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]

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

Reference number
ISO/IEC 14360:1996(E)

Contents

FIGURES

TABLES

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 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.

Annex A of this International Standard is for information only.

Introduction

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

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

Related Standards

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

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

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

Overview

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

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

The OM API consists of the syntactical definition of an OM object and of the operations that an application can invoke to manipulate instances of OM objects.

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

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

However, the OM interface does allow any number of OM interface implementations to coexist, each representing objects differently. This is accomplished by means of workspaces.

Related Standards Activities

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

(1) X.400-based message handling

(2) Directory services

(3) FTAM API

(4) Verification testing methods

(5) Network interface facilities

(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.

This page intentionally left blank

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

Section 1: General

1.1 Scope

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

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

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

This International Standard describes the external characteristics and facilities that are of importance to applications developers, rather than the internal construction techniques employed to achieve these capabilities. Special emphasis is placed on those functions and facilities that are needed in a wide variety of commercial 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: 1990[1] (CCITT X.208: 1988[2]), *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].*

1.3 Conformance

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.

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

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

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

1.3.1.2 Documentation

A conformance document with the following information shall be available for an implementation claiming conformance to a programming language binding specification for this International Standard. The conformance document shall be in two parts. The first part shall have the same structure as this International Standard, with the information presented in the appropriately numbered sections, clauses, and subclauses. The second part shall have the same structure as the programming language binding specification, with the information presented in the appropriately numbered sections, clauses, and subclauses. The conformance document shall not contain information about extended features or capabilities outside the scope of this International Standard and the programming language binding specification.

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

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

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

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

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

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

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

The phrases "shall document" or "shall be documented" in this International Standard or in a programming language binding specification for this International Standard mean that documentation of the feature shall appear in the conformance document, as described previously, unless the system documentation is explicitly mentioned.

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

1.3.1.3 Conforming Implementation Options

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

(1) The local character set representation and the precise mappings between it and the various string syntaxes.

(2) The length of the longest string that the om_get operation will return. This number is no less than 1024 octets.

(3) Whether the service reports an exception if an object supplied to it as an argument is not minimally consistent.

(4) Whether the interface operations are atomic.

1.3.2 Application Conformance

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

1.3.2.1 Strictly Conforming Application

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

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 of a programming language binding specification for this International Standard is an application that uses only the facilities described in this International Standard, in the programming language binding specification, in the applicable language standards, and in other language binding standards approved by ISO or IEC. Such an application shall include a statement of conformance that documents all options and limit dependencies, and all other ISO or IEC standards used.

1.3.2.3 Conforming Application Using Extensions

A Conforming Application Using Extensions of a programming language binding specification for this International Standard is an application that differs from a Conforming Application only in that it uses nonstandard facilities that are consistent with this International Standard and with the programming language binding specification. Such an application shall fully document its requirements for these extended facilities, in addition to the documentation required of a Conforming Application.

1.4 Test Methods

Any measurement of conformance to a programming language binding specification for this International Standard shall be performed using test methods that conform to ISO/IEC 14362 {4} and to any additional requirements that may be imposed by the programming language binding specification.

This page intentionally left blank

Section 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. 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***