

INTERNATIONAL STANDARD

**ISO/IEC
14360**

First edition
1996-06-01

Information technology — Open Systems Interconnection (OSI) abstract data manipulation — Application Program Interface (API) [Language independent]

iTeh STANDARD PREVIEW

(standards.iteh.ai)

*Technologies de l'information — Manipulation de données abstraites en
interconnexion de systèmes ouverts (OSI) — Interface de programme
d'application (API) [Indépendante du langage]*

[https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-
fe5ea2a26d37/iso-iec-14360-1996](https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996)



Reference number
ISO/IEC 14360:1996(E)

Contents

	PAGE
Section 1: General	1
1.1 Scope	1
1.2 Normative References	2
1.3 Conformance	2
1.4 Test Methods	5
Section 2: Terminology and General Requirements	7
2.1 Conventions	7
2.2 Definitions	10
Section 3: Overview	17
3.1 Purpose	17
3.2 Motivation	18
iTeh STANDARD PREVIEW (standards.iteh.ai)	
Section 4: Information Architecture	21
4.1 Introduction	21
4.2 Objects	21
4.3 Object Attributes	22
4.4 Classes	22
4.5 Class Definitions	24
4.6 Packages	25
4.7 Package Definitions	25
4.8 Workspaces	25
4.9 Storage Management	26
Section 5: Information Syntaxes	29
5.1 Introduction	29
5.2 Syntax Templates	29
5.3 Syntaxes	29
5.4 Strings	30
5.4.1 String Lengths	30
5.5 Relationship to ASN.1 Simple Types	31
5.6 Relationship to ASN.1 Useful Types	31
5.7 Relationship to ASN.1 Character String Types	32
5.8 Relationship to ASN.1 Type Constructors	33

© 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

Section 6: Service Interface	35
6.1 Introduction	35
6.2 Datatypes	35
6.2.1 Signed Integers	35
6.2.2 Unsigned Integers	36
6.2.3 Boolean	36
6.2.4 Enumeration	37
6.2.5 Exclusions	37
6.2.6 Integer	38
6.2.7 Modification	38
6.2.8 Object	38
6.2.9 Object Identifier	39
6.2.10 Private Object	39
6.2.11 Public Object	39
6.2.12 Service-Generated Public Object	40
6.2.13 Client-Generated Public Object	40
6.2.14 Attribute Set	40
6.2.15 Attribute	41
6.2.16 Value	41
6.2.17 Value Data	41
6.2.18 Null Value	42
6.2.19 String Value	42
6.2.20 String Flags	43
6.2.21 String Choice	43
6.2.22 Real	44
6.2.23 Return Code	44
6.2.24 String	45
6.2.25 Type	45
6.2.26 Type List	46
6.2.27 Value Length	46
6.2.28 Value Position	46
6.2.29 Workspace	47
6.3 Operations	47
6.3.1 Copy	49
6.3.2 Copy Value	50
6.3.3 Create	51
6.3.4 Decode	52
6.3.5 Delete	53
6.3.6 Encode	54
6.3.7 Get	55
6.3.8 Instance	58
6.3.9 Put	59
6.3.10 Read	61
6.3.11 Remove	63
6.3.12 Write	64
6.4 Return Codes	66

Section 7: Object Management Package	71
7.1 Class Hierarchy	71
7.2 Class Definitions	71
7.2.1 Encoding	71
7.2.2 External	72
7.2.3 Object	73
Section 8: Definitions of Constants	75
8.1 Introduction	75
8.2 Object Management API	75
8.3 Object Management Package	75
Annex A (informative) Bibliography	77
Alphabetic Topical Index	79

FIGURES

Figure 3-1 – Conceptual Model of Object Management	18
Figure 4-1 – Structure of an Object	22

iTeh STANDARD PREVIEW (standards.iteh.ai)

TABLES

Table 5-1 – String Syntax Identifiers ISO/IEC 14360:1996	31
Table 5-2 – Syntax for ASN.1 Simple Types https://iteh.ai/standards/iso/iso-iec-14360-1996/asn1/asn1-type-constructors/asn1-type-constructors.html#list/626296a6-aa5e-4aah-a475-3e23427181	32
Table 5-3 – Syntaxes for ASN.1 Useful Types ISO/IEC 14360:1996	32
Table 5-4 – Syntaxes for ASN.1 Character String Types	32
Table 5-5 – Syntaxes for ASN.1 Type Constructors	33
Table 6-1 – Service Interface Operations	47
Table 6-2 – Service Interface Return Codes	67
Table 7-1 – Attributes Specific to Encoding	71
Table 7-2 – Attributes Specific to External	72
Table 7-3 – Attributes Specific to Object	73
Table 8-1 – Class Object Identifier Constant Suffixes	75
Table 8-2 – Attribute Type Identifiers	76

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.

iTen STANDARD PREVIEW

International Standard ISO/IEC 14360 was prepared by IEEE (as IEEE Std 1224-1993) 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.

<https://standards.iec.ch/catalog/standards/sist/626296a6-aa5c-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996>

Annex A of this International Standard is for information only.

1 **Introduction**

2 (This introduction is not a normative part of ISO/IEC 14360, Information technology—Open Systems
3 Interconnection (OSI) abstract data manipulation—Application Program Interface (API) [Language
4 independent], but is included for information only.)

5 The purpose of this International Standard is to define a general-purpose OSI
6 Abstract Data Manipulation (OM) Application Program Interface (API) in terms
7 that are independent of any particular programming language. The OM interface
8 is designed to be used with application-specific APIs that provide OSI services to
9 allow the transfer of Abstract Syntax Notation One (ASN.1) protocol elements in
10 an application-independent fashion.

11 **Related Standards**

12 This International Standard is intended to provide the basis for the definition of
13 programming language bindings to which implementations and applications can
14 conform. Such a language binding, for the C programming language, is contained
15 in ISO/IEC 14364 {B4}.

16 This International Standard, and the language bindings derived from it, are
17 intended to be used in the definition of application-specific APIs that provide OSI
18 services, such as the X.400-based API defined in ISO/IEC 14361 {B3} and the API
19 to directory services defined in ISO/IEC 14392 {B6}.

20 ISO/IEC 14362 {4} contains a set of requirements to be satisfied by test methods for
21 measuring conformance to this International Standard. They are stated in terms
22 that are independent of any particular programming language, and they apply to
23 test methods for measuring conformance to all standards defining programming
24 language bindings to this International Standard.

25 **Overview**

26 This International Standard defines an information architecture that addresses
27 objects that are based on ASN.1. By providing tools for manipulating ASN.1
28 objects, the OM interface shields the application programmer from much of the
29 complexity of encoding and decoding ASN.1 elements using the ASN.1 Basic
30 Encoding Rules (BER).

31 The Object Management API provides a platform on which a range of application-
32 specific APIs can be built. An application program must format its data into objects
33 and then submit (or retrieve) these objects using programmatic “calls” that are
34 standardized by the OM interface.

35 The OM API consists of the syntactical definition of an OM object and of the opera-
36 tions that an application can invoke to manipulate instances of OM objects.

37 The OM API presents to the programmer a uniform model for information based on
38 the concept of classes. A class is described by a collection of OM attributes; each
39 attribute consists of one or more values, each of which is characterized by a type
40 and an OM syntax (e.g., Integer, Boolean, and String).

41 The representation hiding that the OM interface provides is not by itself fully adequate
42 to meet the needs of environments supporting several application-specific
43 APIs (e.g., where an X.400 Application API and a Directory Services API can co-
44 exist in the same run-time environment). The different APIs may impose different,
45 even conflicting, requirements on the internal representations of objects, and they
46 might be implemented by different vendors.

47 However, the OM interface does allow any number of OM interface implementa-
48 tions to coexist, each representing objects differently. This is accomplished by
49 means of workspaces.

50 Related Standards Activities

51 The following areas are under active consideration at this time, or are expected to
52 become active in the near future, concerning standards for application APIs that
53 use the mechanism defined in this International Standard. Similar efforts can be
54 anticipated in the future:¹⁾

- 55 (1) X.400-based message handling
- 56 (2) Directory services (standards.iteh.ai)
- 57 (3) FTAM API
- 58 (4) Verification testing methods [ISO/IEC 14360:1996
<https://standards.iec.ch/catalog/standards/sist/626296a6-aa5e-4aab-a475-65326d37/iso-iec-14360-1996>](https://standards.iec.ch/catalog/standards/sist/626296a6-aa5e-4aab-a475-65326d37/iso-iec-14360-1996)
- 59 (5) Network interface facilities <https://standards.iec.ch/catalog/standards/sist/626296a6-aa5e-4aab-a475-65326d37/iso-iec-14360-1996>
- 60 (6) System administration.

This International Standard is based on IEEE Std 1224-1993 {B7}, 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.

iTeh **STANDARD PREVIEW**
(standards.iteh.ai)

[ISO/IEC 14360:1996](#)

[https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-
fe5ea2a26d37/iso-iec-14360-1996](https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996)

1 **Information technology—Open Systems**
2 **Interconnection (OSI) abstract data**
3 **manipulation—Application Program**
4 **Interface (API) [Language independent]**

5 **Section 1: General**
6 **iTeh STANDARD PREVIEW**
7 **(standards.iteh.ai)**

8 This International Standard defines a standard interface supporting the manipulation
9 of complex arguments and parameters used by X.400 and Directory Services
10 APIs. The interface supports manipulation of abstract data defined in ASN.1 and
11 is for use in conjunction with, but is otherwise independent of, the X.400 and Directory Services APIs.

12 An application shall be able to link and use multiple implementations of this API.

13 This International Standard provides a language-independent specification of an
14 interface and environment to support application portability at the source-code
15 level. It is intended to be used by application developers, system implementors,
16 test method writers, and users.

17 This International Standard describes the external characteristics and facilities
18 that are of importance to applications developers, rather than the internal construction
19 techniques employed to achieve these capabilities. Special emphasis is placed
20 on those functions and facilities that are needed in a wide variety of commercial
21 applications.

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 8824: 1990¹⁾ (CCITT X.208: 1988²⁾), *Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1)*.
- {2} ISO/IEC 8825: 1990 (CCITT X.209: 1988), *Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.
- {3} ISO/IEC 9594-8: 1990 (CCITT X.509: 1988), *Information technology—Open Systems Interconnection—The Directory—Part 8: Authentication framework*.
- {4} ISO/IEC 14362: 1996, *Information technology—Test methods for measuring conformance to Open Systems Interconnection (OSI) abstract data manipulation—Application Program Interface (API) [Language independent]*.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

1.3 Conformance

[ISO/IEC 14360:1996](#)

<https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996>

1.3.1 Implementation Conformance

1.3.1.1 Requirements

A conforming implementation for a programming language binding specification for this International Standard shall meet all of the following criteria:

- (1) The implementation shall support all required behavior defined in this International Standard.
- (2) The implementation shall support all required interfaces defined in the programming language binding specification. Those interfaces shall support the behavior described in this International Standard and in the programming language specification.

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) CCITT documents can be obtained from the Telecommunication Standardization Bureau of the International Telecommunication Union, Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse.

57 (3) The implementation may provide additional functions or facilities not
58 required by this International Standard or by the programming language
59 binding specification. Each such nonstandard extension shall be identified
60 as such in the system documentation. Nonstandard extensions, when
61 used, may change the behavior of functions or facilities defined by this
62 International Standard or by the programming language binding
63 specification. The conformance document shall define an environment in
64 which an application can be run with the behavior specified by this Inter-
65 national Standard and the programming language binding specification.
66 In no case shall such an environment require modification of a Strictly
Conforming Application.

67 (4) The implementation shall comprise one or more workspaces (see 4.8),
68 which shall support the OM package (see Section 7).

69 (5) In its implementation of the Encode and Decode operations, the service
70 shall support the Basic Encoding Rules (BER) from ISO/IEC 8825 {2}.³⁾

71 1.3.1.2 Documentation

72 A conformance document with the following information shall be available for an
73 implementation claiming conformance to a programming language binding
74 specification for this International Standard. The conformance document shall be
75 in two parts. The first part shall have the same structure as this International
76 Standard, with the information presented in the appropriately numbered sections,
77 clauses, and subclauses. The second part shall have the same structure as the pro-
78 gramming language binding specification, with the information presented in the
79 appropriately numbered sections, clauses, and subclauses. The conformance doc-
80 ument shall not contain information about extended features or capabilities outside
81 the scope of this International Standard and the programming language binding
82 specification.

83 The conformance document shall identify the programming language binding
84 specification to which the implementation conforms.

85 The conformance document shall contain a statement that indicates the full
86 names, numbers, and dates of the language-independent and programming
87 language binding specification standards that apply.

88 The conformance document shall state which of the optional features defined in
89 this International Standard and in the programming language binding
90 specification are supported by the implementation.

91 The conformance document shall describe the behavior of the implementation for
92 all implementation-defined features defined in this International Standard and in
93 the programming language binding specification. This requirement shall be met
94 by listing these features and providing either a specific reference to the system
95 documentation or providing full syntax and semantics of these features. The con-
96 formance document may specify the behavior of the implementation for those

97 3) The numbers in curly brackets correspond to those of the references in 1.2.

98 features where this International Standard or the programming language binding
99 specification states that implementations may vary or where features are identified
as undefined or unspecified.

100 No specifications other than those specified by this International Standard and the
101 programming language binding specification shall be present in the conformance
102 document.

103 The phrases “shall document” or “shall be documented” in this International Stan-
104 dard or in a programming language binding specification for this International
105 Standard mean that documentation of the feature shall appear in the conformance
106 document, as described previously, unless the system documentation is explicitly
107 mentioned.

108 The system documentation should also contain the information found in the confor-
109 mance document.

110 **1.3.1.3 Conforming Implementation Options**

111 The following aspects of the behavior of the service are implementation defined:

- 112 (1) The local character set representation and the precise mappings between
113 it and the various string syntaxes.
- 114 (2) The length of the longest string that the `om_get` operation will return.
115 This number is no less than 1024 octets.
- 116 (3) Whether the service reports an exception if an object supplied to it as an
117 argument is not minimally consistent.
- 118 (4) Whether the interface operations are atomic. [ISO/IEC 14360:1996](https://www.iso.org/standard/96a6-aa5e-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996)

119 **1.3.2 Application Conformance**

120 All applications claiming conformance to a programming language binding
121 specification for this International Standard shall fall within one of the categories
122 defined in the following subclauses.

123 **1.3.2.1 Strictly Conforming Application**

124 A Strictly Conforming Application is an application that requires only the facilities
125 described in this International Standard, in the programming language binding
126 specification, and in the applicable language standards. Such an application shall
127 accept any behavior described in this International Standard or in the program-
128 ming language binding specification as unspecified or implementation-defined and,
129 for symbolic constants, shall accept any value in the ranges permitted by this
130 International Standard and the programming language binding specification.

131 **1.3.2.2 Conforming Application**

132 There is only one type of Conforming Application.

1.3.2.2.1 ISO/IEC Conforming Application

133 An ISO/IEC Conforming Application of a programming language binding
134 specification for this International Standard is an application that uses only the
135 facilities described in this International Standard, in the programming language
136 binding specification, in the applicable language standards, and in other language
137 binding standards approved by ISO or IEC. Such an application shall include a
138 statement of conformance that documents all options and limit dependencies, and
139 all other ISO or IEC standards used.

140 **1.3.2.3 Conforming Application Using Extensions**

141 A Conforming Application Using Extensions of a programming language binding
142 specification for this International Standard is an application that differs from a
143 Conforming Application only in that it uses nonstandard facilities that are con-
144 sistent with this International Standard and with the programming language
145 binding specification. Such an application shall fully document its requirements for
146 these extended facilities, in addition to the documentation required of a Conform-
147 ing Application.

148 **1.4 Test Methods** **STANDARD PREVIEW**

149 Any measurement of conformance to a programming language binding
150 specification for this International Standard shall be performed using test methods
151 that conform to ISO/IEC 14362 (4) and to any additional requirements that may be
152 imposed by the programming language binding specification.
[https://standards.iec.ch/catalog/standards/iso/02291/iso-iec-14360-1996-a475-
fe5ea2a26d37/iso-iec-14360-1996](https://standards.iec.ch/catalog/standards/iso/02291/iso-iec-14360-1996-a475-fe5ea2a26d37/iso-iec-14360-1996)

iTeh **STANDARD PREVIEW**
(standards.iteh.ai)

[ISO/IEC 14360:1996](#)

[https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-
fe5ea2a26d37/iso-iec-14360-1996](https://standards.iteh.ai/catalog/standards/sist/626296a6-aa5e-4aab-a475-fe5ea2a26d37/iso-iec-14360-1996)

1

Section 2: Terminology and General Requirements

2

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. The first letters of language-independent OM class and OM attribute names are capitalized (e.g., **Arbitrary-Encoding**).

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

The use of fonts in this International Standard is as follows.

- The **Helvetica** font is used for:
 - Language-independent operation names, such as `om_copy_value`
 - Language-independent datatype names, such as `om_exclusions_type`
 - Language-independent error names, such as `no_such_class`
- The *italic* font is used for:
 - Language-independent operation arguments, such as `source_value_position`
 - The introduction of important terms
 - Cross-references in 2.2
- The **bold** font is used for:
 - Language-independent concrete OM class names, such as **External**
 - Language-independent OM attribute names, such as **Arbitrary-Encoding**
 - Language-independent OM attribute values, such as **insert-at-certain-point**
- The **bold italic** font is used for:
 - Language-independent abstract OM class names, such as **Object**