

Part 4: AMH23 - IPM requirements for MTS Access (P3).

This part covers access to an MTS using the P3 MTS Access Protocol to support an IPM environment. It specifies any additional P3 support to that specified in AMH1n and defines conformance requirements for an MTA which supports remote access for IPM use, and for a remote MTS-user in an IPM context (i.e., IPMS UA or MS), with respect to support of P3 and associated functionality (requiring conformance to AMH12 and by reference to the common IPM specifications in part 1).

Part 5: AMH24 - IPM requirements for Enhanced MS Access (P7).

This part covers access to an MS using the P7 MS Access Protocol to support an IPM environment. It specifies any additional P7 support to that specified in AMH1n and defines conformance requirements for an MS which supports remote access for IPM use, and for a remote MS-user in an IPM context (i.e., IPMS UA), with respect to support of P7 and associated functionality (requiring conformance to AMH13 and by reference to the common IPM specifications in part 1).

4.3 AMH3

Editor's Note: [See the OIW Working Implementation Agreements, Chapter 8, Clause 4.3.**]**

5 Conformance

MHS implementations may be configured as any single or multiple occurrence or combination of MTA, MS and UA, as illustrated in Figure 4. It is not intended to restrict the types of system that may be configured for conformance to this Agreement (although it is equally recognized that not all configuration types may be commercially viable).

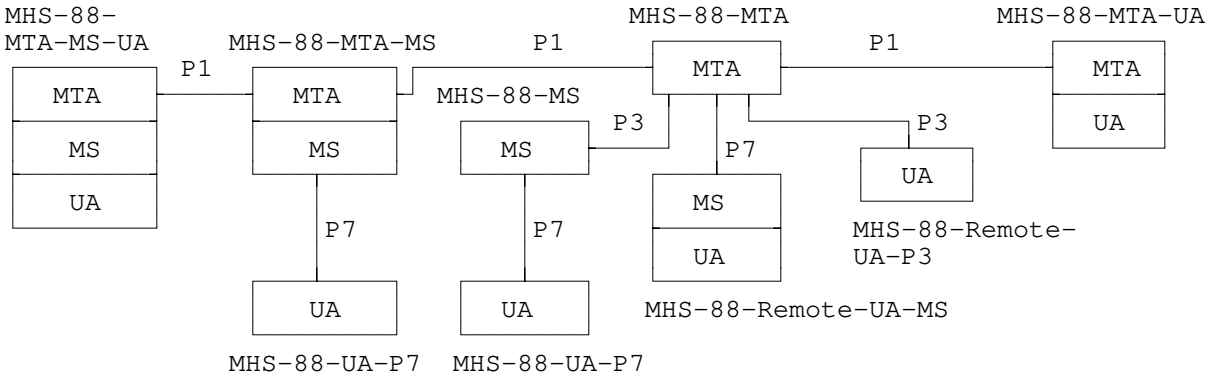

Figure 4 - 1988 MHS Physical Configurations

Figure 4 shows the possible physical configurations for 1988 MHS implementations. The following lists the conformance requirements for each according to the name in that figure and the requirements in this Agreement.

"MHS-88-MTA" specifies a 1988 relay MTA. It must conform to AMH11 as enhanced by the delta described in section 6 of this Agreement. If the MTA also supports a particular content type it may claim conformance to AMH22 for IPMS or AMH32 for EDI, again as enhanced by sections 8 for IPM or 9 for EDI, support for additional content types can be specified in the PICS for AMH11, section A.3.2.

"MHS-88-MTA-UA" specifies a 1988 end system in which the MTA is co-located with a User Agent. If the UA is a CCITT 1988 Interpersonal Messaging (IPM) UA, it must conform to AMH21 and AMH22 as enhanced by section 8 of this Agreement. If the UA is an Electronic Data Interchange (EDI) UA it must conform to AMH31 and AMH32 as enhanced by section 9 of this Agreement. If the UA supports any other content type, the implementation must conform to AMH11. The same UA implementation may support multiple content types by conforming to more than one of these profile combinations.

"MHS-88-MTA-MS-UA" specifies an end system in which a Message Store and User Agent are co-located with the MTA. Conformance to this configuration can only be tested in terms of the MTA and UA interfaces, therefore the conformance requirements are identical to the "MHS-88-MTA-UA."

"MHS-88-MTA-MS" specifies an end system in which a Message Store is co-located with the MTA. At a minimum this configuration must conform to AMH11 and AMH13 as enhanced by section 6 of this Agreement If the MS supports one or more content types these must be specified in filling out the PICS for AMH13 or by conformance to AMH24 for IPMS or AMH34 for EDI, again as enhanced by this Agreement.

"MHS-88-Remote-UA-P3" specifies a remote User Agent that does not require Message Store services. If the
UA is a CCITT 1988 Interpersonal Messaging (IPM) UA, it must conform to AMH21 and AMH23 as enhanced by section 8 of this Agreement. If the UA is an Electronic

Data Interchange (EDI) UA it must conform to AMH31 and AMH33 as enhanced by section 9 of this Agreement. If the UA supports any other content type, the implementation must conform to AMH12. The same UA implementation may support multiple content types by conforming to more than one of these profile combinations.

"MHS-88-Remote-UA-P7" specifies a remote User Agent that does require Message Store services. If the UA is a CCITT 1988 Interpersonal Messaging (IPM) UA, it must conform to AMH21 and AMH24 as enhanced by section 8 of this Agreement. If the UA is an Electronic Data Interchange (EDI) UA it must conform to AMH31 and AMH34 as enhanced by section 9 of this Agreement. If the UA supports any other content type, the implementation must conform to AMH12. The same UA implementation may support multiple content types by conforming to more than one of these profile combinations.

"MHS-88-MS" specifies a remote Message Store that serves a remote User Agent. If the MS is a CCITT 1988 Interpersonal Messaging (IPM) MS, it must conform to AMH24 and AMH22 as enhanced by section 8 of this Agreement. If the MS is an Electronic Data Interchange (EDI) MS, it must conform to AMH34 and AMH33 as enhanced by section 9 of this Agreement. If the MS supports any other content type, the implementation must conform to both AMH12 and AMH13 and specify the content type(s) supported, if any, in section A.1.3 of the PICS for AMH13.

"MHS-88-Remote-UA-MS" specifies a remote User Agent that is co-located with a Message Store. For conformance purposes this is the same as the "MHS-88-Remote UA-P3."

The following table summarizes the conformance requirements for each possible '88 MHS implementation.

Table 1 - MHS Configurations

Entity	Protocol(s)	Conformance
MHS-88-MTA	P1 + possible content types IPMS EDI other	AMH11 + Section 6 AMH22 + Section 8 AMH32 + Section 9 details in PICS in AMH11 (A.3.2)

Table 1 - MHS Configurations (concluded)

Entity	Protocol(s)	Conformance
MHS-88-MTA-UA	P1 + possible content types IPMS EDI other	AMH11 + Section 6 AMH21 + AMH22 + Sec. 6 AMH31 + AMH32 + Sec. 9 details in PICS in AMH11 (A.3.2)
MHS-88-MTA-MS	P1 + P7 + possible content types IPMS EDI other	AMH11 + AMH13 + Sec. 6 AMH22 + AMH24 + Sec. 8 AMH32 + AMH34 + Sec. 9 details in PICS in AMH11 (A.3.2) and AMH13 (A.3)
MHS-88-Remote-UA-P3	P3 + possible content types IPMS EDI other	AMH12 + Sec. 6 AMH21 + AMH24 + Sec. 8 AMH31 + AMH34 + Sec. 9 detail in PICS in AMH13 (A.3)
MHS-88-Remote-UA-P7	P7 + possible content types IPMS EDI other	AMH13 + Sec. 6 AMH21 + AMH24 + Sec. 8 AMH31 + AMH34 + Sec. 9 details in PICS in AMH13 (A.3)
MHS-88-MS	P7 + possible content types IPMS EDI other	AMH12 + AMH13 + Sec. 6 AMH23 + AMH24 + Sec. 8 AMH32 + AMH34 + Sec. 9 details in PICS in AMH13 (A.3) and AMH14 (A.3)
MHS-88-Remote-UA-MS	P3 + possible content types IPMS EDI other	AMH12 + Sec. 6 AMH21 + AMH23 + Sec. 8 AMH31 + AMH33 + Sec. 8 details in PICS in AMH12 (A.3)
MHS-88-MTA-MS-UA	P1 + possible content types IPMS EDI other	AMH11 + Sec. 6 AMH21 + AMH22 + Sec. 8 AMH31 + AMH32 + Sec. 9 details in PICS in AMH11 (A.3.2)

6 Common Messaging

6.1 Introduction

A minimal 1988-based MTA shall conform to AMH111 and AMH112, and will support the 1984 Interworking functional group, in order to achieve interworking with 1984-based MTAs and to facilitate migration to full 1988 operation. In addition, a conforming implementation shall obey the criticality mechanism defined in the base standards. The following protocol elements are made critical for delivery for these Implementation Agreements: message token, content integrity check, and content confidentiality algorithm ID.

Note that when a table entry is blank then the classification shall be that of the appropriate referenced ISP.

6.2 Elements of Service

Implementations conforming to these agreements shall conform to the Element of Service (EoS) requirements of ISP 10611-1, as modified by the following tables.

6.3 MTS Transfer Protocol (P1)

Implementations of MTAs conforming to these agreements shall, at a minimum, implement the AMH111 and AMH112 profiles specified in ISP 10611-3. Collectively, these profiles require support of all three application contexts defined in the 1988 base standards. The OIW requires support of both profiles in order to encourage use of the *mts-transfer* application context, and to provide a solid foundation for 1984 interworking.

Implementations conforming to these agreements shall conform to the requirements of ISP 10611-3, as modified by the following tables.

Table 2 - Deltas to Clause A.1.2 of ISP 10611-3

Ref	Application Context	Profile
1	mts-transfer	m
2	mts-transfer-protocol	m
3	mts-transfer-protocol-1984	m

6.4 MTS Access Protocol (P3)

Implementations conforming to these agreements shall conform to the EoS requirements of ISP 10611-4, as modified by the following tables.

Table 3 - Deltas to Table A.1.2.4 of ISP 10611-4

Ref	Operation	MTS-user		MTA	
		Base	Profile	Base	Profile
1	Register				m
2	ChangeCredentials (MTA to UA)		m		
3	ChangeCredentials (UA to MTA)				m

6.5 MS Transfer Protocol (P7)

Implementations conforming to these agreements shall conform to the EoS requirements of ISP 10611-5, as modified by the following tables.

Table 4 - Deltas to Table A.1.2.4 of ISP 10611-5

Ref	Operation	UA		MS	
		Base	Profile	Base	Profile
2	ChangeCredentials (MTA to UA)		m		

Table 5 - Deltas to Table A.1.3.1 of ISP 10611-5

Ref	Element	UA		MS	
		Base	Profile	Base	Profile
1	ARGUMENT				
1.4	fetch-restrictions				
1.4.1	allowed-content-types				m
1.4.2	allowed-EITs				m
1.4.3	maximum-content-length				m

Table 6 - Deltas to Table A.1.11 of ISP 10611-5

Ref	Attribute	UA		MS	
		Base	Profile	Base	Profile
28	originator-name				m9

o1 - This element is classified as m in the ISP.

m9 - Presently classified as o in ISP. MISG #7 proposed to change this field to m.

6.6 Pragmatic Constraints

6.6.1 MTS - APDU Size

This clause is not intended to constrain the size of PDUs that are transferred across the network, since some body part types and content types (e.g., voice, file transfer, and EDI) may require very large PDUs.

The following agreements govern the size of MTS-APDUs:

a) All MTAEs must support at least one MTS-APDU of at least two megabytes; and,

b) The size of the largest MTS-APDU content supported by a UAE is a local matter.

6.6.2 Number of Recipient Names

There is no specified bound on the number of recipient-names an implementation must support, other than the 32K-1 specified in the standard (Annex B/X.411).

6.7 1988/84 Interworking Considerations

a) Internal Trace Information - If the 1984-based MTA does not support Internal Trace Information per clause 7.3.2 of part 7, the following description is not applicable. When a 1988-based MTA supports interworking with a 1984-based MTA that generates Internal Trace Information as per clause 7.3.3 of part 7, the 1988-based MTA must support reception of the Internal Trace Information by converting the Internal Trace Information from the form in clause 7.3.2 of part 7 to the form specified in 1988 X.411, as per the following description. When the 1988-based MTA sends to a 1984 MTA, the 1988-based MTA must apply the conversion to 1984, as described below. The OIW Stable Implementation Agreements X.400 (1984) definition for MTA's Internal Trace Information is different from the X.400 (1988) MTA definition. Consequently, a X.400 (1988) MTA operating in an MD with other MTAs of 1984 vintage, must map the Internal Trace Information to and/or from the 1984 format.

Figures 5 and 6 depict algorithms for mapping between X.400 (1988) Internal Trace element formats and the OIW IA X.400 (1984) Internal Trace element format.

To avoid potential looping within a MD composed of 1984 and 1988 vintage MTAs, MD administrators are strongly advised to name all MTAs (1984 and 1988 vintages) using only the Printable String characters. In X.400 (1988) the MTA-Name is defined to be named using IA5 String characters where in the IAs for X.400 (1984) MTAs, NBS restricted the MTA-Name to be formed using the Printable String character subset of IA5. If the 1988-based MTA Name uses IA5 characters not in the Printable String subset, that Internal Trace Element should be omitted when converting from 1988 to 1984.

```
For each Internal Trace element in the sequence:
DO
  IF global-domain-identifier does not identify the
   current domain  THEN
     Discard all internal trace elements up to this point,
          including this element;
  ELSE IF converted-encoded-information-types present THEN
     Discard all internal trace elements up to this point,
          including this element;
  ELSE IF MTA-Name is made up of non-PrintableString
   characters  THEN
    Discard this Internal Trace element;
  ELSE
  {    Discard the GlobalDomainIdentifier;
       Within the MTASuppliedInformation:
         Copy the arrival time over;
         Copy the routing action over;
         IF attempted is present
         {    IF it is a domain:
                Discard the 'attempted' attribute;
              IF it is an MTA:
                Copy it to PreviousMTAName;
         }
         IF the additional actions are present:
         {    IF the deferred time is present:
                Copy it over;
              IF other-actions is present:
                 IF 'redirected' or 'dl-operation' (from
                  A/3311) THEN
                  [NOTE: Another instance of Internal Trace
                  Info must be added following the instance
                  being processed!]
                    Discard it;
         }
         Append the Internal Trace Info to the output list;
         IF other-actions requires an additional instance THEN
         {   Copy the arrival time from the previous instance;
             Copy the MTAName from the previous instance;
             Set the 'action' attribute to 'recipient-
               reassigned (2)';
             Append the Internal Trace Info to the
                 output list;
         }
  }
END-DO
```

Figure 5 - 1988 to 1984 Mapping

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

  END-DO
```

Figure 6 - 1984 to 1988 Mapping

NOTE - The 1988 X.419 Recommendation acknowledges that a 1984 system may receive messages containing new distinguished [integer] values that it is not expecting, and that this may result in service irregularities. It is implied that it would be optimal for 1984 systems to accept these unexpected integer values if at all possible. No downgrading should be done for these values when passing affected messages from newer systems to older systems.

7 MHS Management

Editor's Note: [See OIW Working Implementation Agreements, Chapter 8, Clause 7.]

8 IPM Service

8.1 Introduction

This clause specifies IPM conformance requirements. Conformance to AMH2 is required, as well as support of the 1984 Interworking functional group.

9 EDI Messaging Service

Editor's Note: [See OIW Working Implementation Agreements, Chapter 8, Clause 9]

Annex A (normative)

Naming, Addressing and Routing

A.1 ORAddress Attribute List Equivalence Rules

Two ORAddresses are equivalent if each contains the same set of attributes and each attribute compares in type and value.

The following equivalence rules apply when comparing a provided ORAddress with a collection of known ORAddresses. For example, in order to perform delivery of a message to a recipient, the MTA must unambiguously match the ORAddress contained in the message with the known ORAddresses. See X.402 (1988), section 18.4, for the base standard attribute equivalence rules. The following additional rules must also be applied by the delivering (or non-delivering) MTA:

a) An ADMD or PRMD name that is all numeric but encoded as Printable String is considered to be equivalent to the same ADMD or PRMD name, respectively, with the same numeric values encoded as Numeric String.

b) An extension attribute encoded as Teletex String shall be compared with the corresponding standard attribute encoded as Printable String if that extension attribute is not present in both ORAddresses. Matching rules are as specified in clause 18.4 of X.402 (1988) (as modified in the MHS Implementors' Guide) except that only teletex graphic characters from repertoire no. 102 need to be compared for Printable String equivalence (i.e., the presence of graphic characters from other repertoires can be treated as a mismatch).

NOTES

1 An X.500 Directory service may or may not support these matching rules for equivalence.

A.2 MHS Use of Directory

Editor's Note - It has been suggested that much of this material could be moved to an informative annex.

A.2.1 Introduction

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

NOTE - The MTS may also use the directory service elements to obtain information, for example, to be used in the routing of messages. This application of the directory service is not defined by the base

standards and is therefore not addressed by this Agreement.

A.2.2 Functional Configuration

A.2.3 Functionality

Examples of functional usages of directories have been identified for UAs and the MTAs in conjunction with their DUAs. These are:

a) UA Specific Functionality:

1) Verify the existence of a Directory Name.

2) Given a partial name, return a list of possibilities.

3) Search the Directory for entries containing a specified attribute type and value and return the Distinguished Names of the matching entries.

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

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

6) Return the members of a Distribution List.

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

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

b) MTA Specific Functionality:

1) Authentication.

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

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

4) Return the members of a Distribution List.

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

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

In addition to functionality, a number of operational aspects must be considered. These include

user-friendliness, flexibility, availability, expandability and reliability.

A.2.4 Naming and Attributes

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

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

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

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

c) The naming for mhs-user is that of organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality.

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

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

Figure A1 - Example of Unregistered Object Class Definition

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

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

A.2.5 Directory Services

These Implementation Agreements require the Directory services as defined in the following table. Indicated are the Directory services required to support the needs of the MHS UA/MTA and MHS Administrator.

Table A1 - Directory Service Support Requirements

Directory Service	MHS UA/MTA	MHS Admin
Bind and Unbind	M	M
Read	M	M
Compare	M	M
Abandon	M	M
List	M	M
Search	M	M
Add Entry	O	M
Remove Entry	O	M
Modify Entry	M	M
Modify RDN	O	O

A.2.6 OIW Application Specific Attributes and Attribute Sets

The following attribute is proposed as an addition to mhs-user.

```
mhs-or-addresses-with-capabilities ATTRIBUTE
      WITH ATTRIBUTE SYNTAX mhs-or-addresses-with-capabilities-syntax
      MULTI VALUE
      ::= id-at-mhs-or-addresses-with-capabilities
```

This is similar to a proposal in "Working Draft for ISO/IEC 10021-2/PDAM 3, Second Minor Enhancements," which is expected to be ballotted as a PDAM.

Logically, both the present ORAddress and individual capabilities and mhs-or-addresses-with-capabilities would be populated in the Directory for users with multiple O/R addresses. If multiple O/R addresses are returned when an O/R address is requested, the user can then query the new attribute for capabilities of each O/R address. The capabilities of ORAddress would be a union of the capabilities in the 1988 standard of all the O/R addresses.

The syntax proposed in the expected PDAM does not fulfill user requirements or future standards requirements, because it is not extensible. Furthermore, the syntax does not make sense, since it specifies multiple sets of capabilities for one ORAddress, and there is no matching rule allowing one to find an ORAddress having a particular capability. The following syntax and matching rules are suggested to overcome the shortcoming in the expected PDAM.

```
mhs-or-addresses-with-capabilities-syntax ::= SEQUENCE {
      address                 ORAddress,
```

```
      capabilities                    SEQUENCE OF Attribute OPTIONAL }
```

The following matching rule matches on the ORAddress part:

```
address-part-Match      MATCHING-RULE ::= {
      SYNTAX      ORAddress
      ID          id-mr-address-part-Match }
```

The following matching rule matches on the capabilities:

```
capabilities-part-Match MATCHING-RULE ::= {
      SYNTAX      AttributeValueAssertion
      ID          id-mr-capabilities-part-Match }
```

For 1993 systems, actual evaluation of assertions would use the equality matching rule associated with the capability attribute presented in the assertion. The returnMatchedValues extension to the Directory Abstract Service could be used to return only the values of the attribute which matched.

Matching rules could be defined for the syntax proposed in the working draft but would require tedious enumeration to take into account all of the component of the syntax and the extensions.

Automatic construction of a filter by an MTA or an MHS UA for multiple capabilities may result in a filter that exceeds the limits of the DSA holding the recipient's entry.

In 1988 systems, all values of the mhs-or-addresses-with-capabilities would be returned.

In addition, we propose adding the following attribute to identify the delivery method supported by an ORAddress because it is generally useful to the messaging community.

```
mhs-delivery-method ATTRIBUTE
      WITH ATTRIBUTE SYNTAX Mhs-delivery-method
      MULTI VALUE
      ::= id-at-mhs-delivery-method

Mhs-delivery-method     ::= INTEGER {
      mhs-delivery (1),
      physical-delivery (2),
      telex-delivery (3),
      teletex-delivery (4),
      g3-facsimile-delivery (5),
      g4-facsimile-delivery (6),
      ia5-terminal-delivery (7),
      videotex-delivery (8),
      telephone-delivery (9) }
```

NOTE - Mhs-delivery-method includes selected delivery methods from preferredDeliveryMethod in CCITT X.520|ISO/IEC 9594-6.

A.2.7 OIW Application Specific Object Classes

There are no application specific object classes defined by these Implementation Agreements.

A.2.8 Structure Rules

This clause defines the naming and structure rules for the MHS object classes which are subclasses of top.

A.2.8.1 MHS Distribution List

Attribute commonName is used for naming.

The mhs-distribution-list, organization, organizationalUnit, organizationalRole, organizationalPerson, locality, or groupOfNames can be immediately superior to entries of object class mhs-distribution-list.

A.2.8.2 MHS User

The naming for mhs-user is that of organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality.

The organizationalPerson, residentialPerson, organizationalRole, organizationalUnit, organization, or locality object classes can be combined with the mhs-user object class to form a new composite object class.

A.2.9 Use of Capabilities Information

The capabilities information in the X.500 Directory should not be considered sufficient to warrant a non-delivery decision by an originating or relaying MTA. This clause is not intended to impose any conformance requirement.

Annex B (normative)

IPM Body Part Support

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

A UA must support those IPM body part types defined in Annex E of X.420 (1988) as listed and qualified in AMH22. Support for reception means that the UA can receive the body part's encoding and, in the case of text body parts, accept all the character encodings in the supported repertoire(s). If an implementation supports a particular body part type for reception, it should also be able to support that body part type for reception if it is part of a forwarded message. If an implementation supports origination of forwarded messages, it must be capable of forwarding every body part that is supported on reception. The reception requirements on the UA do not necessarily include the ability to render (display) all of the characters received. If the message is forwarded, the UA must transmit exactly equivalent characters, but not necessarily from the same character set.

```
BodyPart              ::=  CHOICE {
  ia5-text                   [0] IA5TextBodyPart,
                             .
  oda-1984                   [12] IMPLICIT OCTET STRING,
  iso-6937                   [13] ISO6937BodyPart,
  bilaterally-defined        [14] Unidentified,
  externally-defined         [15] ExternallyDefinedBodyPart,
                             .
                             .
                             [310] IMPLICIT
                                  USAPrivatelyDefinedBodyParts,
                             .        }

Unidentified := OCTET STRING

The content of the ODA OCTET STRING will contain a value of
type ODABodyPart as follows:

ODABodyPart ::= SEQUENCE {
   ODABodyPartParameters,
   ODAData }

The Parameters and Data components are defined in Annex E
of CCITT Recommendation T.411 (1988) (ISO 8613-1).

USAPrivatelyDefinedBodyParts are defined as:

                           SEQUENCE {BodyPartNumber, ANY}

BodyPartNumber        ::=  INTEGER

These privately-defined body part types are specified as an
interim measure to provide backward compatibility with 1984
MHS implementations.  For interworking between UAs based on
the 1988 (or later) MHS standards, it is strongly
recommended that the externally-defined body part be used
instead.

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

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

Figure B1 - Privately-Defined Body Parts

Annex C (normative)

Object Identifiers

C.1 X.400 SIG Object Identifiers

The X.400 SIG object identifiers all allocated under the *mhsig* node in the OIW object identifier subtree, as defined in part 6 of the Stable Implementors Agreements document. This definition is duplicated in the following figure.

```
id-mhsig  OBJECT IDENTIFIER  ::=
            { iso (1)  identified-organization (3)  oiw (14)  mhsig (6) }
```

Figure C1 - Definition of the *mhsig* Object Identifier

The X.400 SIG has defined several categories of object identifiers. Their definition is provided in the following figure.

```
id-mhsig-content-types    OBJECT IDENTIFIER  ::=
                            { id-mhsig  content-types (0)  }

id-mhsig-body-part-types  OBJECT IDENTIFIER  ::=
                            { id-mhsig  body-part-types (1)  }
```

Figure C2 - Defintion of the X.400 SIG Object Identifier Categories.

C.2 Content Types

There are presently no object identifiers for content types allocated by the X.400 SIG.

C.3 Body Part Types

The object identifiers for the external body part types allocated by the X.400 SIG are defined in the following figure.

```
id-privacy-enhanced-mail  OBJECT IDENTIFIER  ::=
                            { id-mhsig-body-part-types  pem (0) }
```

Figure C3 - Definition of the External Body Part Object Identifiers

C.4 Security Classes

Editor's Note - Identical to the ISP.

Annex D (informative)

Interpretation of Elements of Service

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

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

Reply Request Indication: The reply-recipients and the reply-time may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request.

> **NOTE -** For an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Forwarded IP-message Indication: The following use of the original encoded information type in the context of forwarded messages is clarified:

> a) The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.
>
> b) If forwarding a privately defined body part (see Figure B1), the originator of the forwarding message shall set the original encoded information types in the P1 envelope to Undefined for that body part.

Annex E (informative)

Recommended Practices

This clause provides guidelines on areas not addressed by the base standards. These guidelines have been produced in order to promote awareness of interim solution to problems as agree by members of the OIW X.400 SIG. However implementors of these recommended practices should note that it is not necessary to follow the recommended practices when claiming conformance to these agreements.

Implementors should also note that future standardization by CCITT and ISO/IEC on area covered by this clause may result in different solutions to those proposed in this clause.

E.1 Printable String

There are existing mail systems that include a small set of non-Printable String characters in their identifiers. For these systems to communicate with MHS systems, either for pass-through service or delivery to MHS users, gateways will be employed to encode these special characters into a sequence of Printable String characters. This conversion should be performed by the gateway according to a common scheme and before insertion in Domain Defined Attributes, which are intended to carry electronic mail identifiers. MHS UAs may also perform such conversions.

It is recommended that the following symmetrical encoding and decoding algorithm for non-Printable String characters be employed. The encoding algorithm maps an ASCII representation to a PrintableString representation. Any non-printable string characters not specified in Table E1 are covered by the category "other."

Table E1 - Printable String to ASCII Mapping

ASCII Character	Printable String Character
% (percent)	(p)
@ (at sign)	(a)
! (exclamation)	(b)
" (quote mark)	(q)
_ (underline)	(u)
((left paren.)	(l)
) (right paren.)	(r)
other	(3DIGIT)

where 3DIGIT has the range 000 to 377 and is interpreted as the octal encoding of an ASCII character.

To encode an ASCII representation to a PrintableString, Table E1 and the algorithm in Figure E1 should be used.

```
IF current character is in the encoding set THEN
  encode the character according to Table E1
ELSE
  write the current character;
continue reading;
```

Figure E1 - ASCII to PrintableString Algorithm

To decode a PrintableString representation to an ASCII representation, Table E1 and the algorithm in Figure E2 should be used.

```
IF current character is not "(" THEN
  write character
ELSE
  {
  look ahead appropriate characters;
  IF composite characters are in Table E1 THEN
     decode per Table E1
  ELSE
  write current character;
  }
continue reading;
```

Figure E2 - PrintableString to ASCII Algorithm

E.2 Rendition of IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations as defined in the following table.

Table E2 - Interpretation of Format Effector Combinations

Combination	Interpretation
CR LF	to start a new line
CR FF	to start a new page (and line)
LF .. LF	to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect. The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition.

> **NOTE -** X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

E.3 EDI Use of MHS

E.3.1 P0 Recommended Practice

This section outlines a recommended method for interworking between a P(edi) UA with a UA implementing the Recommended Practice (EDI Use of X.400) in parts 7 and 8 of the OIW Stable Implementation Agreements. That Recommended Practice is commonly referred to as the "P0" approach to EDI use of the X.400 MTS.

This section does not define where the conversion between the two content types occurs. It is possible for the conversion to be performed by the P0 UA, the P(edi) UA, or a gateway. The Recommended Practice outlined in this section only attempts to document the rules that should be followed to ensure a conversion which retains the maximum amount of information.

E.3.1.1 P0 to P(edi) Conversion

The converting entity may assume that the P0 content contains only one EDI interchange. This interchange will become the first and only body part of the EDIM.

The content type field of the message will have the value "undefined" before the conversion and will have the integer value "35" or the object identifier value for P(edi) which is specified in X.435 after conversion. The EDIM Heading fields can be formed using the following rules:

EDIMIdentifier: Originator ORName concatenated with the UTCTime at which the conversion from P0 to P(edi) was performed.

Originator: Originator ORName.

Recipients: Recipients from the P1 envelope. EDI Notification Requests are not specified as none are requested when using the P0 approach.

EDIBodyPartType: This element may have one of deveral values depending on the encoded information type (EIT) value of the P0 message or the ability of the converting entity to determine which

EDI syntax is present in the content:

a) X.435-defined value for ANSI X12/EBCDIC if the EIT field of the P1 envelope has the value "undefined."

b) X.435-defined value for ANSI X12/ISO 646 if the EIT field of the P1 envelope has the value "IA5String."

c) Any other valid value if the entity performing the conversion can determine which EDI syntax is contained in the content and which character encoding is used for the EDI syntax.

Other heading fields will only be set if the entity performing the conversion is capable of parsing the EDI Interchange and discovering the correct values of EDI Heading fields.

As the P0 message will not contain requests for EDI Notifications, an EDI UA will never create an EDIN when it receives an EDIM converted from P0 .

E.3.1.2 P(edi) to P0 Conversion

When converting a P(edi) content to a P0 content, the following rules apply:

The first body part of the EDIM will be copied to the content. **All other body parts of the EDIM will be discarded.**

The P1 envelope fields shall have the following values:

Content Type: Value for "undefined."

Originator: Originator ORName.

Recipients: Recipients from the EDIM Heading. An NN EDIN with NN Reason Code set to the value "unspecified" is created for each Recipient for whom a Notification Request was specified. The EDIN Originator is set to the Recipient ORName. It is recommended that the supplementary information field of the NN be used to provide additional information on the disposition of the EDIM.

Encoded Information Types (EITs): This element may have one of several values depending on the value of the EDI Body Part Type:

a) The EIT is set to "undefined" if the EDI Body Part Type is encoded with the EBCDIC character set.

b) The EIT is set to "IA5String" if the EDI Body Part Type is encoded using the ISO 646 (ASCII) character set.

c) A value is not present for the EIT if EDI Body Part Type does not contain one of the above mentioned values.

E.3.2 P2 Recommended Practice

As there are a substantial number of users in the NIST OIW community that implemented the CEC TEDIS "P2" approach to EDI use of the X.400 MTS, this section will also include text that describes interworking between a P(edi) UA and a P2 UA. This text is not maintained by the EDI Working Group of the NIST OIW X.400 SIG but is included for the convenience of our user community. Users intending to interwork between P2 and P(edi) User Agents should consult the current version of the EWOS/ETSI document "A/3331 - Functional Profile of an Electronic Data Interchange User Agent." This will ensure that the most up to date technical information is obtained.

E.3.2.1 Conversion from IPMS to EDIMS (P2 to P(edi))

It is assumed that there is one and only one body part in the IPM Message, and that this body part contains an EDI interchange.

The IPM becomes the first, and only, body part of the EDIM.

The EDIM Heading fields are set as follows:

EDIMIdentifier: Originator ORName concatenated with the LocalIPMIdentifier portion of the IPM Identifier.

Originator: Originator ORName.

Recipients: Recipient ORNames from the IPM Heading. The edi-notification-requests-field is not coded.

EDIBodyPartType: The value is a local implementation issue. If the entity performing the conversion can identify the EDI syntax of the EDI Interchange then it can specify an appropriate value. Otherwise, the entity must be assuming a specific encoding and will specify the value for the syntax it is assuming.

Other heading fields may be set if the entity performing the conversion is capable of parsing the EDI Interchange and discovering the correct values of the EDIM Heading fields.

Since there are not notification requests, the EDI UA will never create an EDIN when it receives a converted EDIM and therefore the action for handling EDINs in the reverse direction does not need to be considered.

E.3.2.2 Conversion from EDIMS to IPMS (P(edi) to P2)

> **NOTE -** The verification of authority to perform a particular conversion is outside the scope of this annex. It is assumed that such conversion will be done with the full knowledge of the originating and recipient parties.

The EDIBodyPart of the EDIM will be copied to the IPM body as an IA5TextBodyPart. All other body parts of the EDIM will be discarded.

The IPM Heading fields are set as follows:

IPM Identifier: EDIMIdentifier.

Originator: Originator ORName.

Recipients: Recipients from the EDIM Heading. All recipients become IPM Primary Recipients. An NN EDIN with NN Reason Code set to the value "unspecified" is created for each Recipient for whom a Notification Request was specified. The EDIN Originator is set to the Recipient ORName. The EDIN Originator is set to the Recipient ORName. IPM Notifications shall not be requested.

Subject: Not present or set to a single blank character.

If EDINs have been requested the originator will always receive an NN. Since no IPM notifications are requested, the IPM UA will never create an IPM notification when it receives an IPM converted from an EDIM and therefore handling of notifications in the reverse direction does not need to be considered and is not an option for generating EDINs.

E.4 ODA Transfer

To ease interworking with 1984 implementations when transferring Office Document Architecture (ODA) documents, the following are recommended for 1988 implementations:

a) Origination UA implementing 1988 Implementation Agreements. The 1988 will generate the ODA according to CCITT Recommendation T.411 Annex E for the destination UA(s) implementing 1988 Implementation Agreements. If the destination UA supports 1984 Implementation Agreements, the approach as described in section 7.12.8 is recommended.

b) Recipient UA implementing 1988 Implementation Agreements. The recipient system will be able to handle the ODA bodypart in P2 (1984) as defined in section 7.12.8 for interworking with 1984 implementation, and will also be able to handle the ODA bodypart as defined in the appropriate base standards.

c) MTA downgrading rules. When transferring an P22 with ODA body part in P22 as described in T.411 to an 1984 MTA, the EITs identified by ODA Object Identifiers are mapped to bits 0 and 10 of the built-in EITs.

If the UA does not register to support P22 or ODA bodypart, a Non-Delivery-Report will be generated as required.

E.5 Use of Externally Defined Body Part

E.5.1 General

An Externally Defined body part represents an information object whose semantics and abstract syntax are denoted by an Object Identifier which the body part carries. This body part type enables the exchantge of information objects of all kinds, each unambiguously and uniquely identified.

The Externally Defined Body Part definition is reproduced in Figure E3.

```
ExternallyDefinedBodyPart   ::= SEQUENCE {
  parameters                    [0] ExternallyDefinedParameters  OPTIONAL,
  data                            ExternallyDefinedData  }

ExternallyDefinedParameters ::= EXTERNAL
ExternallyDefinedData       ::= EXTERNAL

EXTERNAL                    ::= [UNIVERSAL 8]  IMPLICIT SEQUENCE  {
  direct-reference              OBJECT IDENTIFIER  OPTIONAL,
  indirect-reference            INTEGER  OPTIONAL,
  data-value-descriptor         ObjectDescriptor  OPTIONAL,
  encoding                      CHOICE  {
    single-ASN1-type              [0]  ANY,
    octet-aligned                 [1]  IMPLICIT OCTET STRING,
    arbitrary                     [2]  IMPLICIT BIT STRING  }  }
```

Note - In the case of transfer of EXTERNAL in P2 BodyPart, the direct-reference component is mandatory and the indirect-reference and data-value-descriptor components must be absent.

Figure E3 - Externally Defined Body Part Definition

On the basis of the Externally Defined body part type, all body part types are divided into two important classes as follows:

a) *basic:* Said of any body part type except Externally Defined. All basic body part types are denoted by an integer (an ASN.1 context-specific tag) and are defined in section 7.3 of X.420.

b) *extended:* Said of the Externally Defined body part type restricted to any one value of the Direct-reference component of the Data component of such a body part. Denoted by an Object Identifier.

Annex B of Recommendation X.420 defines some (but not necessarily all) extended body part types.

E.5.2 Use of Equivalents of Basic Body Part Types

For each basic body part types, section B.1 of Recommendation X.420 defines an equivalent extended body part type. In order to facilitate interworking with 1984 systems, use of these extended body part types is not recommended; the basic body part types should be used instead.

Editor's Note: The requirements of this clause may change when interworking with 1984 systems is no longer critical.

E.5.3 Use of General Text Body Part Type

Unless otherwise specified in these agreements (e.g., IA5Text, 6937Text, Teletex) the General Text body part as defined in ISO 10021-7 Annex B.2 is the preferred means of supporting unstructured text body parts. The character set registration referred to in that annex is provided by ECMA.

E.5.4 Use of File Transfer Body Part Type

The File Transfer body part type is the recommended mechanism for the exchange of complex computer data via intra- and inter-company X.400 messages. It enables automatic type recognition for the file being sent and, possibly, automatic invocation of the appropriate application necessary to process the data.

E.5.4.1 Encoding of General Identifier

In order to optimize the machine-processing of information encoded in the Parameters and to enable registration, it is recommended that, if present, General Identifiers should be encoded as Object Identifiers.

E.5.4.2 Encoding of Contents Type

It is recommended that the Contents Type parameter be encoded as document type. The encoding as constraint-set-and-abstract-syntax has been provided only for backward compatibility with FTAM and its use is discouraged.

E.5.4.3 Encoding of Application Specific Information

The type of a file can be considered from several perspectives:

a) As a specific data structure consisting of a sequence of presentation data values - the position taken by the FTAM standard;

b) As the output of a certain application - the position taken by e-mail users requiring the interchange of office documents.

The fact that registered OSI document types have to be recognized by FTAM implementations and be described according to the requirements of ISO/IEC 9834-2 "Registration procedures for OSI document types" makes use of the Contents Type parameter inappropriate for expressing point of view (b).

Considering that the environment parameter "application-reference" could describe not only the application that generated a document but, more generally, the application-level format of the document, it is recommended that the values given to the "application-reference" parameter component be Object Identifiers associated with such a format.

Example: If an Object Identifier has been associated with a certain word-processing file format then this Object Identifier should be used as the value of "application-reference" when a file of that format is carried by a File Transfer body part, while the Content Type parameter should have as its value the Object Identifier associated with the "unstrucutred-binary" document type.

E.5.4.4 EITs for the File Transfer Body Part

It is recommended to use only the id-eit-file-transfer Object Identifier in association with the File Transfer body part.

The use of EITs describing other parameters of the File Transfer body part such as contents types, application references, etc., would force all potential recipients to register a possibly large number of EITs in order to avoid non-delivery of messages.

E.5.5 Use of Other Extended Body Part Types

The following are guidelines regarding the use of Externally Defined body part types not defined in the X.400 or other standards:

a) *Use of Parameters component:* In simple cases, to ease the integration of applications to X.400 systems, the Parameters component need not be used.

b) *Use of Data component:* For each different format of data, different Object Identifiers for the Data component are recommended. If an application chooses to use ASN.1 to format the data to achieve a single representation across platforms, the single-ASN1-type encoding choice should be used. Otherwise:

1) The octet- (i.e., byte) aligned choice is used if the data format is octet-aligned; or,

2) The arbitrary choice is used if the data is bit-aligned.

c) *Assignment of Object Identifiers:* Object Identifiers need to be assigned for the EXTERNALs, and these identifiers for the Parameters and Data components should be different. The Object Identifier for an EXTERNAL also indicates the syntax of the data encoding, i.e., whether single-ASN1-type or octet-aligned or bit-aligned is being used.

NOTE - Use of proprietary Externally Defined body part types is recommended only if the extended body part types already defined in the standards do not provide the apporpriate functionality.

In order to communicate with 1984 systems, the use of the Bilaterally Defined body part is recommended.

E.5.6 Obtaining Object Identifiers

There are many ways to obtain object identifiers. One such way is described as follows:

a) The application provider obtains a unique Numeric Name form for their organization from ANSI, as described in ANSI ISSB 840 and ISSB 843, and appends this number form to {iso (1) member-body (2) US (840)} to form an object identifier denoting their organization.

b) The application provider (organization) allocates a series of numbers to identify the application data format; these numbers are appended to the object identifier constructed in step (i) to form an object identifier that is globally unique. It is recommended that the application provider (organization) use a hierarchical structure for identifying their data types to ease the administration of the identifiers.

For example, company PCSoftware Inc. obtains the organization number "999" from ANSI. The PCSoftware SpreadSheet file for MS-DOS might be assigned the following object identifier.

NOTE - ASN.1 notation is used. The numbers in parentheses form the identifier, the associated words describe the number.

{ iso (1) member-body (2) US (840) PCSoftware Inc. (999) MS-DOS-Application (1) SpreadSheet (3) Data (1) }

E.6 Privacy Enhanced Mail Body Part

This clause describes a mechanism to convey an Internet Privacy Enhanced Mail (PEM) message across an X.400 MHS. PEM is described in Internet RFCs 1421, 1422, and 1423 and their successors.

The general Internet mail message format is described in RFC 822. Mapping of RFC 822 messages to and from X.400 Inter Personal Messages is described in RFC 987 for 1984 X.400 and in RFC 1148 for 1988 X.400.

The PEM message is conveyed as a P2(2) body part. All of the RFC 822 header information is conveyed in the P1 envelope and P2 header per RFC 987 and RFC 1148. The PEM message

(encapsulated security header and, possibly encrypted, message text as described in RFC 1113) is conveyed in a single body part. On the X.400 side, this body part may be manipulated like any other body part; e.g., it may be included in a multi-part body.

For 1988 (P22), the PEM body part is externally defined and does not require parameters. This definition is provided in the following figure.

```
privacy-enhanced-mail     EXTENDED-BODY-PART-TYPE
                            DATA  OCTET STRING
                      ::=  id-privacy-enhanced-mail

-- The object identifier is defined in annex B.
```

Figure E4 - Definition of the Privacy Enhanced Mail Body Part Type

For interworking with 1984 (P2) systems, a USA body part (integer) will be allocated by NIST as described in Figure B1.

E.7 Selection of OR Name Attributes

E.7.1 Teletex Name Attributes

To support the transition to addresses with Teletex components, it is recommended that a printable string alternative address be established for each address containing Teletex strings.

E.8 Use of the Teletex Body Part

The Teletex body part should be used purely for structured teletex documents, as described in F.200 and T.60, obeying page rules, etc. It should not be used to transfer T.61 characters, in a general sense, across the MTS. If only IA5 characters are being used, the IA5Text body part should be used, especially when interworking with 1984 UAs is relevant. Otherwise, the GeneralText body part should be used to transfer unstructured character data.

E.9 Provision of Security Class S0A Using Asymmetric Algorithms

This clause describes one method of providing the security services of class S0A when using asymmetric (public key) cryptographic algorithms. It is recommended that this method be used unless the security requirements or policy specifies otherwise. Asymmetric cryptographic algorithms such as RSA are used to provide digital signatures in support of the content integrity and (end-to-end) message origin authentication services, as well as proof of delivery. Since asymmetric algorithms are used, the nonrepudiation of origin and nonrepudiation of delivery services of security class S2 are also provided. Content confidentiality is provided using a combination of symmetric and asymmetric encryption. The following paragraphs discuss the protocol elements used to provide these services, as well as certificate management and other issues.

E.9.1 Protocol Elements

The following protocol elements are provided by the originating UA in the submission envelope in support of the S0A security services.

Content: If the content confidentiality services is required, the message content is encrypted under the content confidentiality key.

Content Integrity Check: This per-recipient security element is a signature over the message content, and provides the content integrity, message origin authentication, and nonrepudiation of origin services if content confidentiality is not required. (If the message is encrypted, the content integrity check is included in the message token.)

> **NOTE -** The message origin authentication check provides a single signature, rather than a signature per recipient, thus reducing total message size in the case where multiple recipients are present. However, support for this protocol element is optional for security class S0. In addition, it is computed over the message content as sent (i.e., the encrypted content if content confidentiality is used). If the content is encrypted, this protocol element does not truly provide nonrepudiation of the unencrypted content. In this case, smaller message size was traded off for the additional service of nonrepudiation.

Proof Of Delivery Request: This per-recipient security element is used to request the recipient to generate a proof of delivery, in the case where content confidentiality is not used. (Where content confidentiality is used, the proof of delivery request is included in the message token, as shown below.)

Originator Certificate: This security element is a set of one or more certificates which the recipient may use to obtain the oroginator's public key. For example, it might contain the chain of certificates from the originator, through the certification hierarchy to a top-level certification authority.

Message Token: The asymmetric message token conveys security information from an originator to a single recipient. It is a signed structure, some of whose fields may be encrypted. The message token is used only when content confidentiality is desired, and supports the content integrity, message origin authentication, content confidentiality, and nonrepudiation of origin services. The following fields are required, and all other fields are optional:

- *Signature Algorithm Identifier:* The algorithm identifier of the asymmetric algorithm used to

sign the token.

- *Recipient Name:* The OR Address and/or Directory Name of the recipient with whom the token is associated. Since the encrypted portion of the token is encrypted under the recipient's public key, it is recommended that the directory name be included, since the recipient's certificate contains his/her directory name rather than OR Address.

- *Time:* The time of day when the token was generated.

- *Signed Data:* The following fields are signed but not encrypted:

a) *Content Confidentiality Algorithm Identifier:* The algorithm to be used to encrypt the message content.

b) *Proof of Delivery Request:* This element is used to request the recipient to compute a proof of delivery over the received message.

- *Encrypted Data:* These fields are encrypted under the recipient's public key:

c) *Content Confidentiality Key:* The symmetric key used to encrypt the message content.

d) *Content Integrity Check:* A signature on the unencrypted message content. If content confidentiality is required, this element provides the content integrity, message origin authentication, and nonrepudiation of origin services. This signature is encrypted in order to protect against the "low entropy" attack described in Internet RFC 1113. (In RFC 1113, the signature is encrypted under the content confidentiality key.)

NOTE - The encrypted portion of the token will then comprise two RSA encryption blocks.

The following element of service is generated by the recipient, if requested by the originator.

Proof Of Delivery: This security element provides proof and nonrepudiation of delivery. It is a digital signature computed over the received (possibly encrypted) message content and various delivery envelope fields, as defined in the base standard.

E.9.2 Algorithm Selection

This clause makes no recommendation as to hash algorithms, asymmetric encryption algorithms, or symmetric encryption algorithms. The implementor must select appropriate algorithms, based on factors such as performance, cost, and licensing and export restrictions. A fairly complete list of algorithms can be found in clause 7 (Security Algorithms) of Part 12 of these Agreements. In some cases, the implementor must also specify certain algorithm-dependent information. For example, when using the symmetric algorithm **DES-CBC**, the implementor must specify the padding mechanism used, since this algorithm operates on 8-byte input blocks. Internet RFC 1115 defines such padding rules for DES and RSA in various modes, and these mechanisms are recommended unless security requirements dictate otherwise. PKCS #1 (see Bibliography, Annex F) discusses such matters in more detail.

E.9.3 Certificate Management

Management of public key certificates is beyond the scope of this recommended practice. X.509 provides a generic authentication framework which uses the Directory to store certificates. In the absence of a ubiquitous Directory, local means may be used to obtain certificates. For example, the recipient of a message might choose to cache those certificates received in the **OriginatorCertificate** protocol element of the delivery envelope.

Each community of interest will define its own policy regarding certificate management and the associated trust model. An example of a centralized trust model can be found in Internet RFC 1114, while the most complete example of a decentralized trust model can be found in the paper on Digital's Distributed System Security Architecture cited in the Bibliography (Annex F).

E.9.4 Other Issues

In the case of the P2 content type, addressing information may be protected by replicating the P1/P3 recipient names in the P2 heading fields (To:, CC:, and BCC:). The X.400 security services discussed above are applied to the entire P2 IPM, including the heading and all body parts. Additional protection of heading and envelope fields may be provided using double enveloping.

When using X.400 (1988) distribution lists (DLs), one might choose to distribute the private key associated with the DL to all members of the DL. This allows an originator to create a single message token in which the content confidentiality key is encrypted under the DL's public key. (This requires support of the DL expansion history protocol element on delivery, so that the recipient may select the proper private key for decryption. Alternatively, the originating UA may expand the DL locally and generate a message token for each member (recursively). There is no architected support for this mechanism in the base standard, nor is there architected support for performance of this function by an MTA when expanding a DL.

Annex F (informative)

Bibliography

F.1 ANSI

Procedures for Registering Organization Names in the United States of America, ISSB 843, December 5, 1989.

Procedures for Registering Names in the United States of America, ISSB 840, December 5, 1989. The U.S. Register is included.

F.2 Internet

Message Encipherment and Authentication Procedures, RFC 1421.

Certificate-based Key Management, RFC 1422.

Algorithms, Modes, and Identifiers, RFC 1423.

F.3 Other References

RSA Data Security, Inc., "PKCS #1: RSA Encryption Standard," June 1991.

Gasser, M., A. Goldstein, C. Kaufman and B. Lampson, "The Digital Distributed System Security Architecture," Proceedings of the 12th National Computer Security Conference, 1989.

Annex G (informative)

Defense Message Handling Profiles

G.1 Introduction

Several additional requirements for Message Handling Systems (MHS) are currently being investigated by the U.S. DoD Data Communications Protocol Standards (DCPS) Technical Management Panel (DTMP). This annex describes the DoD Standardized Profile(s) (DSP) that are required for Defense Message System (DMS) use.

Two multipart DoD profiles are currently defined, namely:

- DSP AMH1n(D) - Information Technology - Defense Standardized Profiles AMH1n(D) - Message Handling Systems - Common DoD Messaging

- DSP AMH2n(D) - Information Technology - Defense Standardized Profiles AMH1n(D) - Message Handling Systems - Military Messaging

These profiles will be published as part of the MIL-STD-2045 series. The AMH1n(D) profile consists of a DoD delta to the AMH1n ISP. AMH2n(D) is a standalone profile of a new military messaging content type (P772) based on the IPM content type. These extensions support military-unique functionality required by the DMS.

For further information on these profiles, contact:

DTMP WG/2 Chairman
c/o Defense Information Systems Agency (DISA)
Joint Interoperability Engineering Office (JIEO)
Code TBBD
Fort Monmouth, NJ 07703-5000
Phone: 908-532-7726

Annex H (informative)

Management Domains

The sections above describe agreements among implementors of particular X.400 components (e.g. MTAs, UAs, MSs). There are some agreements that don't apply to a single X.400 component, but instead apply to an entire domain of X.400 components. This section details any requirements for X.400 domains, independent of those for individual X.400 components. A single X.400 component cannot be conformance tested for these domain requirements, but for a domain to claim to be "operationally OIW compliant", it must abide by the rules stated below.

H.1 Management Domain Names

This section contains requirements on matters being considered by the U.S. CCITT Study Group D for national decisions. Such decisions are likely to supersede the relevant portions of this clause.

The Implementation Agreements for 1984-based MHS implementations requires that all Management Domain Names (both Private and Administration) shall be unique within the U.S. This is also a requirement for 1988-based MHS implementations.

A "Construction Syntax" is defined, which uses a registered OSI Organization Name from the ANSI US Register of Organization Names as a "root" in the construction of MHS Management Domain Names e.g., ADMD and PRMD). The constructed combinations based on this "root" will be guaranteed to be unique, and thus be safely used as MHS MD names in the United States. Other countries may wish to adopt these same rules.

MHS MD (PRMD and ADMD) names shall be constructed according to the Extended BNF grammar shown in the following figure.

```
<ADMDName> ::= <MDName>

<PRMDName> ::= <MDName>

<MDName> ::=
    <NationalOrganizationName> |
    <ConstructedName> |
    <NationalOrganizationNumber>

<ConstructedName> ::=
    <NationalOrganizationName>"+"<OrganizationallyDeterminedPart>
```

Figure H1 - Management Domain Name Construction

Subject to all of the following rules:

Rule 1. The entire <MDName> must not exceed 16 bytes (including any constructor operators that may be included, and shall be composed entirely of PrintableString characters.

Rule 2. The <NationalOrganizationName> shall be drawn from the alphanumeric names registered in the US Register. It shall contain at least one non-numeric character, and not contain the constructor operator "+" (plus sign).

Rule 3. Each <NationalOrganizationName> obtained from the US Registry will be accompanied by a NumberForm (numeric value) which shall be bound as the <NationalOrganizationNumber> to the <NationalOrganizationName>.

Rule 4. In a <ConstructedName>, the <OrganizationallyDeterminedPart> shall be certified to be unique under the <NationalOrganizationName> (sub)authority, by the <NationalOrganizationName> registration authority.

Rule 5. A <NationalOrganizationNumber> shall be obtained from the US Register and bound to the <ConstructedName>.

Rule 6. A Private Management Domain's PrivateDomainIdentifier shall be the same as its PrivateDomainName.

NOTES

1 The PRMD names resulting from the <ConstructedName> syntax (those having a "+" in them) are atomic values from the point of view of the MTA -- in particular, it is not permissible for the MTA to route on components of the PRMD name.

2 The construction rules are such that if ABC is a Registered National Organization Name, then the owner of that name controls the MHS Domain Name space including "ABC" and "ABC+<anything>", but not "ABC<anything>."

3 A "+" is legal in an ANSI provided name.

4 If a Registered Organization Name already contains the construction operator ("+" sign), then in order to use the name as an <MDName>, its owner must also register the "root" which precedes the first "+" sign, with the US Register of Organization Names. (e.g., company B+Z+P would need to register "B" to be able to use the "constructed" name of B+Z+P.)

5 For the special case of the construction operator ("+" sign) being the first character of a Nationally Registered Name, no special action is required beyond its normal registration with the US Registry of Organization Names.

6 If the sub-authority determined by <NationalOrganizationName> so wishes, the <OrganizationallyDeterminedPart> can be constructed using rules similar to the above, resulting in a hierarchical construction separated by "+"s. In particular, the sub-authority must maintain its own registry and might (for example) define the <OrganizationallyDeterminedPart> using the syntax shown in the following figure.

```
<OrganizationallyDeterminedPart>  ::=  <DivisionName>
      |  <DivisionName> "+" <DivisionallyDeterminedPart>
```

Figure H2 - Name Construction by Subauthorities

where the <DivisionName> is drawn from the sub-authority's registry (and does not contain a "+"). Thus the sub-authority can delegate the use of the prefix described in the following figure.

```
<NationalOrganizationName>+<DivisionName>
```

Figure H3 - Prefix

to someone else.

H.2 Use of ADMD Names

This subsection was developed by an X.400 SIG working group in April, 1990. It contains extremely controversial positions that invoke national, commercial, and quality of service issues. The OIW may not be the correct forum to make these national decisions. Until these decisions can be reached or a national forum established, this section remains as a placeholder in the OIW X.400 SIG Working Text document only.

> **NOTE -** Version 2 of the CCITT X.400 Implementors Guide, dated 16 March 1990, allows for a single zero ("0") character as the ADMD name for the case of a PRMD that is not reachable from any ADMD. The following discussion does not apply to such PRMDs.

A PRMD may be directly connected to more than one ADMD. Since a PRMD may not alter the originators ORAddress, the Country/ADMD name pair provided in the Originator ORAddress may not match those of the first ADMD to receive the message from the PRMD. The first ADMD is required to accept such messages and may not alter the originator's ORAddress.

Any message originated by a PRMD must have an Originator's ORAddress that either uses the single space ADMD name or uses a Country/ADMD name pair for an ADMD to which the PRMD is connected. (In both cases the Country name is required.)

The X.400 Recommendations have defined a mechanism that enables PRMDs connected to multiple ADMDs to enter a single space as the ADMD name. To support this, these agreements recognize two classes of ADMDs. ADMDs in the first class, "space-supporting" ADMDs, must be able to route on PRMD name, independently from the ADMD name. Furthermore, the space-supporting ADMDs must arrange their routing configuration such that all PRMDs are reachable from all ADMDs. PRMDs using the single space ADMD name must be connected to at least one space-supporting ADMD.

ADMDs in the other class, "non-space-supporting" ADMDs, must, at a minimum, route messages for which the ADMD name is a single space to a space-supporting ADMD (in the indicated country). It is hoped that in the long term, all ADMDs will be able to route on the PRMD name when the ADMD name is a single space.

H.3 Uniqueness of MTS Identifiers Within a Management Domain

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

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