

INTERNATIONAL
STANDARD

ISO/IEC
14365

First edition
1996-06-01

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

*Technologies de l'information — Interfaces de langage C en messagerie
électronique basée sur X.400 — Liant pour l'interface de programme
d'application (API)*



Reference number
ISO/IEC 14365:1996(E)

AVA **Documents**
Document Sharing Hub

Contents

	PAGE
Section 1: General	1
1.1 Scope	1
1.2 Normative References	1
1.3 Conformance	2
1.4 Test Methods	3
Section 2: Terminology and General Requirements	5
2.1 Conventions	5
2.2 Definitions	6
2.3 General Terms	7
2.4 Abbreviations	7
Section 3: Message Handling Interfaces	9
3.1 Introduction	9
3.2 Datatypes	9
3.2.1 Feature	9
3.2.2 Interval	9
3.2.3 Object Count	10
3.2.4 Sequence Number	10
3.3 Access Functions	10
3.3.1 Cancel Submission	10
3.3.2 Close	11
3.3.3 Finish Delivery	11
3.3.4 Finish Retrieval	12
3.3.5 Open	12
3.3.6 Size	13
3.3.7 Start Delivery	13
3.3.8 Start Retrieval	14
3.3.9 Submit	14
3.3.10 Wait	14
3.4 Transfer Functions	15
3.4.1 Close	15
3.4.2 Finish Transfer In	16
3.4.3 Open	16
3.4.4 Size	17
3.4.5 Start Transfer In	17
3.4.6 Transfer Out	18
3.4.7 Wait	18
3.5 Return Codes	19

© 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

3.6 Header File	19
Section 4: Interpersonal Messaging Packages	21
Section 5: Message Handling Packages	25
Section 6: Secure Messaging Packages	35
Annex A (informative) Example <ximp.h> Header	37
Annex B (informative) Example <xmhp.h> Header	41
Annex C (informative) Example <xsmhp.h> Header	51
Annex D (informative) Bibliography	53
Alphabetic Topical Index	55

TABLES

Table 2-1 – Derivation of C Identifiers	6
Table 4-1 – Identifiers for IM Package Classes	22
Table 4-2 – Identifiers for IM Package Acknowledgement Mode Enumeration	22
Table 4-3 – Identifiers for IM Package Discard Reason Enumeration	22
Table 4-4 – Identifiers for IM Package IA5 Repertoire Enumeration	22
Table 4-5 – Identifiers for IM Package Auto-submitted Enumeration	23
Table 4-6 – Identifiers for IM Package Importance Enumeration	23
Table 4-7 – Identifiers for IM Package Non-receipt Reason Enumeration	23
Table 4-8 – Identifiers for IM Package Notification Request Enumeration	23
Table 4-9 – Identifiers for IM Package Sensitivity Enumeration	23
Table 4-10 – Identifiers for IM Package Videotex Syntax Enumeration	23
Table 4-11 – Identifiers for IM Package Attribute Types	24
Table 4-12 – Identifiers for IM Package Value Lengths	24
Table 5-1 – Identifiers for MH Package Classes	25
Table 5-2 – Identifiers for MH Package Action Enumeration	26
Table 5-3 – Identifiers for MH Package Builtin EIT Enumeration	27
Table 5-4 – Identifiers for MH Package Delivery Mode Enumeration	27
Table 5-5 – Identifiers for MH Package Delivery Point Enumeration	27
Table 5-6 – Identifiers for MH Package Diagnostic Enumeration	28
Table 5-7 – Identifiers for MH Package Explicit Conversion Enumeration	28
Table 5-8 – Identifiers for MH Package Postal Mode Enumeration	29
Table 5-9 – Identifiers for MH Package Postal Report Enumeration	29
Table 5-10 – Identifiers for MH Package Priority Enumeration	29
Table 5-11 – Identifiers for MH Package Reason Enumeration	29
Table 5-12 – Identifiers for MH Package Redirection Reason Enumeration	29
Table 5-13 – Identifiers for MH Package Registration Enumeration	30

Table 5-14	–	Identifiers for MH Package Report Request Enumeration . .	30
Table 5-15	–	Identifiers for MH Package Security Classification Enumeration	30
Table 5-16	–	Identifiers for MH Package Terminal Type Enumeration . .	30
Table 5-17	–	Identifiers for MH Package Content Type Integer	30
Table 5-18	–	Identifiers for MH Package Content Type OIDs	31
Table 5-19	–	Identifiers for MH Package External EITs OIDs	31
Table 5-20	–	Identifiers for MH Package ODA EIT OIDs	31
Table 5-21	–	Identifiers for MH Package Rendition Attributes OIDs . .	31
Table 5-22	–	Identifiers for MH Package Attribute Types	32
Table 5-23	–	Identifiers for MH Package Value Lengths	34
Table 5-24	–	Identifiers for MH Package Value Numbers	34
Table 6-1	–	Identifiers for SM Package Classes	35
Table 6-2	–	Identifiers for SM Package Attribute Types	36
Table 6-3	–	Identifiers for SM Package Value Lengths	36
Table 6-4	–	Identifiers for SM Package Value Numbers	36

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 14365 was prepared by IEEE (as IEEE Std 1327.1-1993) 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.

Annexes A to D of this International Standard are for information only.

Introduction

(This introduction is not a normative part of ISO/IEC 14365, Information technology—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 define a C programming language binding for the X.400-based electronic messaging Application Program Interface (API) that is defined in ISO/IEC 14361 {3}.

Related Standards

The programming language independent specification on which the programming language binding defined in this International Standard is based is contained in ISO/IEC 14361 {3}.

ISO/IEC 14361 {3}, and the language bindings derived from it, are intended to be used in conjunction with the OSI abstract data manipulation API, ISO/IEC 14360 {2}, which provides a general-purpose API for the creation, examination, modification, and deletion of OSI information objects.

ISO/IEC 14367 {5} specifies a set of requirements to be satisfied by test methods for measuring conformance to this International Standard. In addition, such test methods must also satisfy the requirements specified in ISO/IEC 14363 {4} for test methods for measuring conformance to the programming language independent specification of the API.

Overview

The API defined in this International Standard consists of C language bindings for the language independent datatypes and operation defined in ISO/IEC 14361 {3}. In addition, this International Standard provides definitions of symbolic constants and their values associated with the OM packages defined by that International Standard.

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 1327.1-1993 {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—MHS-based electronic messaging C language interfaces—Binding for Application Program Interface (API)

Section 1: General

1.1 Scope

This International Standard defines an ISO/IEC 9989 {1} C language binding to the language-independent API defined in ISO/IEC 14361 {3}.¹⁾ It is intended to support application portability at the source-code level and to be used by application developers, system implementors, test method writers, and users.

The interface is oriented towards the needs of gateways and applications that require full access to the features of the protocols.

NOTE: There are industry-standard and proprietary APIs that can be used by simple applications that only require the main features of the protocols.

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

1) The numbers in curly brackets correspond to those of the normative references in 1.2.

below. Members of IEC and ISO maintain registers of currently valid International Standards.

- {1} ISO/IEC 9989: 1990²⁾, *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 14367: 1996, *Information technology—Test methods for measuring conformance to MHS-based electronic messaging C language interfaces—Binding for Application Program Interface (API)*.

1.3 Conformance

1.3.1 Implementation Conformance

1.3.1.1 Requirements

A conforming implementation for this International Standard shall be a conforming implementation, as defined in ISO/IEC 14361 {3}, for the programming language binding specification that is contained in this International Standard.

1.3.1.2 Documentation

A conformance document shall be available for an implementation claiming conformance to this International Standard. It shall satisfy the requirements that are defined in ISO/IEC 14361 {3} for a conformance document for a programming language binding specification.

The phrases “shall document” or “shall be documented” in this International Standard mean that documentation of the feature shall appear in the conformance document, unless the system documentation is explicitly mentioned.

The system documentation should also contain the information found in the conformance document.

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

1.3.2 Application Conformance

All applications claiming conformance to this International Standard shall fall within one of the following categories.

1.3.2.1 Strictly Conforming Application

A Strictly Conforming Application for this International Standard is a Strictly Conforming Application as defined in ISO/IEC 14361 {3} for the programming language binding specification contained in this International Standard.

1.3.2.2 Conforming Application

There is only one type of Conforming Application.

1.3.2.2.1 ISO/IEC Conforming Application

An ISO/IEC Conforming Application for this International Standard is an ISO/IEC Conforming Application as defined in ISO/IEC 14361 {3} for the programming language binding specification contained in this International Standard.

1.3.2.3 Conforming Application Using Extensions

A Conforming Application Using Extensions for this International Standard is a Conforming Application Using Extensions as defined in ISO/IEC 14361 {3} for the programming language binding specification contained in this International Standard.

1.4 Test Methods

Any measurement of conformance to this International Standard shall be performed using test methods that conform to ISO/IEC 14363 {4} and to IEEE Std 1328.1-1993 {5}.

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 the element as specified in Table 2-1. 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 extensions are 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 data type.

2.2 Definitions

2.2.1 Terminology

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

2.2.1.1 conformance document: A document provided by an implementor that contains implementation details as described in 1.3.1.2. [ISO/IEC 9945-1 {B1}]

2.2.1.2 implementation defined: An indication that the implementation shall define and document the requirements for correct program constructs and correct data of a value or behavior. [ISO/IEC 9945-1 {B1}]

2.2.1.3 may: An indication of an optional feature.

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. [ISO/IEC 9945-1 {B1}]

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

2.2.1.5 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}]

2.2.1.6 supported: A condition regarding optional functionality.

Certain functionality in this International Standard is optional, but the interfaces to that functionality are always required. If the functionality is *supported*, the interfaces work as specified by this International 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 {B1}]

2.2.1.7 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 {B1}]

2.2.1.8 undefined: An indication that this International Standard imposes no portability requirements on an application's use of an indeterminate value or its behavior with erroneous program constructs or erroneous data.

Implementations (or other standards) may specify the result of using that value or causing that behavior. An application using such behaviors is using extensions, as defined in 1.3.2.3. [ISO/IEC 9945-1 {B1}]

2.2.1.9 unspecified: An indication that this International Standard imposes no portability requirements on applications for correct program constructs or correct data regarding a value or behavior.

Implementations (or other standards) may specify the result of using that value or causing that behavior. An application requiring a specific behavior, rather than tolerating any behavior when using that functionality, is using extensions, as defined in 1.3.2.3. [ISO/IEC 9945-1 {B1}]

2.3 General Terms

The terms used in this International Standard are the same as those defined in ISO/IEC 14361 {3}.

2.4 Abbreviations

The abbreviations used in this standard are the same as those defined in ISO/IEC 14361 {3}.

Section 3: Message Handling Interfaces

3.1 Introduction

This section defines the C language binding to the message handling interfaces in ISO/IEC 14361 {3}.

3.2 Datatypes

3.2.1 Feature

3.2.1.1 C Declaration

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

3.2.1.2 Description

This single datatype, used for negotiating the features of a session, corresponds to both the `mh_feature_list_type` and `mh_activated_list_type` datatypes defined in ISO/IEC 14361 {3}.

3.2.2 Interval

3.2.2.1 C Declaration

```
typedef OM_uint32 MH_interval;
```

3.2.2.2 Description

This datatype, used to denote an interval of time in milliseconds, corresponds to the `mh_feature_type` datatype defined in ISO/IEC 14361 {3}.

3.2.3 Object Count

3.2.3.1 C Declaration

```
typedef OM_uint32 MH_object_count;
```

3.2.3.2 Description

This datatype, an integer that denotes a number of objects, corresponds to the `mh_object_count_type` datatype defined in ISO/IEC 14361 {3}.

3.2.4 Sequence Number

3.2.4.1 C Declaration

```
typedef OM_uint32 MH_sequence_number;
```

3.2.4.2 Description

This datatype, used to denote the sequence number of an object in a retrieval queue, corresponds to the `mh_sequence_number_type` datatype defined in ISO/IEC 14361 {3}.

3.3 Access Functions

The C language functions defined in this section correspond directly to the language-independent operations of ISO/IEC 14361 {3}. They are therefore not described further in this International Standard, except where there is some feature of the C language binding that is not present in the language-independent specification.

3.3.1 Cancel Submission

3.3.1.1 C Declaration

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

3.3.1.2 Description

This function, used to cancel a submitted message with the deferred delivery option, corresponds to the `ma_cancel_submission` operation defined in ISO/IEC 14361 {3}.

3.3.1.3 Errors

51 [OM_POINTER_INVALID]

52 3.3.2 Close

53 3.3.2.1 C Declaration

```
54 OM_return_code  
55 ma_close (  
56     OM_private_object    session  
57 );
```

58 3.3.2.2 Description

59 This function, used to terminate an MA session between the client and the service,
60 corresponds to the `ma_close` operation defined in ISO/IEC 14361 {3}.

61 3.3.2.3 Errors

62 [OM_POINTER_INVALID]

63 3.3.3 Finish Delivery

64 3.3.3.1 C Declaration

```
65 OM_return_code  
66 ma_finish_delivery (  
67     OM_private_object    session,  
68     OM_object            delivery_confirmations,  
69     OM_object            non_delivery_reports  
70 );
```

71 3.3.3.2 Description

72 This function, used to conclude the delivery in progress in a session, corresponds to
73 the `ma_finish_delivery` operation defined in ISO/IEC 14361 {3}.

74 3.3.3.3 Arguments

75 *delivery_confirmations*
76 When the argument *delivery_confirmations* is omitted, its
77 absence is signaled by the null pointer.

78 *non_delivery_reports*
79 When the argument *non_delivery_reports* is omitted, its absence
80 is signaled by the null pointer.

3.3.3.4 Errors

81 [OM_POINTER_INVALID]

3.3.4 Finish Retrieval**3.3.4.1 C Declaration**

```

84 OM_return_code
85 ma_finish_retrieval (
86     OM_private_object    session,
87     OM_boolean           remove
88 );

```

3.3.4.2 Description

90 This function, used to conclude the retrieval in progress in a session, corresponds to
 91 the `ma_finish_retrieval` operation defined in ISO/IEC 14361 {3}.

3.3.4.3 Errors

93 [OM_POINTER_INVALID]

3.3.5 Open**3.3.5.1 C Declaration**

```

96 OM_return_code
97 ma_open (
98     OM_public_object    user_address,
99     OM_string           client_name,
100     MH_feature          feature_list[],
101     OM_private_object   *session,
102     OM_workspace        *workspace
103 );

```

3.3.5.2 Description

105 This function, used to establish an MA session between the client and the service,
 106 corresponds to the `ma_open` operation defined in ISO/IEC 14361 {3}.

3.3.5.3 Arguments

108 *feature_list* The sequence of *MH_feature* structures is terminated by an object
 109 identifier with no components (a length of zero and any value of
 110 the data pointer).

111 *user_address* When the *user_address* argument is omitted, its absence is sig-
 112 naled by the null pointer.

client_name When the *client_name* argument is omitted, its absence is signaled by a zero-length string.

3.3.5.4 Errors

[OM_POINTER_INVALID]

3.3.6 Size

3.3.6.1 C Declaration

```
OM_return_code  
ma_size (  
    OM_private_object    session,  
    MH_object_count      *number  
);
```

3.3.6.2 Description

This function, used to determine the number of messages and reports in the delivery or retrieval queue, corresponds to the *ma_size* operation defined in ISO/IEC 14361 {3}.

3.3.6.3 Errors

[OM_POINTER_INVALID]

3.3.7 Start Delivery

3.3.7.1 C Declaration

```
OM_return_code  
ma_start_delivery (  
    OM_private_object    session,  
    OM_private_object    *object  
);
```

3.3.7.2 Description

This function, used to begin the delivery of a message or a report to one or more of the users associated with a session, corresponds to the *ma_start_delivery* operation defined in ISO/IEC 14361 {3}.

3.3.7.3 Errors

[OM_POINTER_INVALID]