
**Information technology — Test methods
for measuring conformance to MHS-based
electronic messaging C language
interfaces — Binding for Application
Program Interface (API)**

*Technologies de l'information — Méthodes d'essai pour mesurer la
conformité aux interfaces de langage C en messagerie électronique basée
sur MHS — Liant pour l'interface de programme d'application (API)*

This material is reproduced from ISO documents under International Organization for Standardization (ISO) Copyright License number IHS/ICC/1996. Not for resale. No part of these ISO documents may be reproduced in any form, electronic retrieval system or otherwise, except as allowed in the copyright law of the country of use, or with the prior written consent of ISO (Case postale 56, 1211 Geneva 20, Switzerland, Fax +41 22 734 10 79), IHS or the ISO Licensor's members.



Reference number
ISO/IEC 14367:1996(E)

AVA Documents
Document Sharing Hub

Contents

	PAGE
Section 1: General	1
1.1 Scope	1
1.2 Normative References	2
1.3 Conformance	2
Section 2: Terminology and General Requirements	5
2.1 Conventions	5
2.2 Definitions	6
2.3 General Terms	6
2.4 Abbreviations	6
Section 3: Test Methods for Section General	7
Section 4: Test Methods for Terminology and General Requirements	9
4.1 Conventions	9
4.1.1 General and Typographic Conventions	9
4.1.2 C Naming Conventions	9
4.1.3 Language-Independent Conventions	9
4.2 Definitions	10
4.3 General Terms	10
4.4 Abbreviations	10
Section 5: Test Methods for Message Handling Interfaces	11
5.1 Introduction	11
5.2 Datatypes	11
5.2.1 Feature	11
5.2.2 Interval	11
5.2.3 Object Count	12
5.2.4 Sequence Number	12
5.3 Access Functions	12
5.3.1 Cancel Submission	12
5.3.2 Close	13
5.3.3 Finish Delivery()	14
5.3.4 Finish Retrieval()	15
5.3.5 Open()	15
5.3.6 Size()	16
5.3.7 Start Delivery()	17
5.3.8 Start Retrieval()	18
5.3.9 Submit()	18

© ISO/IEC 1996

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

5.3.10 Wait()	19
5.4 Transfer Functions	20
5.4.1 Close	20
5.4.2 Finish Transfer In	21
5.4.3 Open	21
5.4.4 Size	22
5.4.5 Start Transfer In	23
5.4.6 Transfer Out	24
5.4.7 Wait	24
5.5 Return Codes	25
5.6 Header File	25
Section 6: Test Methods for Section Interpersonal Messaging Packages	27
Section 7: Test Methods for Section Message Handling Packages	35
Section 8: Test Methods for Section Secure Messaging Packages	53
Annex A (informative) Bibliography	57
Alphabetic Topical Index	59

TABLES

Table 2-1	– Derivation of C Identifiers	6
Table 4-1	– Derivation of C Identifiers	9
Table 6-1	– IM Package—Classes	27
Table 6-2	– IM Package—Acknowledgement Mode Enumeration	28
Table 6-3	– IM Package—Discard Reason Enumeration	28
Table 6-4	– IM Package—IA5 Repertoire Enumeration	28
Table 6-5	– IM Package—Auto-submitted Enumeration	28
Table 6-6	– IM Package—Importance Enumeration	28
Table 6-7	– IM Package—Nonreceipt Reason Enumeration	28
Table 6-8	– IM Package—Notification Request Enumeration	29
Table 6-9	– IM Package—Sensitivity Enumeration	29
Table 6-10	– IM Package—Videotex Syntax Enumeration	29
Table 6-11	– IM Package—Types	30
Table 6-12	– IM Package—Value Lengths	30
Table 6-13	– IM Package—Classes	31
Table 6-14	– IM Package—Acknowledgement Mode Enumeration	31
Table 6-15	– IM Package—Discard Reason Enumeration	31
Table 6-16	– IM Package—IA5 Repertoire Enumeration	31
Table 6-17	– IM Package—Auto-submitted Enumeration	32
Table 6-18	– IM Package—Importance Enumeration	32
Table 6-19	– IM Package—Nonreceipt Reason Enumeration	32
Table 6-20	– IM Package—Notification Request Enumeration	32
Table 6-21	– IM Package—Sensitivity Enumeration	32

Table 6-22	– IM Package—Videotex Syntax Enumeration	32
Table 6-23	– IM Package—Types	33
Table 6-24	– IM Package—Value Lengths	33
Table 7-1	– MH Package—Classes	35
Table 7-2	– MH Package—Classes	36
Table 7-3	– MH Package—Builtin EIT Enumeration	36
Table 7-4	– MH Package—Delivery Mode Enumeration	37
Table 7-5	– MH Package—Delivery Point Enumeration	37
Table 7-6	– MH Package—Diagnostic Enumeration	37
Table 7-7	– MH Package—Explicit Conversion Enumeration	38
Table 7-8	– MH Package—Postal Mode Enumeration	38
Table 7-9	– MH Package—Postal Report Enumeration	38
Table 7-10	– MH Package—Priority Enumeration	38
Table 7-11	– MH Package—Reason Enumeration	39
Table 7-12	– MH Package—Redirection Reason Enumeration	39
Table 7-13	– MH Package—Registration Enumeration	39
Table 7-14	– MH Package—Report Request Enumeration	39
Table 7-15	– MH Package—Security Classification Enumeration	39
Table 7-16	– MH Package—Terminal Type Enumeration	40
Table 7-17	– MH Package—Content Type Integer	40
Table 7-18	– MH Package—Content Type OIDs	40
Table 7-19	– MH Package—External EITs OIDs	40
Table 7-20	– MH Package—ODA EIT OIDs	40
Table 7-21	– MH Package—Rendition Attributes OIDs	40
Table 7-22	– MH Package—Types	41
Table 7-23	– MH Package—Value Lengths	43
Table 7-24	– MH Package—Value Numbers	43
Table 7-25	– MH Package—Class Values	44
Table 7-26	– MH Package—Action Enumeration Values	45
Table 7-27	– MH Package—Builtin EIT Enumeration Values	45
Table 7-28	– MH Package—Delivery Mode Enumeration Values	46
Table 7-29	– MH Package—Delivery Point Enumeration Values	46
Table 7-30	– MH Package—Diagnostic Enumeration Values	46
Table 7-31	– MH Package—Explicit Conversion Enumeration Values	47
Table 7-32	– MH Package—Postal Mode Enumeration Values	47
Table 7-33	– MH Package—Postal Report Enumeration Values	47
Table 7-34	– MH Package—Priority Enumeration Values	47
Table 7-35	– MH Package—Reason Enumeration Values	48
Table 7-36	– MH Package—Redirection Reason Enumeration Values	48
Table 7-37	– MH Package—Registration Enumeration Values	48
Table 7-38	– MH Package—Report Request Enumeration Values	48
Table 7-39	– MH Package—Security Classification Enumeration Values	48
Table 7-40	– MH Package—Terminal Type Enumeration Values	49
Table 7-41	– MH Package—Content Type Integer Values	49
Table 7-42	– MH Package—Content Type OIDs Values	49
Table 7-43	– MH Package—External EITs OIDs Values	49

Table 7-44 – MH Package—ODA EIT OIDs Values 49

Table 7-45 – MH Package—Rendition Attributes OIDs Values 49

Table 7-46 – MH Package—Type Values 50

Table 7-47 – MH Package—Value Lengths Values 52

Table 7-48 – MH Package—Value Numbers Values 52

Table 8-1 – SM Package—Classes 53

Table 8-2 – Identifiers for SM Package Attribute Types 53

Table 8-3 – Identifiers for SM Package Value Lengths 54

Table 8-4 – Identifiers for SM Package Value Numbers 54

Table 8-5 – Identifiers for SM Package Classes 54

Table 8-6 – Identifiers for SM Package Attribute Types 54

Table 8-7 – Identifiers for SM Package Value Lengths 55

Table 8-8 – Identifiers for SM Package Value Numbers 55

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 14367 was prepared by IEEE (as IEEE Std 1328.1-1993) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Annex A of this International Standard is for information only.

Introduction

(This introduction is not a normative part of ISO/IEC 14367, Information technology—Test methods for measuring conformance to MHS-based electronic messaging C language interfaces—Binding for Application Program Interface (API), but is included for information only.)

The purpose of this International Standard is to specify the requirements for test methods for measuring conformance to ISO/IEC 14365 {5}, which defines a C programming language binding to the X.400-Based Electronic Messaging Application Program Interface (API).

Related Standards

ISO/IEC 14365 {5} is a C programming language binding for the programming language-independent specification contained in ISO/IEC 14361 {3}. The requirements for test methods for measuring conformance to ISO/IEC 14361 {3} are contained in ISO/IEC 14363 {4}. The requirements specified in ISO/IEC 14363 {4} apply to test methods for measuring conformance to ISO/IEC 14365 {5}, in addition to the requirements specified in this International Standard.

ISO/IEC 14365 {5} is intended to be used in conjunction with ISO/IEC 14360 {2}, which provides a general purpose API for the creation, examination, modification, and deletion of OSI information objects.

The test assertions in this International Standard were produced in accordance with IEEE Std 1003.3-1991 {6}.

Overview

This International Standard contains a section corresponding to each section of ISO/IEC 14365 {5}. Each such section contains test assertions for that section of ISO/IEC 14365 {5}. A set of test methods for measuring conformance to ISO/IEC 14365 {5} is required to test these assertions in conformance with the requirements of IEEE Std 1003.3-1991 {6}.

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 International Standard. 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.

This International Standard is based on IEEE Std 1328.1-1993 {B2}, which was prepared by the P1224 Working Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society.

1) A *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, USA; Telephone: +1 202 371-0101; FAX: +1 202 728-9614.

Information technology—Test methods for measuring conformance to MHS-based electronic messaging C language interfaces—Binding for Application Program Interface (API)

Section 1: General

1.1 Scope

This International Standard defines requirements for test methods for measuring conformance to ISO/IEC 14365 (5).

ISO/IEC 14365 (5) contains a programming language binding specification for ISO/IEC 14361 (3), using the C programming language. ISO/IEC 14363 (4) contains language-independent requirements for test methods for measuring conformance to programming language binding specifications for ISO/IEC 14361 (3), such as that contained in IEEE Std ISO/IEC 14365 (5). This International Standard contains C language specific requirements for the test methods. Taken in conjunction with the requirements imposed by ISO/IEC 14363 (4), they constitute the requirements that shall be satisfied by test methods used for measuring conformance to ISO/IEC 14365 (5).

1.2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

- {1} ISO/IEC 9899: 1990,¹⁾ *Programming languages—C*.
- {2} ISO/IEC 14360: 1996, *Information technology—Open Systems Interconnection (OSI) abstract data manipulation—Application Program Interface (API) [Language independent]*.
- {3} ISO/IEC 14361: 1996, *Information technology—MHS-based electronic messaging—Application Program Interface (API) [Language independent]*.
- {4} ISO/IEC 14363: 1996, *Information technology—Test methods for measuring conformance to MHS-based electronic messaging—Application Program Interface (API) [Language independent]*.
- {5} ISO/IEC 14365: 1996, *Information technology—MHS-based electronic messaging C language interfaces—Binding for Application Program Interface (API)*.
- {6} IEEE Std 1003.3-1991,²⁾ *IEEE Standard for Information Technology—Test Methods for Measuring Conformance to POSIX*.

1.3 Conformance

A set of test methods that conforms to this International Standard shall conform to IEEE Std 1003.3-1991 [6], with references to the “POSIX.n test method specification” being interpreted as references to this International Standard, and references to “the POSIX standard for which conformance is being measured” being interpreted as references to ISO/IEC 14365 {5}.

1) ISO/IEC documents can be obtained from the ISO Central Secretariat, 1 Rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse.

2) IEEE documents can be obtained from the IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ, 08855-1331, USA.

50 In addition to meeting the conformance criteria defined in IEEE Std 1003.3-1991
51 {6}, a set of test methods that conforms to this International Standard shall test all
documentation assertions defined in this International Standard.

52 NOTE: Conformance to IEEE Std 1003.3-1991 {6} implies that the test methods will test all other
53 assertions defined in this International Standard.

54 A set of test methods that conforms to this International Standard shall also con-
55 form to ISO/IEC 14363 {4}.

Section 2: Terminology and General Requirements

2.1 Conventions

2.1.1 General and Typographic Conventions

When presented in **Synopsis** sections or in extended code fragments, C language source code is printed using fixed width font. Where C language names or code fragments appear in the text of this International Standard, the following conventions apply:

- A C language function, datatype, or code fragment appears in *italic* font. A function name is indicated by following parentheses, for example, *ma_submit()*.
- A symbolic constant, other than an error name, is surrounded by braces, for example, {OMP_O_MH_FE_BASIC_ACCESS}.
- A symbolic constant for an error name is surrounded by brackets, for example, [MH_RC_BAD_MESSAGE].

2.1.2 C Naming Conventions

How the identifier for an element of a C interface is derived from the name of the corresponding element of the generic interface depends upon the type of element as specified in Table 1. The generic name is prefixed with the character string in the second column of the table, alphabetic characters are converted to the case in the third column, and an underscore (`_`) is substituted for each hyphen (`-`) or space ().

The prefix “mhP” is reserved for use by implementors of the service. The prefixes “mhX” and “MHX” are reserved for the proprietary extension of the interface. In all other respects, such extension is outside the scope of this International Standard.

2.1.3 Language-Independent Conventions

The language-independent conventions defined in ISO/IEC 14361 (3) shall apply in this International Standard.

Table 2-1 – Derivation of C Identifiers

Element Type	Prefix	Case
Datatype	MH_	Lower
Data value	IM_, MH_, SM_	Upper
Data value (Class) [See note]	IM_C_, MH_C_, SM_C_	Upper
Data value (Class) [See note]	IM_C_, MH_C_, SM_C_	Upper
Data value (Value Length)	IM_VL_, MH_VL_, SM_VL_	Upper
Data value (Value Number)	IM_VN_, MH_VN_, SM_VN_	Upper
Data value component	none	Lower
Function	ma_, mt_	Lower
Function argument	none	Lower
Function result	none	Lower

NOTE: The C identifiers denote only the Elements components of data values of the OM String datatype.

2.2 Definitions

2.2.1 Terminology

For the purposes of this International Standard, the following definitions apply:

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

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

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

2.2.1.3 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 (B1)].

2.3 General Terms

The terms used in this International Standard are the same as those defined in the language-independent API specification from which it is derived.

2.4 Abbreviations

The abbreviations used in this International Standard are the same as those defined in the language-independent API specification from which it is derived.

Section 3: Test Methods for Section General

There are no test assertions for this section.

Section 4: Test Methods for Terminology and General Requirements

4.1 Conventions

4.1.1 General and Typographic Conventions

4.1.2 C Naming Conventions

001(A) The identifier for an element is derived from the name of the corresponding element of the generic interface, depending on the type of the element as defined in the following table:

Table 4-1 – Derivation of C Identifiers

Element Type	Prefix	Case
Datatype	MH_	Lower
Data value	IM_, MH_, SM_	Upper
Data value (Class)	IM_C_, MH_C_, SM_C_	Upper
Data value (Class)	IM_C_, MH_C_, SM_C_	Upper
Data value (Value Length)	IM_VL_, MH_VL_, SM_VL_	Upper
Data value (Value Number)	IM_VN_, MH_VN_, SM_VN_	Upper
Data value component	none	Lower
Function	ma_, mt_	Lower
Function argument	none	Lower
Function result	none	Lower

4.1.3 Language-Independent Conventions

There are no test assertions for this subclause.

4.2 Definitions

22 There are no test assertions for this clause.

4.3 General Terms

24 There are no test assertions for this clause.

4.4 Abbreviations

26 There are no test assertions for this clause.

Section 5: Test Methods for Message Handling Interfaces

5.1 Introduction

5.2 Datatypes

5.2.1 Feature

5.2.1.1 Synopsis

R01 The structure MH_feature is declared as follows:

```
typedef struct {  
    OM_object_identifier    feature;  
    OM_boolean              activated;  
} MH_feature;
```

5.2.2 Interval

5.2.2.1 Synopsis

R02 The type MH_interval is declared as follows:

```
typedef OM_uint32 MH_interval;
```

5.2.3 Object Count

5.2.3.1 Synopsis

R03 The type `MH_object_count` is declared as follows:

```
typedef OM_uint32 MH_object_count;
```

5.2.4 Sequence Number

5.2.4.1 Synopsis

R04 The type `MH_sequence_number` is declared as follows:

```
typedef OM_uint32 MH_sequence_number;
```

5.3 Access Functions

5.3.1 Cancel Submission

5.3.1.1 Synopsis

R05 If the Submission FU is supported:

When the header file `<xmh.h>` is included, then the function `ma_cancel_submission()` is declared as follows:

```
OM_return_code  
ma_cancel_submission (  
    OM_private_object    session,  
    OM_object            mts_identifier  
);
```

5.3.1.2 Results

33 001(D) If the Submission FU is supported:
34 When *ma_cancel_submission()* is called and an invalid pointer
35 was supplied as a function argument or as the receptacle for a
36 function result and no other error condition applies,
37 then it will return either [OM_SUCCESS] or the
38 [OM_POINTER_INVALID] return code. (See GA01 in 5.5.)

39 5.3.2 Close

40 5.3.2.1 Synopsis

41 R06 If the Basic Access FU is supported:
42 When the header file <xmlh.h> is included, then the function
43 *ma_close()* is declared as follows:
44 OM_return_code
45 ma_close {
46 OM_private_object session
47 };

48 5.3.2.2 Results

49 002(D) If the Basic Access FU is supported:
50 When *ma_close()* is called and an invalid pointer was supplied
51 as a function argument or as the receptacle for a function result
52 and no other error condition applies, then it will return either -
53 [OM_SUCCESS] or the [OM_POINTER_INVALID] return code
54 (See GA01 in 5.5).

5.3.3 Finish Delivery()

5.3.3.1 Synopsis

R07 If the Delivery FU is supported:
 When the header file <xmh.h> is included, then the function *ma_finish_delivery()* is declared as follows:

```

OM_return_code
ma_finish_delivery (
    OM_private_object    session,
    OM_object             delivery_confirmations,
    OM_object             non_delivery_reports
);
  
```

5.3.3.2 Description

- 003(C) If the Delivery FU is supported:
 When *ma_finish_delivery(delivery_confirmations,)* is called and completes successfully, then the absence of *delivery_confirmations* is signaled by the null pointer.
- 004(C) If the Delivery FU is supported:
 When *ma_finish_delivery(,non_delivery_reports)* is called and completes successfully, then the absence of *non_delivery_reports* is signaled by the null pointer.

5.3.3.3 Results

- 005(D) If the Delivery FU is supported:
 When *ma_finish_delivery()* is called and an invalid pointer was supplied as a function argument or as the receptacle for a function result and no other error condition applies, then it will return either [OM_SUCCESS] or the [OM_POINTER_INVALID] return code. (See GA01 in 5.5.)