

IEEE Standards for Information Technology—Test Methods for Measuring Conformance to X.400-Based Electronic Messaging—Application Program Interface (API) [Language Independent]

Sponsor

**Portable Applications Standards Committee
of the
IEEE Computer Society**

Approved March 18, 1993

IEEE Standards Board

Approved September 16, 1993

American National Standards Institute

Abstract: The test methods that may be used to test conformance to IEEE Std 1224.1-1993, IEEE Standard for Information Technology—Open Systems Interconnection (OSI) X.400-Based Electronic Messaging—Application Program Interface (API) [Language Independent], are defined.

Keywords: API, OSI, X.400, application portability, electronic messaging, language independent, open systems interconnection, test methods

The Institute of Electrical and Electronics Engineers, Inc.

345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1993 by the Institute of Electrical and Electronics Engineers, Inc.

All rights reserved. Published 1993. Printed in the United States of America

ISBN 1-55937-309-1

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE standards documents may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers, Inc. does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Introduction

(This introduction is not a normative part of IEEE Std 1326.1-1993, IEEE Standard for Information Technology—Test Methods for Measuring Conformance to X.400-Based Electronic Messaging Application Program Interface (API) [Language Independent], but is included for information only.)

Purpose

The purpose of this standard is to define test methods for the language independent X.400-based electronic messaging API confined in IEEE Std 1224.18 -1993 {5}.

Related Standards

IEEE Std 1224.1-1993 {5} is intended to provide the basis for the definition of programming language bindings to which implementations and applications can conform. A specification for such a language binding, for the C programming language, is contained in IEEE Std 1327.1-1993 {B3}. This standard applies to test methods for measuring conformance to any such programming language binding specification.

The API defined in IEEE Std 1224.1-1993 {5} uses the mechanism for OSI abstract data manipulation (OM) defined in IEEE Std 1224-1993 {4}. IEEE Std 1326-1993 {6} defines the requirements that shall apply to test methods for measuring conformance to IEEE Std 1224-1993 {4}. A set of test methods used to measure conformance to IEEE Std 1224.1-1993 {5} shall satisfy the requirements of IEEE Std 1326-1993 {6}, as well as conforming to this standard.

IEEE Std 1003.3-1991 {3} defines the general requirements that shall apply to test methods for measuring conformance to POSIX. A set of test methods used to measure conformance to IEEE Std 1224.1-1993 {5} shall satisfy the requirements of IEEE Std 1003.3-1991 {3}, as well as conforming to this standard.

Overview

For each section of IEEE Std 1224.1-1993 {5}, this standard contains a corresponding section containing test assertions, in accordance with IEEE Std 1003.3-1991 {3}. A set of test methods conforms to this standard if it tests all of the test assertions.

Related Standards Activities

The following areas are under active consideration at this time, or are expected to become active in the near future, concerning extensions to this standard and similar efforts can be anticipated in the future:¹

- 1) Directory services
- 2) FTAM API
- 3) Verification testing methods
- 4) Network interface facilities
- 5) System administration

Extensions are approved as “amendments” or “revisions” to this document, following IEEE Standards procedures.

Approved amendments are published separately and distributed with orders from the IEEE for this document until the full document is reprinted and such amendments are incorporated in their proper positions.

If you have any question about the completeness of your version, you may contact the IEEE Standards Department [+1 (908) 562-3800] to determine what supplements have been published.

If you have an interest in participating in the working groups addressing these issues, please send your name, address, and phone number to the Secretary, IEEE Standards Board, Institute of Electrical and Electronics Engineers, Inc., P.O. Box 1331, 445 Hoes Lane, Piscataway, NJ 08855-1331, USA, and ask to have this forwarded to the chairperson of the appropriate PASC working group.

¹A *Standards Status Report* that lists all current IEEE Computer Society standards projects is available from the IEEE Computer Society, 1730 Massachusetts Avenue NW, Washington, DC 20036-1903; Telephone: +1 202 371-0101; FAX: +1 202 728-9614.

This standard was prepared by the P1224 Working Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society. At the time this standard was approved, the membership of the P1224 Working Group was as follows:

Portable Applications Standards Committee (PASC)

Chair: Jim Isaak
Vice Chairs: Hal Jespersen
Lorraine Kevra
Pete Meier
Andrew Twigger
Treasurer: Peter Smith
Secretary: Shane McCarron

P1224 Working Group Officials

Chair: Steve Trus
Vice Chair: Tim Carter
Editor/Secretary: Iain Devine
Graham Jack

P1224 Working Group

Curt Campbell	Mark Karan	Enzo Signore
Tony Cincotta	Eva Kuiper	Daniella Sirocchi
Chris Harding	Kevin Montgomery	Andrew Twigger
Khiem Ho	Ed Owens	Bruce Weiner
Petr Janecek	Pierre Perrot	

The following persons were members of the P1224 balloting group that approved this standard for submission to the IEEE Standards Board:

Stephen R. Walli *EurOpen Organizational Representative*
Ralph Barker *Uniform Organizational Representative*

Timothy Baker	Steve Griesmer	Ronald K. Sellars
Ralph Barker	Charles Hammons	Sandra Swearingen
Andy Bihain	Michael Hannah	James G. Tanner
Mack W. Bishop	Mark Hazzard	Donn S. Terry
Robert Bismuth	Barry Hedquist	Alan E. Turner
George Carson	Khiem Ho	Mark-Rene Uchida
Tim Carter	Petr Janecek	Andrew Walker
Andy Cheese	Hal Jespersen	Paul Wanish
Fred D. Crowner	Robert Knighten	Peter Wishart
Ana Maria De Alvare	Kevin Lewis	Oren Yuen
Iain Devine	Roger Martin	George R. Zerdian
Stephen L. Diamond	Joberto S.B. Martins	Jason Zions
Philip H. Enslow	Diana Norwood	John Zolnowsky
Donald Folland	Edward Owens	
Michel Gien	Scott g. Preece	

The IEEE Standards Board approved this standard on March 18, 1993 with the following membership:

Wallace S. Read, *Chair*
Donald C. Loughry, *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ramiro Garcia
Donald N. Heirman
Jim Isaak

Ben C. Johnson
Walter J. Karplus
Lorraine C. Kevra
E. G. “Al” Kiener
Ivor N. Knight
Joseph Koepfinger*
D. N. “Jim” Logothetis
Don T. Michael*

Marco W. Migliaro
L. John Rankine
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp
Donald W. Zipse

*Member Emeritus

Also included are the following nonvoting IEEE Standards Boards liaisons:

Satish K. Aggarwal
James Beall

Richard B. Engelman
David E. Soffrin

Stanley I. Warshaw

Rachel Auslander
IEEE Standards Project Editor

CLAUSE	PAGE
1. General	1
1.1 Scope	1
1.2 Normative References	1
1.3 Conformance	2
2. Terminology and General Requirements	2
2.1 Conventions	2
2.2 Definitions	3
3. Test Assertions for Section General.....	4
3.1 Scope	4
3.2 Normative References	4
3.3 Conformance	4
3.4 Test Assertions	6
4. Test Assertions for Section Terminology and General Requirements.....	6
5. Test Assertions for Section Overview	7
5.1 Purpose	7
5.2 Overview	7
5.3 Object Management	7
5.4 Features	7
6. Test Assertions for Section Message Handling Interfaces.....	8
6.1 Datatypes	8
6.2 Access Operations	9
6.3 Transfer Operations.....	152
6.4 Return Codes.....	231
7. Test Assertions for Section Interpersonal Messaging Packages	235
7.1 Class Hierarchy	235
7.2 Class Definitions	235
7.3 Class Definitions	236
7.4 Syntax Definitions.....	266
8. Test Assertions for Section Message Handling Packages	272
8.1 Class Hierarchy	272
8.2 Class Definitions	273
8.3 Syntax Definitions.....	366
9. Test Assertions for Section Secure Messaging Package.....	381
9.1 Class Hierarchy	382
9.2 Class Definitions	382
9.3 Syntax Definitions.....	389

CLAUSE	PAGE
10. Test Assertions for Section Definition of Constants	389
10.1 Object Identifier for This Standard	389
10.2 OIDs for Features and Packages	389
10.3 Interpersonal Messaging Package OIDs	390
10.4 Message Handling Package Value Definitions	392
10.5 Secure Messaging Package Value Definitions.....	396
Annex A (Informative) Test Assertions for Annex Functional Architecture of API.....	398
Annex B (Informative) Bibliography.....	401

IEEE Standard for Information Technology—Test Methods for Measuring Conformance to X.400-Based Electronic Messaging—Application Program Interface (API) [Language Independent]

1. General

1.1 Scope

This standard defines requirements for test methods for measuring conformance to IEEE Std 1224.1-1993 {5}.

IEEE Std 1224.1-1993 {5} is stated in terms that are independent of any particular programming language, but each implementation that conforms to it implements a particular programming language binding, and conforms to its programming language binding specification. Each set of test methods for measuring conformance to IEEE Std 1224.1-1993 {5} assumes a particular programming language binding specification. Each programming language binding specification may impose further programming language specific requirements on the test methods which, in conjunction with the requirements imposed by this standard, constitute the requirements that shall be satisfied by test methods used for measuring conformance to that programming language binding specification. This standard applies to all sets of test methods for measuring conformance to any programming language binding specification for IEEE Std 1224.1-1993 {5}.

1.2 Normative References

The following standards contain provisions that, through references in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

{1} ISO 3166: 1988,¹ *Codes for the Representation of Names of Countries*.

{2} CCITT Recommendation X. 121: 1984,² *International Numbering Plan for Public Data Networks. CCITT Red Book*, Fascicle VIII.4.

{3} IEEE Std 1003.3-1991,³ *IEEE Standard for Information Technology—Test Methods for Measuring Conformance to POSIX*.

{4} IEEE Std 1224-1993, *Open Systems Interconnection (OSI) Abstract Data Manipulation—Application Programming Interfaces (API) [Language Independent]*.

{5} IEEE Std 1224.1-1993, *Information Technology—X. 400 Based Electronic Messaging Application Program Interface (API) [Language Independent]*.

{6} IEEE Std 1326-1993, *IEEE Standard for Information Technology—Test Methods for Measuring Conformance to Open Systems Interconnection (OSI) Abstract Data Manipulation—Application Programming Interfaces (API) [Language Independent]*.

1.3 Conformance

A set of test methods that conforms to this standard shall conform to IEEE Std 1003.3-1991 {3}, with references to the “POSIX.n test method specification” being interpreted as references to this standard, and references to “the POSIX standard for which conformance is being measured” being interpreted as references to IEEE Std 1224.1-1993 {5}.

In addition to meeting the conformance criteria defined in IEEE Std 1003.3-1991 {3}, a set of test methods that conforms to this standard shall test all documentation assertions defined in this standard.

NOTE — Conformance to IEEE Std 1003.3 {3} implies that the test methods will test all other assertions defined in this standard.

2. Terminology and General Requirements

2.1 Conventions

2.1.1 General and Typographical Conventions

Language-independent concrete OM class names, OM attribute names, and OM-attribute value names are spelled with hyphens between words (e.g., **conversion-loss-prohibited**). The first letters of language-independent OM class and OM attribute names are capitalized (e.g., **Redirection-Record**).

Language-independent datatype, operation, argument and error names are lowercase and are spelled with underscores between words (e.g., `ma_start_delivery`).

The use of fonts in this standard is as follows.

- The Helvetica font is used for:
 - Language-independent operation names, such as `ma_start_delivery`
 - Language-independent datatype names, such as `mh_interval_type`
 - Language-independent error names, such as `feature_unavailable`

¹ISO documents can be obtained from the ISO office, 1 Rue de Varembe, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse.

²CCITT documents can be obtained from the CCITT General Secretariat, International Telecommunications Union, Sales Section. Place des Nations, CH-1211, Genève 20, Switzerland/Suisse.

³IEEE documents can be obtained from the IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ, 08855-1331, USA. Telephone: 1(800)678-IEEE or +1 (908) 981-1393 (outside US).

- The *italic* font is used for:
 - Language-independent operation arguments, such as *user_address*
 - The introduction of important terms
 - Cross-references in 2.2
- The **bold** font is used for:
 - Language-independent concrete OM class names, such as **Delivered-Report**
 - Language-independent OM attribute names, such as **Algorithm-Datum**
 - Language-independent OM attribute values, such as **conversion-loss-prohibited**
- The ***bold italic*** font is used for:
 - Language-independent abstract OM class names, such as ***Delivery-Report***
- The **constant width** (Courier) font is used for:
 - References to terms defined in international standards upon which this standard is based.

2.2 Definitions

2.2.1 Terminology

For the purposes of this standard, the following definitions apply:

2.2.1.1 conformance document: A document provided by an implementor that contains implementation details. [ISO/IEC 9945-1 :1990 {B1}]

2.2.1.2 may: An indication of an optional feature. [ISO/IEC 9945-1 : 1990 {B1}]

With respect to implementations, the word *may* is to be interpreted as an optional feature that is not required in this standard, but that can be provided. With respect to Strictly Conforming Applications, the word *may* means that the optional feature shall not be used.

2.2.1.3 shall: An indication of a requirement on the implementation or on Strictly Conforming Applications, where appropriate. [ISO/IEC 9945-1 : 1990 {B1}]

2.2.1.4 should:

- 1) With respect to implementations, an indication of an implementation recommendation, but not a requirement.
- 2) With respect to applications, an indication of a recommended programming practice for applications and a requirement for Strictly Conforming Applications. [ISO/IEC 9945-1 :1990 {B1}]

2.2.1.5 supported: A condition regarding optional functionality.

Certain functionality in this standard is optional, but the interfaces to that functionality are always required. If the functionality is *supported*, the interfaces work as specified by this standard (except that they do not return the error condition indicated for the unsupported case). If the functionality is not *supported*, the interface shall always return the indication specified for this situation. [ISO/IEC 9945-1 : 1990 {B1}]

2.2.1.6 system documentation: All documentation provided with an implementation, except the conformance document.

Electronically distributed documents for an implementation are considered part of the system documentation. [ISO/IEC 9945-1 :1990 {B1}]

2.2.2 General Terms

For the purposes of this standard, the following definitions apply:

2.2.2.1 assertion: A statement that is derived from the standard to which conformance is being measured, that is true for a conforming implementation, and that pertains either to functionality or behavior of a functional interface or namespace allocation or to the documentation associated with the implementation being tested [IEEE Std 1224-1993 {4}].

2.2.2.2 generic interface: A version of an interface that is independent of any particular programming language.

2.2.2.3 programming language binding specification: For a language-independent specification, a document that specifies, in terms of a particular programming language, the behavior that the language-independent specification specifies in language-independent terms.

It may also specify additional behavior that is relevant to the usage of the particular programming language [IEEE Std 1224-1993 {4}].

2.2.2.4 test methods: The software, procedures, or other means specified to measure conformance to a specification [IEEE Std 1224-1993 {4}].

2.2.3 Abbreviations

For the purposes of this standard, the following abbreviations apply:

2.2.3.1 OM: Object Management.

3. Test Assertions for Section General

3.1 Scope

There are no test assertions for this clause.

3.2 Normative References

There are no test assertions for this clause.

3.3 Conformance

3.3.1 Implementation Conformance

3.3.1.1 Conformance Requirements

D01	The manufacturer shall identify the interface (MA, MT, or both) that the product implements, and state what roles (client, service, or both) it plays for each.
001(A)	The product shall implement the OM interface as defined in IEEE Std 1224-1993 {4}, satisfying its conformance requirements, and play the same roles for that interface as it plays for the MA and MT interfaces.
D02	If the product plays the role of service for the MA or MT interface: The manufacturer shall state which features the product implements.
002(C)	If the product plays the role of service for the MA interface: The product shall implement the Basic Access FU.
003(C)	If the product plays the role of service for the MA interface: The product shall implement the Submission FU, the Delivery FU, the Retrieval FU, or any combination of the three.
004(C)	If the product plays the role of service for the MT interface: The product shall implement the Basic Transfer FU.

- 005(C) If the product plays the role of service for the MT interface:
The product shall implement the Transfer In FU, the Transfer Out FU, or both.
- 006(C) If the product plays the role of service for the MA or MT interface:
The product shall implement the MH 88 Package, the MH 84 Package, or both.
- GA01 For *FU* of Basic Access FU, Submission FU, Delivery FU, Retrieval FU, Basic Transfer FU, Transfer In FU, Transfer Out FU:
If the product plays the role of service for FU:
The product shall implement every aspect of every function in FU.
- D03 For *package* of IM 88 Package, MH 88 Package:
If the product implements *package*:
The product documentation shall state which (if any) of the 1988 classes of *package* the product implements.
- DO4 For *package* of IM 88 Package, IM 84 Package:
If the product implements *package*:
The product documentation shall state which (if any) immediate subclasses of the **Body-Part** class the product implements.
- GA02 For *package* of IM 84 Package, SM 88 Package:
If the product implements *package*:
The product shall implement every class the *package* contains.
- 007(C) If the product implements the MH 84 package:
The product shall implement all of those classes that are used by the functional units that it supports.
- 008(A) The product shall implement the closures of all classes it implements.
- D05 The product documentation shall state what classes it provides the `om_encode` function for.
- D06 If the product plays the role of service for the MA interface:
The manufacturer shall state whether or not the product realizes the interface by means of X.400's *Message Submission and Delivery Protocol (P3)* or *Message Retrieval Protocol (P7)*.
- D07 If the product plays the role of service for the MT interface:
The manufacturer shall state whether or not it realizes the interface by means of X.400's *Message Transfer Protocol (P1)*.
- D08 If the product plays the role of service for the MA interface or the MT interface and implements the IM 88 Package or the IM 84 Package:
The manufacturer shall state whether or not the product realizes the interface by means of X.400's *Interpersonal Messaging Protocol (P2)*.
- GA03 For *protocol* of P1 (1988), P3 (1988), P7 (1988):
If the product implements *protocol*:
The manufacturer and product shall satisfy the conformance requirements of X.400 (1988) with respect to *protocol*.
- 009(D) If the product implements P1 (1988):
The manufacturer and product shall satisfy the conformance requirements of X.400 (1988) with respect to P1 (1988). (See GA03 in 3.3.1.1.)
- 010(D) If the product implements P3 (1988):
The manufacturer and product shall satisfy the conformance requirements of X.400 (1988) with respect to P3 (1988). (See GA03 in 3.3.1.1.)

- 011(D) If the product implements P7 (1988):
 The manufacturer and product shall satisfy the conformance requirements of X.400 (1988) with respect to P7 (1988). (See GA03 in 3.3.1.1.)
- D09 If the product plays the role of service for the MA or MT interface:
 The manufacturer shall state the behavior of implementation-defined options.

3.3.1.2 Conformance Documentation

There are no test assertions for this subclause.

3.3.1.3 Conforming Implementation Options

- D10 The circumstances that define abnormal termination of a session shall be documented.
- D11 For *funct* of *ma_open*, *mt_open*:
 The features of a session that *funct* supplies as defaults shall be documented.
- D12 Documentation shall state whether the service shall accept any circumstances as causes of temporary delivery failure.
- D13 The maximum number of users that may be assigned to a single delivery queue shall be documented.
- D14 The maximum number of sessions that may exist simultaneously shall be documented.
- D15 The documentation shall state which qualifying object in the delivery queue (if there are several such objects) *ma_start_delivery* selects for delivery.
- D16 The documentation shall state which qualifying object in the input queue (if there are several such objects) *ma_start_transfer_in* selects for transfer in.
- D17 For *funct* of *ma_wait*, *mt_wait*:
 The documentation shall state to which and how many sessions *funct* notifies an arrival of an object when several sessions provide access to a delivery, retrieval or input queue.

3.3.2 Application Conformance

There are no test assertions for this subclause.

3.4 Test Assertions

There are no test assertions for this clause.

4. Test Assertions for Section Terminology and General Requirements

There are no test assertions for this section.

5. Test Assertions for Section Overview

5.1 Purpose

There are no test assertions for this clause.

5.2 Overview

- 001(C) If the MA interface is implemented:
Three persistent databases—a submission queue, a delivery queue, and a retrieval queue—are maintained by the service even when it is out of contact with the client.
- 002(C) If the MT interface is implemented:
Two persistent databases—an input queue, and an output queue—are maintained by the service even when it is out of contact with the client.

5.3 Object Management

There are no test assertions for this clause.

5.4 Features

- 003(A) The interfaces enable the client and service to negotiate the use of the functional units and packages of the MA and MT interfaces.
- 004(A) An implementation shall provide the following names to identify the features of the MA and MT interfaces.

Feature	Name
Basic Access FU	basic-access
Submission FU	submission
Delivery FU	delivery
Retrieval FU	retrieval
Basic Transfer FU	basic-transfer
Transfer In FU	transfer-in
Transfer Out FU	transfer-out
IM 84 Package	im-84
IM 88 Package	im-88
MH 84 Package	mh-84
MH 88 Package	mh-88
SM 88 Package	sm-88
Multiple Delivery	multiple-delivery

- 005(C) If the Multiple Delivery feature is not supported:
When `ma_start_delivery` delivers a message or report, it shall restrict itself to one recipient for each invocation.
- GA04 For *feature* of Submission FU, Delivery FU, Retrieval FU:
When the client has not requested *feature* by means of a parameter to the `ma_open` operation, then *feature* shall not be available to the client.
- GA05 For *feature* of Transfer In FU, Transfer Out FU:
When the client has not requested *feature* by means of a parameter to the `mt_open` operation, then *feature* shall not be available to the client.
- GA06 For *feature* of Submission FU, Delivery FU, Retrieval FU:
If the Basic Access FU is supported and *feature* is supported:
When the Basic Access FU is available and *feature* is unavailable and the client invokes a operation in *feature*, then the `feature_unavailable` error arises.
- 006(A) When the client supplies as an operation argument an object, one of whose subobjects is an instance of a class that is not in an available package, then the `no_such_class` error arises.
- 007(C) If the Basic Access FU is supported and the Basic Transfer FU is supported:
The Basic Access FU and the Basic Transfer FU are mutually exclusive for a particular session between the client and the service.
- 008(C) If the Delivery FU is supported and the Retrieval FU is supported:
The Delivery FU and the Retrieval FU are mutually exclusive for a particular session between the client and the service.

6. Test Assertions for Section Message Handling Interfaces

6.1 Datatypes

6.1.1 Boolean List

- 001(A) When a data value of the datatype `mh_boolean_list_type` is defined, then it is a list of Boolean values, used to indicate activated features.

6.1.2 Feature List

- 002(A) When a data value of the datatype `mh_feature_list_type` is defined, then it is a list of `om_object_id_type` values, used to indicate requested features.

6.1.3 Interval

- 003(A) When a data value of the datatype `mh_interval_type` is defined then, it is an integer in the range $0-2^{32}$.
- 004(A) When a data value of the datatype `mh_interval_type` is defined, then it is an integer that denotes an interval of time measured in milliseconds.

6.1.4 Object Count

- 005(A) When a data value of the datatype `mh_object_count_type` is defined, then it is an integer in the range $0-2^{32}$.

- 006(A) When a data value of the datatype `mh_object_count_type` is defined, then it is an integer that denotes a number of objects.
- GA07 For all MA and MT supported *func* in `ma_size`, `mt_size`:
When *func* is called and completes successfully, then it returns an integer of the datatype `mh_object_count_type`.

6.1.5 Sequence Number

- 007(A) When a data value of the datatype `mh_sequence_number_type` is defined, then it is an integer in the range $0-2^{32}$.
- 008(A) When a data value of the datatype `mh_sequence_number_type` is defined, then it is an integer that denotes the sequence number of an object in a retrieval queue.
- 009(A) When a data value of the datatype `mh_sequence_number_type` is defined, then it is a unique integer assigned in ascending order.
- 010(A) When `ma_start_retrieval` is called and completes successfully, then it returns an integer of the datatype `mh_sequence_number_type`.

6.2 Access Operations

- R01 When `ma_cancel_submission` is called and completes successfully, then this operation cancels a submitted message, with the deferred delivery option.
- R02 When `ma_close` is called and completes successfully, then this operation terminates an MA session between the client and the service. If the delivery or retrieval of a message or a report is in progress, the service first unsuccessfully finishes that delivery or retrieval.
- R03 When `ma_finish_delivery` is called and completes successfully, then this operation concludes the delivery in progress in a session. The client supplies delivery confirmations, as required, for users to which the object, a message, was delivered. It also indicates to which users the object, either a message or a report, is undeliverable.
- R04 When `ma_finish_retrieval` is called and completes successfully, then this operation concludes the retrieval in progress in a session. The client indicates whether the service is to remove the retrieved message, report from the retrieval queue, or leave it there.
- R05 When `ma_open` is called and completes successfully, then this operation establishes an MA session between the client and the service, and makes the Basic Access FU and the OM Package initially available in that session. The client may also specify the other features required for the session. The client specifies either its own name or the O/R address of a local user. If the client's own name is specified, then the session provides MTS access to a group of local users statically associated with the client name. If the O/R address of a local user is specified, then the session provides MTS access to the local user at the specified address.
- R06 When `ma_size` is called and completes successfully, then this operation determines the number of unreserved objects in the input queue to which a session provides access. Each object is a message or a report.
- R07 When `ma_start_delivery` is called and completes successfully, then this operation begins the delivery of a message or a report to one or more of the users associated with a session, by reserving an unreserved object in the delivery queue. If no messages or reports await delivery, the operation reports an exception.
- R08 When `ma_start_retrieval` is called and completes successfully, then this operation begins the retrieval of a message or a report, by operating an unreserved object in the retrieval

queue to which a session provides access. If no messages or reports await retrieval, the operation reports an exception.

- R09 When **ma_submit** is called and completes successfully, then this operation submits a communique, by adding it to the submission queue to which a session provides access. The operation first verifies the integrity of the communique.
- R10 When **ma_wait** is called and completes successfully, then this operation returns when a message or a report is available for delivery in the delivery queue, retrieval in the retrieval queue to which a session provides access, or when a period of time elapses, whichever occurs first.
- 001(A) The MA operations are grouped into four FUs (one basic and one each for submission, delivery, and retrieval) as indicated in the table below:

Basic Access	Submission	Delivery	Retrieval
Open	Submit	Size	Size
Close	Cancel Submission	Start Delivery	Start Retrieval
OM API		Finish Delivery	Finish Retrieval
		Wait	Wait

NOTES:

1 — The Delivery and Retrieval FUs are mutually exclusive.

2 — The OM API is defined in IEEE Std 1224-1993 {4}.

- 002(A) When the client has not requested Submission FU by means of a parameter to the **ma_open** function, then Submission FU will not be available to the client. (See GA04 in 5.4.)
- 003(C) If the Basic Access FU is supported and Submission FU is supported:
When the Basic Access FU is available and Submission FU is unavailable and the client invokes a function in Submission FU, then the **feature_unavailable** error arises. (See GA06 in 5.4.)
- 004(A) When the client has not requested Delivery FU by means of a parameter to the **ma_open** function, then Delivery FU will not be available to the client. (See GA04 in 5.4.)
- 005(C) If the Basic Access FU is supported and Delivery FU is supported:
When the Basic Access FU is available and Delivery FU is unavailable and the client invokes an operation in Delivery FU, then the **feature_unavailable** error arises. (See GA06 in 5.4.)
- 006(A) When the client has not requested Retrieval FU by means of a parameter to the **ma_open** function, then Retrieval FU will not be available to the client. (See GA04 in 5.4.)
- 007(C) If the Basic Access FU is supported and Retrieval FU is supported:
When the Basic Access FU is available and Retrieval FU is unavailable and the client invokes a function in Retrieval FU, then the **feature_unavailable** error arises. (See GA06 in 5.4.)
- 008(C) If the product plays the role of service for Basic Access FU:
The product shall implement every aspect of every operation in Basic Access FU. (See GA01 in 3.3.1.1.)
- 009(C) If the product plays the role of service for Submission FU:
The product shall implement every aspect of every operation in Submission FU. (See GA01 in 3.3.1.1.)

- 010(C) If the product plays the role of service for Delivery FU:
The product shall implement every aspect of every operation in Delivery FU. (See GA01 in 3.3.1.1.)
- 011(C) If the product plays the role of service for Retrieval FU:
The product shall implement every aspect of every operation in Retrieval FU. (See GA01 in 3.3.1.1.)

6.2.1 Cancel Submission

6.2.1.1 Synopsis

There are no test assertions for this subclause.

6.2.1.2 Description

- 001(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and completes successfully, then the deferred delivery of a submitted message is cancelled.

6.2.1.3 Results

- 002(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and does not complete successfully, then a return code indicating the reason is returned.
- 003(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)

6.2.1.4 Errors

- 004(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service does not offer the requested feature and no other error condition applies, then it will return either success or the `feature_unavailable` return code. (See GA10 in 6.4.)
- 005(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and the operation is aborted by an external force (e.g., a keystroke that is designated for this purpose at a user interface) and no other error condition applies, then it will return success or the `function_interrupted` return code. (See GA11 in 6.4.)
- 006(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or the `memory_insufficient` return code. (See GA12 in 6.4.)
- 007(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the `network_error` return code. (See GA13 in 6.4.)
NOTE — This should be tested when the service is implemented over a network.

- R01 If the Submission FU is supported:

When `ma_cancel_submission(mts_identifier)` is called, a deferred delivery of a message cannot be canceled because the *mts_identifier* does not identify a message previously submitted by the client, and no other error condition applies, then it will return either success or the `no_such_message` return code. [See 019(C) in 6.4.]

- 008(C) If the Submission FU is supported:
When `ma_cancel_submission(session,)` is called and the specified *session* does not exist and no other error condition applies, then it will return either success or the `no_such_session` return code. (See GA17 in 6.4.)
- 009(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and a purported syntax identifier is undefined and no other error condition applies, then it will return either success or the `no_such_syntax` return code. (See GA18 in 6.4.)
- 010(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and a purported datatype identifier is undefined and no other error condition applies, then it will return either success or the `no_such_type` return code. (See GA19 in 6.4.)
- 011(C) If the Submission FU is supported:
When `ma_cancel_submission(object,)` is called and a public *object* has been specified and no other error condition applies, then it will return either success or the `not_private` return code. (See GA20 in 6.4.)
- 012(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service encounters a permanent difficulty other than those denoted by other return codes, then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)
- 013(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 014(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and the service encounters a temporary difficulty other than those denoted by other return codes, then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- R02 If the Submission FU is supported:
When `ma_cancel_submission` is called and a deferred delivery of a message has progressed for transfer or delivery, or the service has provided the client with proof of submission and no other error condition applies, then it will return either success or the `too_late` return code. [See 023(C) in 6.4.]
- 015(D) If the Submission FU is supported:
When `ma_cancel_submission` is called and an undefined implementation limit prevents the addition to an object of another attribute value and no other error condition applies, then it will return either success or the `too_many_values` return code. (See GA30 in 6.4.)
- 016(C) If the Submission FU is supported:
When `ma_cancel_submission(object)` is called and an *object* is an instance of the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)

- 017(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and an attribute has, or would have, a value that violates the value length constraints in force and no other error condition applies, then it will return either success or the `wrong_value_length` return code. (See GA32 in 6.4.)
- 018(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and an attribute has, or would have, a value that violates a constraint of the syntax of the value and no other error condition applies, then it will return either success or the `wrong_value_makeup` return code. (see GA33 in 6.4.)
- 019(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and an attribute has, or would have, a value that violates the value number constraints in force and no other error condition applies, then it will return either success or the `wrong_value_number` return code. (See GA34 in 6.4.)
- 020(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and an attribute has, or would have, a value whose syntax is not permitted and no other error condition applies, then it will return either success or the `wrong_value_syntax` return code. (See GA35 in 6.4.)
- 021(C) If the Submission FU is supported:
When `ma_cancel_submission` is called and an object has, or would have, a value whose type is not permitted and no other error condition applies, then it will return either success or the `wrong_value_type` return code. (See GA36 in 6.4.)

6.2.2 Close

6.2.2.1 Synopsis

There are no test assertions for this subclause.

6.2.2.2 Description

- 001(C) If the Basic Access FU is supported:
When `ma_close(session)` is called and completes successfully, then the specified MA session is terminated, as well as destroying the workspace.
- 002(C) If the Basic Access FU is supported:
When `ma_close(session)` is called and completes successfully, then any delivery or retrieval of a message or report occurring in the specified session has been unsuccessfully finished before the MA session is terminated and the workspace is destroyed.

6.2.2.3 Results

- 003(C) If the Basic Access FU is supported:
When `ma_close` is called and does not complete successfully, then a return code indicating the reason is returned.
- 004(C) If the Basic Access FU is supported:
When `ma_close` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)

6.2.2.4 Errors

- 005(D) If the Basic Access FU is supported:
When `ma_close` is called and the operation is aborted by an external force (e.g., a keystroke designated for this purpose at a user interface) and no other error condition applies, then it will return success or the `function_interrupted` return code. (See GA11 in 6.4.)
- 006(C) If the Basic Access FU is supported:
When `ma_close` is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or the `memory_insufficient` return code. (See GA12 in 6.4.)
- 007(D) If the Basic Access FU is supported:
When `ma_close` is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the `network_error` return code. (See GA13 in 6.4.)
- 008(C) If the Basic Access FU is supported:
When `ma_close(session)` is called and the specified session does not exist and no other error condition applies, then it will return either success or the `no_such_session` return code. (See GA17 in 6.4.)
- 009(C) If the Basic Access FU is supported:
When `ma_close(session)` is called and a public object has been specified and no other error condition applies, then it will return either success or the `not_private` return code. (See GA20 in 6.4.)
- 010(D) If the Basic Access FU is supported:
When `ma_close` is called and the service encounters a permanent difficulty other than those denoted by other return codes, then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)
- 011(D) If the Basic Access FU is supported:
When `ma_close` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 012(D) If the Basic Access FU is supported:
When `ma_close` is called and the service encounters a temporary difficulty other than those denoted by other return codes, then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- 013(C) If the Basic Access FU is supported:
When `ma_close(session)` is called and an object is the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)

6.2.3 Finish Delivery

6.2.3.1 Synopsis

There are no test assertions for this subclause.

6.2.3.2 Description

- 01(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and completes successfully, then the delivery in progress in a session is concluded.
- 002(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and completes successfully and no delivery or nondelivery confirmations have been specified, then the service assumes that the message has been delivered to the recipients not covered by a confirmation or nondelivery.
- D18 If the Delivery FU is supported:
When `ma_finish_delivery` is called, then whether the service will accept any circumstances as causes of temporary failure is service documented.
- 003(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service does not support temporary failures, then it treats them as permanent.
- 004(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the object is a message, then the service issues Delivery Reports for the users to which it has been delivered.
- 005(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the object is a message, then the service issues Nondelivery Reports for the users to which it is permanently undeliverable.
- 006(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the object is a message, then the service returns the message to the delivery queue for the users to which it is temporarily undeliverable and marks the object unreserved.
- 007(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the object is a report and the user to which it was to be delivered is temporarily undelivered, then the service returns the report to the delivery queue.

6.2.3.3 Arguments In

- 008(C) If the Delivery FU is supported:
When `ma_finish_delivery(delivery_confirmations,)` is called, then *delivery_confirmations* is one or more delivery confirmations, as required, for users to which the object, a message, was delivered.
- 009(C) If the Delivery FU is supported:
When `ma_finish_delivery(,non_delivery_reports)` is called, then *non_delivery_reports* indicates one or more users to which the object, a message, cannot be delivered, or the single user to which the object, a report, cannot be delivered.

6.2.3.4 Results

- 010(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and does not complete successfully, then a return code indicating the reason is returned.
- 011(C) If the Delivery FU is supported:

When `ma_finish_delivery` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)

6.2.3.5 Errors

- 012(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service does not offer the requested feature and no other error condition applies, then it will return either success or the `feature_unavailable` return code. (See GA10 in 6.4.)
- 013(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the operation is aborted by an external force (e.g., a keystroke at a user interface that is designated for this purpose) and no other error condition applies, then it will return success or the `function_interrupted` return code. (See GA11 in 6.4.)
- 014(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or the `memory_insufficient` return code. (See GA12 in 6.4.)
- 015(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the `network_error` return code. (See GA13 in 6.4.)
- 016(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and a purported object is nonexistent or the handle of the purported object is invalid and no other error condition applies, then it will return either success or the `no_such_object` return code. (See GA16 in 6.4.)
- 017(C) If the Delivery FU is supported:
When `ma_finish_delivery(session,,)` is called and the specified session does not exist and no other error condition applies, then it will return either success or the `no_such_session` return code. (See GA17 in 6.4.)
- 018(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and a purported syntax identifier is undefined and no other error condition applies, then it will return either success or the `no_such_syntax` return code. (See GA18 in 6.4.)
- 019(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and a purported type identifier is undefined and no other error condition applies, then it will return either success or the `no_such_type` return code. (See GA19 in 6.4.)
- 020(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and a public object has been specified and no other error condition applies, then it will return either success or the `not_private` return code. (See GA20 in 6.4.)
- 021(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service encounters a permanent difficulty other than those denoted by other return codes, then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)

- 022(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and a delivery or retrieval is not in progress in the session and no other error condition applies, then it will return either success or the `session_not_busy` return code. (See GA25 in 6.4.)
- 023(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 024(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and the service encounters a temporary difficulty other than those denoted by other return codes then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- 025(D) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an undefined implementation limit prevents the addition to an object of another attribute value and no other error condition applies, then it will return either success or the `too_many_values` return code. (See GA30 in 6.4.)
- 026(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an object is an instance of the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)
- 027(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an attribute has, or would have, a value that violates the value length constraints in force and no other error condition applies, then it will return either success or the `wrong_value_length` return code. (See GA32 in 6.4.)
- 028(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an attribute has, or would have, a value that violates a constraint of the syntax of the value, and no other error condition applies, then it will return either success or the `wrong_value_makeup` return code. (See GA33 in 6.4.)
- 029(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an attribute has, or would have, a value that violates the value number constraints in force and no other error condition applies, then it will return either success or the `wrong_value_number` return code. (See GA34 in 6.4.)
- 030(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an attribute has, or would have, a value whose syntax is not permitted and no other error condition applies, then it will return either success or the `wrong_value_syntax` return code. (See GA35 in 6.4.)
- 031(C) If the Delivery FU is supported:
When `ma_finish_delivery` is called and an object has, or would have, a value whose type is not permitted and no other error condition applies, then it will return either success or the `wrong_value_type` return code (See GA36 in 6.4.)

6.2.4 Finish Retrieval

6.2.4.1 Synopsis

There are no test assertions for this subclause.

6.2.4.2 Description

- 001(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and completes successfully, then it concludes the retrieval in progress in the specified session.
- 002(C) If the Retrieval FU is supported:
When `ma_finish_retrieval(session,remove)` is called and *remove* is specified to be true and the operation completes successfully, then the service removes the message or report in progress from the retrieval queue.
- 003(C) If the Retrieval FU is supported:
When `ma_finish_retrieval(session,remove)` is called and *remove* is specified to be false and completes successfully, then the message or report in progress is left in the retrieval queue.
- 004(C) If the Retrieval FU is supported:
When `ma_finish_retrieval(session,remove)` is called and *remove* is specified to be false and completes successfully, then the message or report in progress is marked unreserved and made inaccessible by the service.

6.2.4.3 Results

- 005(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and does not complete successfully, then a return code indicating the reason is returned.
- 006(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)

6.2.4.4 Errors

- 007(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the service does not offer the requested feature and no other error condition applies, then it will return either success or the `feature_unavailable` return code. (See GA10 in 6.4.)
- 008(D) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the operation is aborted by an external force (e.g., a keystroke at user interface that is designated for this purpose) and no other error condition applies, then it will return success or the `function_interrupted` return code. (See GA11 in 6.4.)
- 009(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or the `memory_insufficient` return code. (See GA12 in 6.4.)
- 010(D) If the Retrieval FU is supported:

When `ma_finish_retrieval` is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the `network_error` return code. (See GA13 in 6.4.)

- 011(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and a purported object is nonexistent or the handle of the purported object is invalid and no other error condition applies, then it will return either success or the `no_such_object` return code. (See GA16 in 6.4.)
- 012(C) If the Retrieval FU is supported:
When `ma_finish_retrieval(session,)` is called and the specified session does not exist and no other error condition applies, then it will return either success or the `no_such_session` return code. (See GA17 in 6.4.)
- 013(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and a public object has been specified and no other error condition applies, then it will return either success or the `not_private` return code. (See GA20 in 6.4.)
- 014(D) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the service encounters a permanent difficulty other than those denoted by other return codes, then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)
- 015(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and a delivery or retrieval is not in progress in the session and no other error condition applies, then it will return either success or the `session_not_busy` return code. (See GA25 in 6.4.)
- 016(D) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 017(D) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and the service encounters a temporary difficulty other than those denoted by other return codes, then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- 018(C) If the Retrieval FU is supported:
When `ma_finish_retrieval` is called and an object is an instance of the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)

6.2.5 Open

6.2.5.1 Synopsis

There are no test assertions for this subclause.

6.2.5.2 Description

- 001(C) If the Basic Access FU is supported:
When `ma_open` is called and completes successfully, then an MA session between the client and the service is established.

- 002(C) If the Basic Access FU is supported:
When **ma_open** is called and completes successfully, then the Basic Access FU and the OM Package is made initially available for the specified session.
- D19 If the Basic Access FU is supported:
When **ma_open** is called, then the maximum number of users in a group is documented.
- 003(C) If the Basic Access FU is supported:
When **ma_open**(,,,*,*workspace*) is called and completes successfully, then the workspace **workspace* is created.
- D20 If the Basic Access FU is supported:
When **ma_open** is called, then the maximum number of sessions that may exist simultaneously is documented.
- 004(C) If the Basic Access FU is supported:
When **ma_open** generates an instance of MH Class **Session** and the subject domain is an ADMD, then **ma_open** will set the **ADMD-Name** attribute to the name of the subject domain relative to the country that the **Country-Name** attribute denotes.
Testing Requirements: Test that the value of the **ADMD-Name** attribute is taken from the administration data held by the local MTA. (See GA395 in 8.2.42.)
- 005(C) If the Basic Access FU is supported:
When **ma_open** generates an instance of MH Class **Session** and the subject domain is a PRMD, then **ma_open** will set the **ADMD-Name** attribute to the name of the ADMD to which the subject domain is attached, relative to the country that the **Country-Name** attribute denotes.
Testing Requirements: Test that the value of the **ADMD-Name** attribute is taken from the administration data held by the local MTA. (See GA396 in 8.2.42.)
- 006(C) If the Basic Access FU is supported:
When **ma_open** generates an instance of MH Class **Session**, then **ma_open** will set the **Country-Name** attribute to the name of the country of the ADMD that the **ADMD-Name** attribute denotes.
Testing Requirements: Test that the value of the **Country-Name** attribute is taken from the administration data held by the local MTA. (See GA397 in 8.2.42.)
- 007(C) If the Basic Access FU is supported:
When **ma_open** generates an instance of MH Class **Session** and sets the **Country-Name** attribute, then **ma_open** will set the **Country-Name** attribute to a value that is either the number that CCITT Recommendation X.121:1984 {2} assigns to the country, or the character pair that ISO 3166 :1988 {1} assigns to it. (See GA398 in 8.2.42.)
- 008(C) If the Basic Access FU is supported:
When **ma_open** generates an instance of MH Class **Session** and the subject domain is a PRMD, then **ma_open** will set the **PRMD-Identifier** attribute to identify the PRMD relative to the ADMD that the **ADMD-Name** attribute denotes.
Testing Requirements: Test that the value of the **PRMD-Identifier** attribute is taken from the administration data held by the local MTA. (See GA399 in 8.2.42.)

6.2.5.3 Arguments In

6.2.5.4 Results

- 009(C) If the Basic Access FU is supported:
When **ma_open** is called and does not complete successfully, then a return code indicating the reason is returned.
- 010(C) If the Basic Access FU is supported:
When **ma_open** is called and completes successfully, and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)
- 011(C) If the Basic Access FU is supported:
When **ma_open** is called and completes successfully, then *activated*, an ordered sequence of Boolean values with the same number of elements as the *feature_list*, is returned.
- 012(C) If the Basic Access FU is supported:
When **ma_open** is called, then a true value in the *activated* result indicates that the corresponding feature is now a part of the interface.
- 013(C) If the Basic Access FU is supported:
When **ma_open** is called, then a **false** value in the *activated* result indicates that the corresponding feature is not available.
- 014(C) If the Basic Access FU is supported:
When **ma_open**(,,,**session*,) is called and completes successfully, then **session* is the established MA session between the client and service; it is an instance of the **session** class.
- 015(C) If the Basic Access FU is supported:
When **ma_open**(,,,**session*,) is called and completes successfully, then the service prevents **session* from being modified by the client.
- 016(D) If the Basic Access FU is supported:
When **ma_open**(,,,**session*,) is called and does not complete successfully, then **session* is not present.
- 017(C) If the Basic Access FU is supported:
When **ma_open**(,,,**workspace*) is called and completes successfully, then **workspace*, which will contain all objects returned as a result of the operations invoked in the session, is returned.

6.2.5.5 Errors

- 018(C) If the Basic Access FU is supported:
When **ma_open**(,*feature_list*,) is called and the requested feature conflicts with an existing feature of the session and no other error condition applies, then it will return either success or the **feature_conflicts** return code. (See GA09 in 6.4.)
- 019(C) If the Basic Access FU is supported:
When **ma_open** is called and the service does not offer the requested feature and no other error condition applies, then it will return either success or the **feature_unavailable** return code. (See GA10 in 6.4.)
- 020(D) If the Basic Access FU is supported:
When **ma_open** is called and the operation is aborted by an external force (e.g., a keystroke at a user interface that is designated for this purpose) and no other

- error condition applies, then it will return success or the `function_interrupted` return code. (See GA11 in 6.4.)
- 021(C) If the Basic Access FU is supported:
When `ma_open` is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or the `memory_insufficient` return code. (See GA12 in 6.4.)
- 022(D) If the Basic Access FU is supported:
When `ma_open` is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the `network_error` return code. (See GA13 in 6.4.)
- 023(C) If the Basic Access FU is supported:
When `ma_open(client_name,,)` is called and the service does not recognize the specified client name and no other error condition applies, then it will return either success or the `no_such_client` return code. (See GA15 in 6.4.)
- 024(C) If the Basic Access FU is supported:
When `ma_open` is called and a purported syntax identifier is undefined and no other error condition applies, then it will return either success or the `no_such_syntax` return code. (See GA18 in 6.4.)
- 025(C) If the Basic Access FU is supported:
When `ma_open` is called and a purported type identifier is undefined and no other error condition applies, then it will return either success or the `no_such_type` return code. (See GA19 in 6.4.)
- R01 If the Basic Access FU is supported:
When `ma_open(user_address,,)` is called and no other error condition applies, then it will return either success or the `no_such_user` return code. [See 020(C) in 6.4.]
- 026(D) If the Basic Access FU is supported:
When `ma_open` is called and the service encounters a permanent difficulty other than those denoted by other return codes, then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)
- 027(D) If the Basic Access FU is supported:
When `ma_open` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 028(D) If the Basic Access FU is supported:
When `ma_open` is called and the service encounters a temporary difficulty other than those denoted by other return codes, then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- 029(C) If the Basic Access FU is supported:
When `ma_open` is called and the number of sessions is equal to an implementation defined limit and no other error condition applies, then it will return either success or the `too_many_sessions` return code. (See GA29 in 6.4.)
- 030(D) If the Basic Access FU is supported:
When `ma_open` is called and an undefined implementation limit prevents the addition to an object of another attribute value and no other error condition

applies, then it will return either success or the `too_many_values` return code. (See GA30 in 6.4.)

- 031(C) If the Basic Access FU is supported:
When `ma_open` is called and an object is an instance of the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)
- 032(C) If the Basic Access FU is supported:
When `ma_open` is called and an attribute has, or would have, a value that violates the value length constraints in force and no other error condition applies, then it will return either success or the `wrong_value_length` return code. (See GA32 in 6.4.)
- 033(C) If the Basic Access FU is supported:
When `ma_open` is called and an attribute has, or would have, a value that violates a constraint of the syntax of the value and no other error condition applies, then it will return either success or the `wrong_value_makeup` return code. (See GA33 in 6.4.)
- 034(C) If the Basic Access FU is supported:
When `ma_open` is called and an attribute has, or would have, a value that violates the value number constraints in force and no other error condition applies, then it will return either success or the `wrong_value_number` return code. (See GA34 in 6.4.)
- 035(C) If the Basic Access FU is supported:
When `ma_open` is called and an attribute has, or would have, a value whose syntax is not permitted and no other error condition applies, then it will return either success or the `wrong_value_syntax` return code. (See GA35 in 6.4.)
- 036(C) If the Basic Access FU is supported:
When `ma_open` is called and an object has, or would have, a value whose type is not permitted and no other error condition applies, then it will return either success or the `wrong_value_type` return code. (See GA36 in 6.4.)

6.2.6 Size

6.2.6.1 Synopsis

There are no test assertions for this subclause.

6.2.6.2 Description

- 001(C) If the Delivery FU is supported:
When `ma_size(*number)` is called and completes successfully, then it returns the number of messages and reports in the delivery queue of the specified session.
- 002(C) If the Retrieval FU is supported:
When `ma_size` is called and completes successfully, then it returns the number of messages and reports in the retrieval queue of the specified session.

6.2.6.3 Results

- 003(C) If the Delivery or Retrieval FU is supported:
When `ma_size` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)

- 004(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and does not complete successfully, then a return code indicating the reason is returned.
- 005(C) If the Delivery FU is supported:
When *ma_size* is called and completes successfully, then *number* indicating the number of unreserved objects in the delivery queue will be returned.
- 006(C) If the Retrieval FU is supported:
When *ma_size* is called and completes successfully, then *number* indicating the number of unreserved objects in the retrieval queue will be returned.
- 007(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and completes successfully and *number* returned exceeds $2^{16} - 1$, then $2^{16} - 1$ is returned.
- 008(D) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and does not complete successfully, then *number* will not be present.
- 009(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and completes successfully, then it returns an integer of the type *mh_object_count_type*. (See GA07 in 6.1.4.)

6.2.6.4 Errors

- 010(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and the service does not offer the requested feature and no other error condition applies, then it will return either success or the *feature_unavailable* return code. (See GA10 in 6.4.)
- 011(D) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and the operation is aborted by an external force (e.g., a keystroke at a user interface that is designated for this purpose) and no other error condition applies, then it will return success or the *function_interrupted* return code. (See GA11 in 6.4.)
- 012(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and the service cannot allocate the main memory it needs to complete the operation and no other error condition applies, then it will return either success or *memory_insufficient* return code. (See GA12 in 6.4.)
- 013(D) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and the service could not successfully employ the network upon which its implementation depends and no other error condition applies, then it will return either success or the *network_error* return code. (See GA13 in 6.4.)
- 014(C) If the Delivery or Retrieval FU is supported:
When *ma_size(session,)* is called and the specified session does not exist and no other error condition applies, then it will return either success or the *no_such_session* return code. (See GA17 in 6.4.)
- 015(C) If the Delivery or Retrieval FU is supported:
When *ma_size* is called and a public object has been specified and no other error condition applies, then it will return either success or the *not_private* return code. (See GA20 in 6.4.)

- 016(D) If the Delivery or Retrieval FU is supported:
When `ma_size` is called and the service encounters a permanent difficulty other than those denoted by other return codes then it will return either success or the `permanent_error` return code. (See GA22 in 6.4.)
- 017(D) If the Delivery or Retrieval FU is supported:
When `ma_size` is called and the service could not successfully employ the operating system upon which its implementation depends and no other error condition applies, then it will return either success or the `system_error` return code. (See GA27 in 6.4.)
- 018(D) If the Delivery or Retrieval FU is supported:
When `ma_size` is called and the service encounters a temporary difficulty other than those denoted by other return codes, then it returns the `temporary_error` return code. (See GA28 in 6.4.)
- 019(C) If the Delivery or Retrieval FU is supported:
When `ma_size` is called and an object is an instance of the wrong class and no other error condition applies, then it will return either success or the `wrong_class` return code. (See GA31 in 6.4.)

6.2.7 Start Delivery

6.2.7.1 Synopsis

There are no test assertions for this subclause.

6.2.7.2 Description

- 001(C) If the Delivery FU is supported:
When `ma_start_delivery` is called, then it begins the delivery of a message or a report to one or more of the users associated with a session.
- 002(C) If the Delivery FU is supported:
When `ma_start_delivery` is called and completes successfully, then an unreserved object in the delivery queue is reserved.
- 003(C) If the Delivery FU is supported:
When `ma_start_delivery` is called and no objects await delivery, then an exception is returned.
- 004(C) If the Delivery FU is supported:
When `ma_start_delivery` is called, then the delivery of an object cannot simultaneously be in progress in any other session.
- 005(C) If the Delivery FU is supported:
If the multiple-delivery feature for a session is supported:
When `ma_start_delivery` is called and an object is addressed to several users, then the service begins delivery in one operation invocation.
Otherwise:
When `ma_start_delivery` is called and an object is addressed to several users, then the service begins delivery in one operation invocation per user.
- D21 If the Delivery FU is supported:
When `ma_start_delivery` is called and there are several objects in the delivery queue, then the object qualifying for selection by the service is documented.

6.2.7.3 Results

- 006(C) If the Delivery FU is supported:
When `ma_start_delivery` is called and does not complete successfully, then a return code indicating the reason is returned.
- 007(C) If the Delivery FU is supported:
When `ma_start_delivery` is called and completes successfully and no error condition applies, then it will return the success return code. (See GA26 in 6.4.)
- 008(C) If the Delivery FU is supported:
When `ma_start_delivery(*object)` is called and completes successfully, then **object* is an instance of the **Delivered-Message** class with an envelope for each one or more of the users associated with the session whose delivery has started.
- 009(C) If the Delivery FU is supported:
When `ma_start_delivery(*object)` is called and completes successfully, then **object* is an instance of the **Delivered-Report** class whose delivery has started.
- 010(C) If the Delivery FU is supported:
When `ma_start_delivery(*object)` is called and completes successfully, then **object* is present.
- 011(D) If the Delivery FU is supported:
When `ma_start_delivery(*object)` is called and does not complete successfully, then **object* is not present.
- 012(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Bilaterally-Defined-Body-Part**, then it will set the **Bilateral-Data** attribute to the binary data that the body part is intended to convey in the message.
Testing Requirements: The value of the **Bilateral-Data** attribute will be determined from the conveyed message. (See GA37 in 7.3.1.)
- 013(D) If the IM 88 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Externally-Defined-Body-Part**, then it will set the **External-Data** attribute to the information object that the body part is intended to convey in the message.
Testing Requirements: The value of the **External-Data** attribute will be determined from the conveyed message. (See GA39 in 7.3.3.)
- 014(D) If the IM 88 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Externally-Defined-Body-Part** and the body part contains external parameters, then `ma_start_delivery` will set the **External-Parameters** attribute to the information object that characterizes the value of the External Data attribute.
Testing Requirements: The value of the **External-Parameters** attribute will be determined from the conveyed message. (See GA41 in 7.3.3.)
- 015(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **G3-Fax-Body-Part** and the body part contains the NBPs of the images, then `ma_start_delivery` will set the **G3-Fax-NBPs** attribute to the NBPs of the images.
Testing Requirements: The value of the **G3-Fax-NBPs** attribute will be determined from the conveyed message. (See GA43 in 7.3.4.)

- 016(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **G3-Fax-Body-Part**, then **ma_start_delivery** will set the **Images** attribute to the G3 facsimile image(s) in the body part.
 Testing Requirements: The values of the **Images** attribute will be determined from the conveyed message. (See GA45 in 7.3.4.)
- 017(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **G4-Class-1-Body-Part**, then **ma_start_delivery** will set the **G4-Class-1-Documents** attribute to the final form document(s) of the body part.
 Testing Requirements: The values of the **G4-Class-1-Documents** attribute will be determined from the conveyed message. (See GA47 in 7.3.5.)
- 018(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **General-Text-Body-Part**, then **ma_start_delivery** will set the **Char-Set-Registration** attribute to the list of the character set registrations that are or may be present in the data component of the body part.
 Testing Requirements: The values of the **Char-Set-Registration** attribute will be determined from the conveyed message. (See GA49 in 7.3.6.)
- 019(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **General-Text-Body-Part**, then **ma_start_delivery** will set the **General-Text-Data** attribute to all of the text of the body part.
 Testing Requirements: The value of the **General-Text-Data** attribute will be determined from the conveyed message. (See GA51 in 7.3.6.)
- 020(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **IA5-Text-Body-Part**, then **ma_start_delivery** will set the **Repertoire** attribute to identify the character set to which the text of the body part is constrained.
 Testing Requirements: The value of the **Repertoire** attribute will be determined from the conveyed message. (See GA53 in 7.3.7.)
- 021(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **IA5-Text-Body-Part**, then **ma_start_delivery** will set the **Text** attribute to all of the text of the body part.
 Testing Requirements: The value of the **Text** attribute will be determined from the conveyed message. (See GA55 in 7.3.7.)
- 022(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and the authorizing users are other than the originator of the IPM alone, then **ma_start_delivery** will set the **Authorizing-Users** attribute to the O/R descriptors for the authorizing users of the interpersonal message.
 Testing Requirements: The values of the **Authorizing-Users** attribute will be determined from the conveyed message. (See GA57 in 7.3.8.)

- 023(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the authorizing user is the IPM's originator alone, then `ma_start_delivery` will not set the **Authorizing-Users** attribute. (See GA58 in 7.3.8.)
- 024(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message**, then `ma_start_delivery` will set the **Auto-forwarded** attribute to indicate whether the IPM is the result of automatic (not manual) forwarding.
Testing Requirements: The values of the **Auto-forwarded** attribute will be determined from the conveyed message. (See GA60 in 7.3.8.)
- 025(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the subject IPM instance is not intended for a primary or copy recipient, then `ma_start_delivery` will set the **Blind-Copy-Recipients** attribute to the recipient specifiers for blind copy recipients of the inter-personal message.
Testing Requirements: The values of the **Blind-Copy-Recipients** attribute will be determined from the conveyed message. (See GA62 in 7.3.8.)
- 026(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the subject IPM instance is intended for a primary or copy recipient, then `ma_start_delivery` will not set the **Blind-Copy-Recipients** attribute. (See GA63 in 7.3.8.)
- 027(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message**, then `ma_start_delivery` will set the **Body** attribute to the information objects that the IPM is intended to convey from its originator to its intended recipients, one information object per attribute value.
Testing Requirements: The values of the **Body** attribute will be determined from the conveyed message. (See GA65 in 7.3.8.)
- 028(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and there are copy recipients for the IPM, then `ma_start_delivery` will set the **Copy-Recipients** attribute to the recipient specifiers for the copy recipients of the IPM.
Testing Requirements: The values of the **Copy-Recipients** attribute will be determined from the conveyed message. (See GA67 in 7.3.8.)
- 029(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and there are no copy recipients for the IPM, then `ma_start_delivery` will not set the **Copy-Recipients** attribute. (See GA68 in 7.3.8.)

- 030(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted for an expiry time to be set, then **ma_start_delivery** will set the **Expiry-Time** attribute to the date and time at which the authorizing users consider the IPM to lose its validity.
 Testing Requirements: The value of the **Expiry-Time** attribute will be determined from the conveyed message. (See GA70 in 7.3.8.)
- 031(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message**, then **ma_start_delivery** will set the **Importance** attribute to identify the importance that the authorizing users attach to the IPM.
 Testing Requirements: The value of the **Importance** attribute will be determined from the conveyed message. (See GA72 in 7.3.8.)
- 032(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message**, then **ma_start_delivery** will set the **Incomplete-Copy** attribute to indicate whether one or more values of the **Body** attribute, or all values of another attribute, are absent from the present instance of the IPM.
 Testing Requirements: The value of the **Incomplete-Copy** attribute will be determined from the conveyed message. (See GA74 in 7.3.8.)
- 033(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and the originator has opted to identify the languages used in the composition of the **Body** and **Subject** attributes, then **ma_start_delivery** will set the **Languages** attribute to identify these languages.
 Testing Requirements: The values of the **Languages** attribute will be determined from the conveyed message. (See GA76 in 7.3.8.)
- 034(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted to indicate IPMs obsoleted by the present IPM, then **ma_start_delivery** will set the **Obsoleted-IPMs** attribute to the IPM identifiers of the IPMs that the authorizing users of the present IPM consider it to obsolete.
 Testing Requirements: The values of the **Obsoleted-IPMs** attribute will be determined from the conveyed message. (See GA78 in 7.3.8.)
- 035(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted to provide the O/R descriptor of the originator, then **ma_start_delivery** will set the **Originator** attribute to this O/R descriptor.
 Testing Requirements: The value of the **Originator** attribute will be determined from the conveyed message. (See GA80 in 7.3.8.)
- 036(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Message** and there are primary recipients for the IPM, then **ma_start_delivery**

will set the **Primary-Recipients** attribute to the recipient specifiers for the primary recipients of the IPM.

Testing Requirements: The values of the **Primary-Recipients** attribute will be determined from the conveyed message. (See GA82 in 7.3.8.)

- 037(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and there are no primary recipients for the IPM, then `ma_start_delivery` will not set the **Primary-Recipients** attribute. (See GA83 in 7.3.8.)
- 038(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted to provide an indication of related IPMs, then `ma_start_delivery` will set the **Related-IPMs** attribute to the IPM identifiers of the IPMs that the authorizing users of the present IPM consider related to it.
Testing Requirements: The values of the **Related-IPMs** attribute will be determined from the conveyed message. (See GA85 in 7.3.8.)
- 039(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the IPM is a reply, then `ma_start_delivery` will set the **Replied-to-IPM** attribute to the IPM identifier of the IPM to which the present IPM is a reply.
Testing Requirements: The value of the **Replied-to-IPM** attribute will be determined from the conveyed message. (See GA87 in 7.3.8.)
- 040(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the IPM is not a reply, then `ma_start_delivery` will not set the **Replied-to-IPM** attribute. (See GA88 in 7.3.8.)
- 041(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the desired reply recipients are other than the originator alone, then `ma_start_delivery` will set the **Reply-Recipients** attribute to the O/R descriptors for the users whom the authorizing users request (but do not demand) are among the recipients of any replies to the IPM.
Testing Requirements: The values of the **Reply-Recipients** attribute will be determined from the conveyed message. (See GA90 in 7.3.8.)
- 042(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the desired reply recipient is the originator alone, then `ma_start_delivery` will not set the **Reply-Recipients** attribute. (See GA91 in 7.3.8.)
- 043(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted to provide a reply time, then

`ma_start_delivery` will set the **Reply-Time** attribute to the date and time by which the authorizing users request (but do not demand) that any replies to the IPM are originated.

Testing Requirements: The value of the **Reply-Time** attribute will be determined from the conveyed message. (See GA93 in 7.3.8.)

044(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message**, then `ma_start_delivery` will set the **Sensitivity** attribute to indicate how sensitive the authorizing users consider the IPM to be.

Testing Requirements: The value of the **Sensitivity** attribute will be determined from the conveyed message. (See GA95 in 7.3.8.)

045(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message** and the originator of the IPM opted to provide a textual description of the subject matter of the IPM, then `ma_start_delivery` will set the **Subject** attribute to the description provided.

Testing Requirements: The value of the **Subject** attribute will be determined from the conveyed message. (See GA97 in 7.3.8.)

046(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Message**, then `ma_start_delivery` will set the **This-IPM** attribute to the IPM identifier assigned to the IPM.

Testing Requirements: The value of the **This-IPM** attribute will be determined from the conveyed message. (See GA99 in 7.3.8.)

047(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Notification** and the subject IPM was converted for delivery to the originator of the IPN, then `ma_start_delivery` will set the **Conversion-EITs** attribute to the EITs of the subject IPM at the time of its delivery to that user.

Testing Requirements: The value of the **Conversion-EITs** attribute will be determined from the conveyed message. (See GA101 in 7.3.9.)

048(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Notification** and the subject IPM was not converted for delivery to the originator of the IPN, then `ma_start_delivery` will not set the **Conversion-EITs** attribute. (See Rna in 7.3.9.)

049(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When `ma_start_delivery` generates an instance of IM Class **Interpersonal-Notification** and the subject IPM was delivered to an alternate (not an intended) recipient, then `ma_start_delivery` will set the **IPM-Intended-Recipient** attribute to the O/R descriptor that is the value of the **Recipient** attribute of the subject recipient specifier for the intended recipient.

Testing Requirements: The value of the **IPM-Intended-Recipient** attribute will be determined from the conveyed message. (See GA104 in 7.3.9.)

- 050(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Notification** and the subject IPM was delivered to the intended recipient, then **ma_start_delivery** will not set the **IPM-Intended-Recipient** attribute. (See GA105 in 7.3.9.)
- 051(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Notification** and the user that originates the IPN opts to identify the IPN originator, then **ma_start_delivery** will set the **IPN-Originator** attribute to the O/R descriptor that is the value of the **Recipient** attribute of the subject recipient specifier for the originator of the IPN:
 Testing Requirements: The value of the **IPN-Originator** attribute will be determined from the conveyed message. (See GA107 in 7.3.9.)
- 052(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Interpersonal-Notification**, then **ma_start_delivery** will set the **Subject-IPM** attribute to the IPM identifier of the subject IPM.
 Testing Requirements: The value of the **Subject-IPM** attribute will be determined from the conveyed message. (See GA109 in 7.3.9.)
- 053(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **IPM-Identifier** and the client that originated the IPM opts to provide an O/R name of the originator of the IPM, then **ma_start_delivery** will set the **User** attribute to the O/R provided.
 Testing Requirements: The value of the **User** attribute will be determined from the conveyed message. (See GA111 in 7.3.10.)
- 054(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **IPM-Identifier**, then **ma_start_delivery** will set the **User-Relative-Identifier** attribute to a value that uniquely identifies the IPM, unambiguously distinguishing it from all other IPMs originated by the originator of the IPM.
 Testing Requirements: The value of the **User-Relative-Identifier** attribute will be determined from the conveyed message. (See GA113 in 7.3.10.)
- 055(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **ISO-6937-Text-Body-Part**, then **ma_start_delivery** will set the **Repertoire** attribute to identify the character set to which the text is constrained.
 Testing Requirements: The value of the **Repertoire** attribute will be determined from the conveyed message. (See GA115 in 7.3.11.)
- 056(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **ISO-6937-Text-Body-Part**, then **ma_start_delivery** will set the **Text** attribute to the sequence of lines that constitutes the text.
 Testing Requirements: The value of the **Text** attribute will be determined from the conveyed message. (See GA117 in 7.3.11.)

- 057(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Message-Body-Part** and the user who originated the message of which the body part is a component opted to provide the delivery envelope that accompanied the forwarded IPM when it was delivered to the originator of the forwarding IPM, then **ma_start_delivery** will set the **Envelope** attribute to the envelope provided. Testing Requirements: The value of the **Envelope** attribute will be determined from the conveyed message. (See GA119 in 7.3.12.)
- 058(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Message-Body-Part** and it sets the **Envelope** attribute, then the **MTS-Identifier** of the delivery envelope shall be absent. (See GA120 in 7.3.12.)
- 059(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Message-Body-Part** and it sets the **Envelope** attribute, then the **Delivery-Time** from the delivery envelope shall not be absent when all other attributes from the delivery envelope are absent. (See GA121 in 7.3.12.)
- 060(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Message-Body-Part**, then **ma_start_delivery** will set the **IPM** attribute to the forwarded IPM. Testing Requirements: The value of the **IPM** attribute will be determined from the conveyed message. (See GA123 in 7.3.12.)
- 061(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Mixed-mode-Body-Part** and the IPM includes a mixed-mode document, then **ma_start_delivery** will set the **Mixed-mode-Document** attribute to the final-form document, with each value being a protocol element that contributes to a description of the document's layout structure. Testing Requirements: The values of the **Mixed-mode-Document** attribute will be determined from the conveyed message. (See GA125 in 7.3.13.)
- 062(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Nationally-Defined-Body-Part**, then **ma_start_delivery** will set the **National-Data** attribute to the binary data of the body part, which follows the Basic Encoding Rules. Testing Requirements: The value of the **National-Data** attribute will be determined from the conveyed message. (See GA127 in 7.3.14.)
- 063(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Non-receipt-Notification** and the **Non-receipt-Reason** attribute has the value **ipm-auto-forwarded**, then **ma_start_delivery** will set the **Auto-forwarded-Comment** attribute to the information presupplied by the originator of the nonreceipt notification.

- Testing Requirements: The value of the **Auto-forwarded-Comment** attribute will be determined from the conveyed message. (See GA129 in 7.3.15.)
- 064(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Non-receipt-Notification**, then `ma_start_delivery` will set the **Discard-Reason** attribute to the reason why the subject IPM was discarded.
 Testing Requirements: The value of the **Discard-Reason** attribute will be determined from the conveyed message. (See GA131 in 7.3.15.)
- 065(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Non-receipt-Notification**, then `ma_start_delivery` will set the **Non-receipt-Reason** attribute to the reason why the originator of the nonreceipt notification did not receive the subject IPM (after it was delivered).
 Testing Requirements: The value of the **Non-receipt-Reason** attribute will be determined from the conveyed message. (See GA133 in 7.3.15.)
- 066(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Non-receipt-Notification** and the value of the **IPM-Return-Requested** attribute of the subject recipient specifier is **true** and the subject IPM was not converted for delivery to the originator of the nonreceipt notification, then `ma_start_delivery` will set the **Returned-IPM** attribute to the subject IPM.
 Testing Requirements: The value of the **Returned-IPM** attribute will be determined from the conveyed message. (See GA135 in 7.3.15.)
- 067(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Non-receipt-Notification** and the value of the **IPM-Return-Requested** attribute of the subject recipient specifier is not **true** or the subject IPM was converted for delivery to the originator of the nonreceipt notification, then `ma_start_delivery` will not set the **Returned-IPM** attribute to the subject IPM.
 Testing Requirements: The value of the **Returned-IPM** attribute will be determined from the conveyed message. (See GA136 in 7.3.15.)
- 068(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Office-Document-Architecture-Body-Part**, then `ma_start_delivery` will set the **External-EITs** attribute of the instance of the MH Class **EITs** that accompanies the body part to include the following values: The value of the **Application-Profile** attribute of the body part or **id-et-oda-data**.
 Testing Requirements: The value of the **Application-Profile** attribute will be determined from the conveyed message. (See GA138 in 7.3.16.)
- 069(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Office-Document-Architecture-Body-Part**, then `ma_start_delivery` will set the **Architecture-Class** attribute to identify the class of the ODA document.
 Testing Requirements: The value of the **Architecture-Class** attribute will be determined from the conveyed message. (See GA139 in 7.3.16.)

- 070(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Office-Document-Architecture-Body-Part** and the IPM includes ODA Interchange Data Elements, then **ma_start_delivery** will set the **ODA-Document** attribute to the ODA Interchange Data Elements of the ODA document, one string element per ODA Interchange Data Element.
 Testing Requirements: The values of the **ODA-Document** attribute will be determined from the conveyed message. (See GA141 in 7.3.16.)
- 071(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **OR-Descriptor** and the **Free-Form-Name** attribute is absent, or the O/R descriptor appears in a value of the **Reply-Recipients** attribute of an IPM, or the O/R descriptor is the value of the **Recipient** attribute of a recipient specifier for which a request is made of the recipient, then **ma_start_delivery** will set the **Formal-Name** attribute to one of the O/R names of the user.
 Testing Requirements: The value of the **Formal-Name** attribute will be determined from the conveyed message. (See GA143 in 7.3.17.)
- 072(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **OR-Descriptor** and the **Formal-Name** attribute is absent, then **ma_start_delivery** will set the **Free-Form-Name** attribute to an informal name for the user.
 Testing Requirements: The value of the **Formal-Name** attribute will be determined from the conveyed message. (See GA145 in 7.3.17.)
- 073(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **OR-Descriptor** and the user who generates the O/R descriptor opts to provide the telephone number of the user, then **ma_start_delivery** will set the **Telephone-Number** attribute to the telephone number provided.
 Testing Requirements: The value of the **Telephone-Number** attribute will be determined from the conveyed message. (See GA147 in 7.3.17.)
- 074(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Receipt-Notification**, then **ma_start_delivery** will set the **Acknowledgement-Mode** attribute to identify how the Receipt Notification was originated.
 Testing Requirements: The value of the **Acknowledgement-Mode** attribute will be determined from the conveyed message. (See GA149 in 7.3.18.)
- 075(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of IM Class **Receipt-Notification**, then **ma_start_delivery** will set the **Receipt-Time** attribute to the date and time at which the originator of the receipt notification received the subject IPM.
 Testing Requirements: The value of the **Receipt-Time** attribute will be determined from the conveyed message. (See GA151 in 7.3.18.)

- 076(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Receipt-Notification** and the user who originates the Receipt Notification opts to provide supplementary information about receipt of the subject IPM, then **ma_start_delivery** will set the **Supplementary-Receipt-Info** attribute to the information provided.
Testing Requirements: The value of the **Supplementary-Receipt-Info** attribute will be determined from the conveyed message. (See GA153 in 7.3.18.)
- 077(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Recipient-Specifier**, then **ma_start_delivery** will set the **IPM-Return-Requested** attribute to indicate whether the recipient is asked to return the IPM in any Non-Receipt Notification.
Testing Requirements: The value of the **IPM-Return-Requested** attribute will be determined from the conveyed message. (See GA155 in 7.3.19.)
- 078(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Recipient-Specifier**, then **ma_start_delivery** will set the **Notification-Request** attribute to the kinds of IPN requested of the recipient in the circumstances prescribed for such IPNs.
Testing Requirements: The value of the **Notification-Request** attribute will be determined from the conveyed message. (See GA157 in 7.3.19.)
- 079(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Recipient-Specifier**, then **ma_start_delivery** will set the **Recipient** attribute to an O/R descriptor for the recipient.
Testing Requirements: The value of the **Recipient** attribute will be determined from the conveyed message. (See GA159 in 7.3.19.)
- 080(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Recipient-Specifier**, then **ma_start_delivery** will set the **Reply-Requested** attribute to indicate whether a reply is requested of the recipient.
Testing Requirements: The value of the **Reply-Requested** attribute will be determined from the conveyed message. (See GA161 in 7.3.19.)
- 081(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Teletex-Body-Part**, then **ma_start_delivery** will set the **Teletex-Document** attribute to the Teletex document, one page per attribute value.
Testing Requirements: The values of the **Teletex-Body-Part** attribute will be determined from the conveyed message. (See GA163 in 7.3.20.)
- 082(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
When **ma_start_delivery** generates an instance of IM Class **Teletex-Body-Part** and the IPM contains two or more Teletex body parts, then **ma_start_delivery** shall set the **Teletex-NBPs** attribute to the NBPs of the document.

- Testing Requirements: The value of the **Teletex-NBPs** attribute will be determined from the conveyed message. (See GA165 in 7.3.20.)
- 083(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Teletex-Body-Part**, then `ma_start_delivery` will set the **Telex-compatible** attribute to indicate whether the Teletex document is Telex-compatible.
 Testing Requirements: The value of the **Telex-compatible** attribute will be determined from the conveyed message. (See GA167 in 7.3.20.)
- 084(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Unidentified-Body-Part**, then `ma_start_delivery` will set the **Unidentified-Data** attribute to the BER data that is the whole body part including its leading implicit ASN.1 context tag.
 Testing Requirements: The value of the **Unidentified-Data** attribute will be determined from the conveyed message. (See GA169 in 7.3.21.)
- 085(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Unidentified-Body-Part**, then `ma_start_delivery` will set the **Unidentified-Tag** attribute to the ASN.1 context tag number that differentiates this body part.
 Testing Requirements: The value of the **Unidentified-Tag** attribute will be determined from the first few octets of the unidentified data in the conveyed message. (See GA171 in 7.3.21.)
- 086(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **USA-Nationally-Defined-Body-Part**, then `ma_start_delivery` will set the **Body-Part-Number** attribute to the value assigned by NIST to universally identify a particular body part type.
 Testing Requirements: The value of the **Body-Part-Number** attribute will be determined from the conveyed message. (See GA173 in 7.3.22.)
- 087(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **USA-Nationally-Defined-Body-Part**, then `ma_start_delivery` will set the **USA-Data** attribute to the BER binary data.
 Testing Requirements: The value of the **USA-Data** attribute will be determined from the conveyed message. (See GA175 in 7.3.22.)
- 088(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of IM Class **Videotex-Body-Part**, then `ma_start_delivery` will set the **Videotex-Data** attribute to the Videotex data.
 Testing Requirements: The value of the **Videotex-Data** attribute will be determined from the conveyed message. (See GA177 in 7.3.23.)

- 089(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM Class **Videotex-Body-Part**, then **ma_start_delivery** will set the **Videotex-Syntax** attribute to identify the syntax of the Videotex data using a value from T.100 or T.101.
 Testing Requirements: The value of the **Videotex-Syntax** attribute will be determined from the conveyed message. (See GA179 in 7.4.1)
- 090(D) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Acknowledgement Mode**, then the value of the instance will be chosen from the following values: **automatic** or **manual**. (See GA181 in 7.4.1.)
- 091(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Acknowledgement Mode** and the originator authorizes Receipt Notifications in blanket fashion, then **ma_start_delivery** will set the value of **Acknowledgement-Mode** to **automatic**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA182 in 7.4.1.)
- 092(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Acknowledgement Mode** and the originator authorizes Receipt Notifications individually, then **ma_start_delivery** will set the value of **Acknowledgement-Mode** to **manual**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA183 in 7.4.1.)
- 093(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Discard Reason**, then the value of the instance will be chosen from the following values: **ipm-expired**, **ipm-obsolete**, **no-discard**, or **user-terminated**. (See GA184 in 7.4.2.)
- 094(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Discard Reason** and the IPM is discarded because the time identified by the **Expiry-Time** attribute of the IPM has arrived, then **ma_start_delivery** will set the value of **Discard-Reason** to **ipm-expired**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA185 in 7.4.2.)
- 095(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When **ma_start_delivery** generates an instance of the IM enumeration syntax **Discard Reason** and the IPM is discarded because the **Obsolete-IPMs** attribute of another IPM delivered to the recipient identifies the IPM, then **ma_start_delivery** will set the value of **Discard-Reason** to **ipm-obsolete**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA186 in 7.4.2.)

- 096(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Discard Reason** and the IPM is not discarded, then `ma_start_delivery` will set the value of **Discard-Reason** to **no-discard**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA187 in 7.4.2.)
- 097(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Discard Reason** and the IM subscription of the recipient has terminated, then `ma_start_delivery` will set the value of **Discard-Reason** to **user-terminated**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA188 in 7.4.2.)
- 098(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **IA5 Repertoire**, then the value of the instance will be chosen from the following values: **IA5** or **ITA2**. (See GA189 in 7.4.3.)
- 099(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **IA5 Repertoire** and the IA5 text body part of the IPM can employ the full IA5 character set, then `ma_start_delivery` will set the value of **IA5-Repertoire** to **IA5**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA190 in 7.4.3.)
- 100(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **IA5 Repertoire** and the IA5 text body part of the IPM is restricted to the ITA2 character set, then `ma_start_delivery` will set the value of **IA5-Repertoire** to **ITA2**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA191 in 7.4.3.)
- 101(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Importance**, then the value of the instance will be chosen from the following values: **high**, **low**, or **routine**. (See GA192 in 7.4.4.)
- 102(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **ISO 6937 Repertoire**, then the value of the instance will be chosen from the following values: **graphic**, **part1-and-part2**, or **teletex**. (See GA193 in 7.4.5.)
- 103(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **ISO 6937 Repertoire** and the text portion of the ISO 6937 text body part of the

IPM can use the full graphic repertoire, then **ma_start_delivery** will set the value of **ISO-6937-Repertoire** to **graphic**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA194 in 7.4.5.)

104(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **ISO 6937 Repertoire** and the text portion of the ISO 6937 text body part of the IPM is restricted to the Teletex subrepertoire, then **ma_start_delivery** will set the value of **ISO-6937-Repertoire** to **teletex**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA195 in 7.4.5.)

105(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Non-receipt Reason**, then the value of the instance will be chosen from the following values: **ipm-auto-forwarded** or **ipm-discarded**. (See GA196 in 7.4.6.)

106(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Non-receipt Reason** and the user has not received the IPM because the IPM was automatically forwarded, then **ma_start_delivery** will set the value of **Non-receipt-Reason** to **ipm-auto-forwarded**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA197 in 7.4.6.)

107(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Non-receipt Reason** and the user has not received the IPM because the IPM was discarded, then **ma_start_delivery** will set the value of **Non-receipt-Reason** to **ipm-discarded**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA198 in 7.4.6.)

108(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Notification Request**, then the value of the instance will be chosen from the following values: **always**, **never**, or **non-receipt**. (See GA199 in 7.4.7.)

109(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Notification Request** and the IPM requests the recipient to issue either a Receipt Notification or a Non-Receipt Notification as appropriate, then **ma_start_delivery** will set the value of **Notification-Request** to **always**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA200 in 7.4.7.)

110(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:

When **ma_start_delivery** generates an instance of the IM enumeration syntax **Notification Request** and the IPM requests the recipient to issue neither a Receipt

Notification nor a Non-Receipt Notification, then `ma_start_delivery` will set the value of **Notification-Request** to **never**.

Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA201 in 7.4.7.)

- 111(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Notification Request** and the IPM requests the recipient to issue a Non-Receipt Notification, if appropriate, but not a Receipt Notification, then `ma_start_delivery` will set the value of **Notification-Request** to **non-receipt**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA202 in 7.4.7.)
- 112(C) If the IM 88 Package is supported or the IM 84 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Sensitivity**, then the value of the instance will be chosen from the following values: **company confidential**, **not-sensitive**, **personal**, or **private**. (See GA203 in 7.4.8.)
- 113(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Sensitivity** and the IPM should be handled according to company-defined procedures for confidential information, then `ma_start_delivery` will set the value of **Sensitivity** to **company-confidential**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA204 in 7.4.8.)
- 114(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Sensitivity** and the IPM is not sensitive, then `ma_start_delivery` will set the value of **Sensitivity** to **not-sensitive**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA205 in 7.4.8.)
- 115(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA206 in 7.4.8.)
- 116(D) If the IM 84 Package is supported or the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Sensitivity** and the IPM should be disclosed to no one other than its intended recipients, then `ma_start_delivery` will set the value of **Sensitivity** to **private**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA207 in 7.4.8.)
- 117(C) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax**, then the value of the instance will be chosen from the following values: **data-syntax-1**, **data-syntax-2**, **data-syntax-3**, **ids**, or **unspecified**. (See GA208 in 7.4.9.)

- 118(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax** and the syntax of the data portion of the Videotex body part is data syntax 1 as defined by T.100, then `ma_start_delivery` will set the value of **Videotex-Syntax** to **data-syntax-1**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA209 in 7.4.9.)
- 119(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax** and the syntax of the data portion of the Videotex body part is data syntax 2 as defined by T.100, then `ma_start_delivery` will set the value of **Videotex-Syntax** to **data-syntax-2**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA210 in 7.4.9.)
- 120(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax** and the syntax of the data portion of the Videotex body part is data syntax 3 as defined by T.100, then `ma_start_delivery` will set the value of **Videotex-Syntax** to **data-syntax-3**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA211 in 7.4.9.)
- 121(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax** and the syntax of the data portion of the Videotex body part is the IDS syntax as defined by T.100, then `ma_start_delivery` will set the value of **Videotex-Syntax** to **ids**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data unit. (See GA212 in 7.4.9.)
- 122(D) If the IM 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the IM enumeration syntax **Videotex Syntax** and the syntax of the data portion of the Videotex body part is unspecified, then `ma_start_delivery` will set the value of **Videotex-Syntax** to **unspecified**.
 Testing Requirements: The value of the enumeration syntax will be determined from the protocol data units. (See GA213 in 7.4.9.)
- 123(C) If the MH 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` is called and executes the algorithm associated with an instance of MH Class **Algorithm** and completes successfully, then the result of that execution will be returned in the **Algorithm-Result** attribute of an instance of MH Class **Algorithm-and-Result**. (See GA214 in 8.2.2.)
- 124(D) If the MH 88 Package is supported and the Delivery FU is supported:
 When `ma_start_delivery` generates an instance of the MH Class **Asymmetric-Token** for the recipient of that token and the asymmetric token protects public security-related information, then `ma_start_delivery` will set the **Public-Information** attribute to this protected information.
 Testing Requirements: The value of the **Public-Information** attribute will be the same as the value from the protocol data unit. (See GA215 in 8.2.3.)
- 125(D) If the MH 88 Package is supported and the Delivery FU is supported: