

TECHNICAL REPORT

**ISO/IEC
TR 14252**

**ANSI/IEEE
Std 1003.0**

First edition
1996-12-15

Information technology — Guide to the POSIX® Open System Environment (OSE)

*Technologies de l'information — Guide pour l'environnement de système
ouvert (OSE) POSIX®*



Reference number
ISO/IEC TR 14252:1996(E)
ANSI/IEEE
Std 1003.0-1995 edition

AVA **Documents**
Document Sharing Hub

ISBN 1-55937-692-9

Copyright © 1996 by

**The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017, USA**

*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.*

Information technology— Guide to the POSIX® Open System Environment (OSE)

Sponsor

Portable Applications Standards Committee
of the
IEEE Computer Society

Approved May 2, 1995
IEEE Standards Board

Approved December 19, 1995
American National Standards Institute

Abstract: This guide presents an overview of open system concepts and their applications. Information is provided to persons evaluating systems based on the existence of, and interrelationships among, application software standards, with the objective of enabling application portability and system interoperability. A framework is presented that identifies key information system interfaces involved in application portability and system interoperability and describes the services offered across these interfaces. Standards or standards activities associated with the services are identified where they exist or are in progress. Gaps are identified where POSIX® Open System Environment services are not currently being addressed by formal standards. Finally, the concept of a profile is discussed with examples from several application domains.

Keywords: application portability, application interoperability, open system environments, profiles, POSIX®



Adopted as an International Standard by the
International Organization for Standardization
and by the
International Electrotechnical Commission



Published by
The Institute of Electrical and Electronics Engineers, Inc.



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.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 14252, which is a Technical Report of type 3, was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*.



IEEE Standards documents are developed within 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 societies and Standards Coordinating 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

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Contents

	PAGE
Introduction	v
Section 1: General	1
1.1 Scope	1
1.2 Normative References	1
1.3 Conformance	10
1.4 Test Methods	11
Section 2: Terminology	13
2.1 Conventions	13
2.2 Definitions	14
2.2.1 Terminology	14
2.2.2 General Terms	15
2.2.3 Abbreviations	20
Section 3: POSIX Open System Environment (OSE)	23
3.1 POSIX OSE — General Objectives	24
3.2 POSIX OSE Reference Model	27
3.3 POSIX OSE Services	36
3.4 POSIX OSE Standards	37
3.5 POSIX Profiles	39
3.6 PII's	40
Section 4: POSIX OSE Services	43
4.1 Language Services	45
4.2 Core System Services	54
4.3 Communication Services	67
4.4 Database Services	83
4.5 Data Interchange Services	94
4.6 Transaction Processing Services	101
4.7 User Command Interface Services	110
4.8 Character-Based User Interface Services	118
4.9 Windowing System Services	124
4.10 Graphics Services	140
4.11 Application Software Development Support Services	154
Section 5: POSIX OSE Cross-Category Services	159
5.1 Internationalization Services	160
5.2 System Security Services	175
5.3 Systems Management Services	182
Section 6: Profiles	195
6.1 Scope	195

6.2	Concepts Related to Profiles	196
6.3	Guidance to Profile Developers	197
6.4	Types of Profiles	202
Section 7: POSIX SP Profiling Efforts		203
7.1	Introduction	203
7.2	Multiprocessing Systems Platform Profiles	204
7.3	POSIX Interactive Systems AEP	206
7.4	Supercomputing AEP	206
7.5	Realtime AEPs	208
Annex A (informative) Bibliography		211
Annex B (informative) Standards Organizations and Contact Information		217
B.1	Introduction	217
B.2	The Formal Standards Groups	218
B.3	Related Organizations	237
Alphabetic Topical Index		249

FIGURES

Figure 3-1	– POSIX OSE Reference Model	28
Figure 3-2	– POSIX OSE Reference Model — Entities	29
Figure 3-3	– POSIX OSE Reference Model — Interfaces	31
Figure 3-4	– POSIX OSE Reference Model — Distributed Systems	35
Figure 3-5	– Distributed Application Platform Implementation	36
Figure 3-6	– Service Components and Interfaces	40
Figure 4-1	– POSIX OSE Language Services Reference Model	47
Figure 4-2	– POSIX OSE Core System Services Reference Model	55
Figure 4-3	– POSIX OSE Communication Services Reference Model	68
Figure 4-4	– POSIX OSE Communication Standards	80
Figure 4-5	– Traditional Database Model	84
Figure 4-6	– POSIX OSE Database Services Reference Model	84
Figure 4-7	– POSIX OSE Data Interchange Services Reference Model	95
Figure 4-8	– Transaction Processing Model	103
Figure 4-9	– POSIX OSE Transaction Processing Services Reference Model	105
Figure 4-10	– POSIX OSE User Command Interface Services Reference Model	111
Figure 4-11	– POSIX OSE Character-Based User Interface Services Reference Model	120
Figure 4-12	– POSIX OSE Windowing System Services Reference Model	126
Figure 4-13	– Computer Graphics Reference Model Level Structure	142
Figure 4-14	– POSIX OSE Graphics Services Reference Model	143
Figure 4-15	– POSIX OSE Application Software Development Services Reference Model	155

Figure 5-1	– POSIX OSE Reference Model for Systems Management . .	183
Figure B-1	– Selected Major Standards and Standards-Influencing Bodies	219
Figure B-2	– IEEE Standards Diagram	233

TABLES

Table 2-1	– Typographical Conventions	13
Table 4-1	– Mapping of Service Categories to Section 4 Clauses . . .	43
Table 4-2	– Language Standards	51
Table 4-3	– Core System Services Standards	62
Table 4-4	– Systems Services Standard Language Bindings	63
Table 4-5	– Functionality of ISO/IEC 9945-1: 1990	64
Table 4-6	– Communication Standards — APIs	74
Table 4-7	– Communication Standards — EEIs	75
Table 4-8	– Communication Standards — Services at the EEI	76
Table 4-9	– Communication Standard Language Bindings	76
Table 4-10	– Database Standards	90
Table 4-11	– Database Standard Language Bindings	91
Table 4-12	– Data Interchange Standards	98
Table 4-13	– Transaction Processing Standards	108
Table 4-14	– Transaction Processing Standard Language Bindings . .	109
Table 4-15	– User Command Interface Processing Standards	116
Table 4-16	– User Command Interface Processing Language Bindings .	116
Table 4-17	– Character-Based User Interface Standards	122
Table 4-18	– Character-Based User Interface Standard Language Bindings	123
Table 4-19	– Windowing Standards	136
Table 4-20	– Windowing Graphical Standard Language Bindings . . .	137
Table 4-21	– POSIX OSE Graphics Services Reference Model Standards	148
Table 4-22	– Graphics Standard Language Bindings	148
Table 4-23	– Application Software Development Support Standards . .	157
Table 4-24	– Application Software Development Support Services Bindings	157
Table 5-1	– Internationalization Standards	170
Table 5-2	– System Security Standards	180
Table 5-3	– System Security Standard Language Bindings	180
Table 5-4	– System Management Standards	192
Table 5-5	– Systems Management Standard Language Bindings . . .	192
Table 7-1	– POSIX SPs in Progress	204

Introduction

(This introduction is not a normative part of IEEE Std 1003.0-1995 or of ISO/IEC TR 14252:1996.)

Purpose

This guide describes the POSIX Open System Environment (POSIX OSE). It is intended to be used by anyone interested in using standards to construct an information processing system, including consumers, systems integrators, application developers, systems providers, and procurement agencies.

The scope of this guide is much broader than a single standard. This guide identifies standards from many different areas produced by many different organizations. The POSIX OSE is intended to be broad enough to cover the entire scope of general-purpose information processing systems. While the intent of this guide is to identify completely the user services for a general-purpose information processing system, it is acknowledged that this will take some time, and this version of the guide may be incomplete in areas that are still evolving.

It is important to note that this guide is not a base standard itself; it merely identifies standards that might be used when constructing a complete information processing system.

It is not appropriate to claim conformance to this guide because it contains no mandatory requirements. This guide is intended to be used only as a source of reference material.

Although this guide is a product of the IEEE POSIX standardization efforts, its scope is much broader than those efforts. IEEE POSIX is currently developing base standards and standardized profiles focused primarily on application programming interfaces. At the end of the introduction is a cross-reference of the POSIX standardization efforts and where they fit into the POSIX OSE. For a more detailed discussion of POSIX profiling projects, see Section 7.

The process of selecting standards for a particular application domain is called profiling. Recommendations for the production of different types of profiles are included in this guide.

It may never be necessary to implement an information processing system that provides an implementation of every standard in the POSIX OSE.

In addition to listing and categorizing existing standards efforts, this guide identifies important services that standards have not yet addressed. In areas where these services are not addressed, emerging standards efforts and existing public specifications are described. These emerging standards and public specifications are not part of the POSIX OSE. They are included in this guide to identify some of the existing work that has been done in areas that are gaps in the POSIX OSE. This guide does not promote the use of these specifications that are outside the POSIX OSE. They are included for information purposes only.

User needs and standards to meet those services are continuously expanding. As such, this guide will need regular revision to incorporate new user services and

the new standards that evolve to meet those user needs.

The POSIX OSE Reference Model

To describe the POSIX OSE, this guide develops a reference model used to classify information processing standards. The reference model categorizes standards at two types of interfaces:

Application Program Interface (API) Standards

These standards govern how application software interacts with the computer system. These standards affect application portability.

External Environment Interface (EEI) Standards

These standards affect how an information processing system interacts with its external environment. These standards affect system interoperability, user interface usability, and data portability.

These standards allow users to procure portions of their information processing systems independently from multiple vendors according to the needs of each user.

The services provided at the interfaces are classified into four major categories:

- System services
- Communication services
- Information services
- Human/Computer interaction services

Within these categories, service component areas are identified.

Using the reference model, a general set of services for each component area is developed. For each of the services, existing or emerging standards are identified that address each of the services. If a service is not completely addressed by an existing or emerging standard, this gap in the standards is noted.

Goals

The POSIX OSE described in this guide should provide services to satisfy the following objectives, summarized from 3.1.

Application Portability at the Source Code Level

To allow for movement of source code and data to a variety of application platforms

System Interoperability

To allow application software and application platform interoperability

User Portability

To allow people to use a wide range of application platforms without retraining

Accommodation of Standards

To provide users and vendors with information about key interface specifications related to OSE objectives

- 78 ***Accommodation of New Information System Technology***
 79 To allow for migration to new technologies and a variety of marketplace
 80 solutions
- 81 ***Application Platform Scalability***
 82 To allow portability and software reuse across application platform
 83 types
- 84 ***Distributed System Scalability***
 85 To assure that related standards do not inappropriately limit the
 86 growth of distributed systems
- 87 ***Implementation Transparency***
 88 To allow the widest latitude in providing consistent and standard inter-
 89 faces to the application, regardless of the underlying implementation
 90 technology
- 91 ***Functional Requirements of the User***
 92 To allow clear statement of user needs and provide context for identify-
 93 ing related standards

94 **Benefits**

95 The following items are some of the benefits derived from the use of POSIX OSE.

96 ***Integration of Components From Multiple Vendors***

97 As the standards for system integration and system interoperability are
 98 produced and implemented, users will have the choice of mixing
 99 software and equipment from multiple vendors. This will allow users to
 100 tailor their information processing system to their particular needs by
 101 selecting hardware and software based on the needs of the application
 102 rather than the ability of the hardware and software to interoperate
 103 with the existing equipment.

104 ***Efficient Development and Implementation***

105 Normally, systems users and providers have development and imple-
 106 mentation activities that utilize personnel possessing skills in a specific
 107 computer environment. As a result of this specialization, a change in
 108 the target computer environment for a developer requires significant
 109 retraining expense. As standards for application portability, system
 110 interoperability, and system integration are developed, computer per-
 111 sonnel will begin to develop skills in working with these standards.

112 This will allow a company to hire personnel with existing skills that can
 113 be put to use in their operation. In addition, within a company,
 114 resources can be redeployed between development efforts with a
 115 minimum of retraining.

116 ***Efficient Porting of Applications***

117 The difficulty of moving an application from one hardware or software
 118 environment to another is widely known. The porting of an application
 119 that uses standards-based interfaces to another system that provides
 120 the same standards-based interfaces is considerably simpler than ports
 121 involving completely different systems. The amount of system tailoring

122 (i.e., changes to either the operating or application system required to
123 make them work well together) is greatly reduced.

124 Related Standards Activities

125 In addition to this guide, the Portable Applications Standards Committee (PASC)
126 has authorized other standards activities that are related to the content of this
127 guide.

128 The following table summarizes the current PASC standardization efforts¹⁾ and
129 how they relate to sections of this guide:

130	Project	Standard/Profile	Section
131	P1003.1, .1a	System Interfaces	4.2
132	P1003.1b, .1d	Realtime (formerly P1003.4)	4.2
133	P1003.1c	Threads (formerly P1003.4)	4.2
134	P1003.1e	Security API (formerly P1003.6.1)	5.2
135	P1003.1f	Transparent File Access (formerly P1003.8)	4.3
136	P1003.1g	Protocol-Independent Network API (formerly P1003.12)	4.3
137	P1003.2, .2b	Shell and Utilities	4.7
138	P1003.2c	Security Utilities (formerly P1003.6.2)	5.2
139	P1003.2d	Batch Queueing Extensions	4.7
140	P1003.5	Ada Bindings	4.1
141	P1003.5b	Ada Realtime Binding (formerly P1003.20)	4.1
142	P1003.9	Fortran Bindings	4.1
143	P1003.10	Supercomputing Profile	7.2
144	P1003.13	Realtime Profile	7.2
145	P1003.14	Multiprocessing Profile	7.2
146	P1003.18	POSIX Platform Profile	7.2
147	P1003.21	Realtime Distributed Systems Communications	4.3
148	P1003.22	Guide to POSIX OSE Security Framework	5.2
149	P1201.1	Uniform API for Graphical User Interfaces	4.9
150	P1201.2	User Interface Drivability	4.9
151	P1224	OSI API — Abstract Data Manipulation	4.3
152	P1224.1	OSI API — X.400 Electronic Mail/Messaging	4.3
153	P1224.2	OSI API — X.500 Directory Services (formerly P1003.17)	4.3
154	P1238.0	OSI API Common Support Functions	4.3
155	P1238.1	OSI API FTAM Test Methods and C Binding	4.3
156	P1327	OSI API Abstract Data Manipulation — C Binding	4.3
157	P1327.1	OSI API X.400 — C Binding	4.3
158	P1327.2	OSI API X.500 — C Binding	4.3
159	P1387.n	System Administration (formerly P1003.7.n)	5.3

160 1) A *Standards Status Report* that lists all current IEEE Computer Society standards projects is
161 available from the IEEE Computer Society, 1730 Massachusetts Avenue NW, Washington, DC
162 20036-1903, USA; Telephone: +1 202 371-0101; FAX: +1 202 728-9614.

Project	Standards/Profile	Section
P2003. <i>n</i>	Test Methods (formerly P1003.3. <i>n</i>)	
Most these efforts are in the areas of API standards and standardized profiles.		
Extensions are approved as “amendments” or “revisions” to this document, following IEEE and ISO/IEC procedures.		
Approved amendments are published separately until the full document is reprinted and such amendments are incorporated in their proper positions.		
If you have an interest in participating in the PASC 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. If you have an interest in participating in this work at the international level, contact your ISO/IEC national body.		
IEEE Std 1003.0-1995 was prepared by the IEEE P1003.0 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 P1003.0 group was as follows:		

Portable Applications Standards Committee

Chair:	Lowell Johnson
Vice-Chairs:	Jay Ashford
	Andrew Josey
	Barry Needham
	Charles Severance
	Jon Spencer
Secretary:	Charles Severance
Treasurer:	Peter Smith

P1003.0 Working Group Officials

Chair:	Allen Hankinson (1987-1993)
	Kevin Lewis (1993-1995)
Vice-Chairs:	Kevin Lewis (1987-1993)
	Fritz Schulz (1993-1995)
Technical Editor:	Fritz Schulz
Production Editors:	Anthony Cincotta (ISO/IEC Standard)
	Hal Jespersen (IEEE Standard)
Secretary:	Charles Severance

Working Group

Michael Aaby	Bob Gambrel	Per Pedersen
Michelle Aden	Daniel Green	Arnie Powell
Bill Allcorn	Quin Hahn	Dave Pruett
Gary Andrews	Dale Harris	Brian Purdy
Bengt Asker	John Hill	Lynwood Randolph
Jeanne Baccash	Richard Holbert	Wendy Rauch
Jayne Baker	Michael Howard	Brad Reed
Rick Barbour	Terry Humphrey	Darryl Roberts
Ralph Barker	Jeff Hustad	Mark Ruddock
Tony Barrese	E. Lee Hutchins	Nobuo Saito
Jon Becker	Jim Isaak	Greg Sawyer

210	Erwin R. Bender	Clariest Iselt	Norman Scherer
211	Rich Bergman	Petr Janecek	Carl Schmiedekamp
212	Andy Bihain	Michael Jende	Andy Schoka
213	Lorenzo Bonanni	James Johnson	Richard Scott
214	Kevin Brady	Lorraine Kevra	Glen Seeds
215	Steve Brooks	Walter C. Keyser	Ron Sellars
216	Steve Carpenter	Michael Kjolsrud	Lewis Shannon
217	Tim Carter	Bob Knighten	Karen Sheaffer
218	David Chinn	Bob Kruger	Harry Singh
219	J. J. Cinecoe	Mike Lambert	Pete Smith
220	Michel Colin	Doris Lebovits	Dukjoo Son
221	Bud Conrad	Kevin Leininger	Vinnie Squitieri
222	Art Corey	W. Edward Ludt	Keith Stobie
223	Jean-Michel Cornu	Heinz Lycklama	Jong Sung Sunwoo
224	Joe Cote	Sheila Mallela	Sandra Swearingen
225	Bernard Cox	Roger Martin	Marti Szczur
226	Elizabeth Crouse	Sunil Mehta	Ravi Tavakley
227	Francis Deckelman	Pete Meier	Eva Uristensson
228	Shane Deichman	Gary Miller	Martial Van Neste
229	Simion Diky	Manuel Carbajo Monje	Bob Voigt
230	Dave Dodge	Kevin Murphy	Andrew Walker
231	Dominic Dunlop	Yasushi Nakahara	Gentry Watson
232	Mat Einseln	Shigetatsu Nakao	Alan Weaver
233	Dave Febrache	Barry Needham	James White
234	Donna Fisher	Suy Nguyen	John Wilber
235	Don Folland	Mary Lynne Nielsen	John Williams
236	David Folsom	Patricia Oberndorf	Arnold Winkler
237	Kester Fong	Jim Oblinger	Wayne Yaddow
238	Rick Forberg	Peter Owens	Charles Young
239	Thomas Ford	Ed Palmer	George Zerdian
240		F. G. Patterson, Jr.	

241 The following persons were members of the balloting group:

242 Nick Stoughton *EurOpen Institutional Representative*
243 Robert Boucher *Uniforum Institutional Representative*

244	Norman Aaronson	Joe Gwinn	Wendy Rauch
245	Michelle Aden	Allen L. Hankinson	Robert Sarr
246	Lynda Allen	Barry Hedquist	Andrew M. Schoka
247	Bengt Asker	Hans H. Heilborn	Fritz Schulz
248	Ralph Barker	John L. Hill	Richard L. Scott
249	Richard M. Bergman	James C.M. Ho	Peter Smith
250	Andy R. Bihain	Andrew R. Huber	Jeff Stevenson
251	Robert Bismuth	Richard Hughes-Rowlands	Sandra Swearingen
252	Keith Brophy	Jim Isaak	James G. Tanner
253	Dawn Burnett	Petr Janecek	Ravi Tavakley
254	George S. Carson	Hal Jespersen	Donn S. Terry
255	Stephan M. Chan	Derek Kaufman	Andrew T. Twigger
256	Kilnam Chon	Judy Kerner	Mark-René Uchida
257	William Corwin	Lorraine C. Kevra	Martial Van Neste
258	Fred D. Crowner	Martin J. Kirk	Andrew Walker
259	Dave Decot	Greger Leijonhufvud	Stephen R. Walli
260	Shane Deichman	Kevin Lewis	Paul Wanish
261	Stephen L. Diamond	Lee W. Lucas	Bruce Weiner
262	Ron Elliott	Roger Martin	Andrew E. Wheeler
263	Richard W. Elwood	Roland McGrath	Alex White
264	Philip H. Enslow	Pete Meier	John R. Williams
265	Donna K. Fisher	Gary W. Miller	Peter Wishart
266	Donald Folland	John S. Morris	Charles R. Young
267	Bob Gambrel	Alok C. Nigam	Oren Yuen
268	Michel Gien	Patricia Oberndorf	John J. Zenor
269	Michael Gonzalez	A. W. Powell	George R. Zerdian
270		Scott E. Preece	

271 The final conditions for approval of this standard were met on May 2, 1995. This
272 standard was conditionally approved by the IEEE Standards Board on March 16,
273 1995, with the following membership:

274 **E. G. “Al” Kiener, *Chair*** **Donald C. Loughry, *Vice Chair***
275 **Andrew G. Salem, *Secretary***

276	Gilles A. Baril	Richard J. Holleman	Marco W. Migliaro
277	Clyde R. Camp	Jim Isaak	Mary Lou Padgett
278	Joseph A. Cannatelli	Ben C. Johnson	John W. Pope
279	Stephen L. Diamond	Sonny Kasturi	Arthur K. Reilly
280	Harold E. Epstein	Lorraine C. Kevra	Gary S. Robinson
281	Donald C. Fleckenstein	Ivor N. Knight	Ingo Rusch
282	Jay Forster*	Joseph L. Koepfinger*	Chee Kiow Tan
283	Donald N. Heirman	D. N. “Jim” Logothetis	Leonard L. Tripp
284		L. Bruce McClung	

285 *Member Emeritus

286 Also included are the following nonvoting IEEE Standards Board liaisons:

287	Satish K. Aggarwal
288	Richard B. Engelman
289	Robert E. Hebner
290	Chester C. Taylor

291 Mary Lynne Nielsen
292 *IEEE Standards Project Editor*

293 Apple and Macintosh are trademarks of Apple Computer, Inc., registered in the U.S. and other
294 countries.

295 Microsoft and Windows are registered trademarks of Microsoft Corporation.

296 Open Software Foundation, OSF, the OSF logo, and Motif are registered trademarks of the Open
297 Software Foundation, Inc.

298 OS/2 is a trademark of International Business Machines Corporation.

299 POSIX® is a registered certification mark of the Institute of Electrical and Electronics Engineers, Inc.

300 UNIX® is a registered trademark in the United States and other countries, licensed exclusively
301 through X/Open Company Limited.

302 X/Open and the X device are trademarks of X/Open Company Ltd.

Information technology — Guide to the POSIX Open System Environment (OSE)

Section 1: General

1.1 Scope

This guide provides

- (1) Definitions of the concepts of application portability, application interoperability, data portability, and user portability
- (2) Descriptions of services needed in the areas of application portability, application interoperability, data portability, and user portability
- (3) A survey of existing standards that address these objectives
- (4) Identification of those areas where formal standards do not exist and discussion of near-term strategies for filling these gaps
- (5) Guidance on assembling these standards into profiles

This guide includes no mandatory requirements.

1.2 Normative References

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

NOTE: All formal international, regional, and national standards cited in this guide are listed in this clause. Only those JTC 1 standards that have achieved DIS or ITU-T Recommendation status are listed. Only those national body standards that have completed the process of adoption are listed, and only when there is no international or regional standard covering the same area. All other specifications or references cited in this guide are given as bibliographic entries in Annex A.

- {1} ISO/IEC 646: 1991,¹⁾ *Information processing—ISO 7-bit coded character set for information interchange.*
- {2} ISO/IEC 1539: 1991, *Information technology—Programming languages—Fortran.*
- {3} ISO 1989: 1985 [ANSI X3.23-1985 (Reaff. 1991)], *Programming languages—COBOL.*
- {4} ISO 2014: 1976, *Information processing—Writing of calendar dates in all-numeric form.*²⁾
- {5} ISO/IEC 2022: 1994, *Information technology—Character code structure and extension techniques.*
- {6} ISO 3307: 1975, *Information processing—Representation of time of the day.*²⁾
- {7} ISO 4031: 1987, *Information processing—Representation of local time differentials.*²⁾
- {8} ISO 4217: 1990, *Codes for the representation of currencies and funds.*
- {9} ISO 4873: 1991, *Information technology—ISO 8-bit code for information interchange—Structure and rules for implementation.*
- {10} ISO 6093: 1985, *Information processing—Representation of numerical values in character strings for information interchange.*
- {11} ISO 6160: 1979 [ANSI X3.53-1976 (Reaff. 1993)], *Programming languages—PL/I.*
- {12} ISO/IEC 6429: 1992, *Information technology—Control functions for coded character sets.*
- {13} ISO/IEC 6522: 1992 [ANSI X3.74-1987 (Reaff. 1993)], *Programming languages—Information technology—PL/I general purpose subset.*
- {14} ISO 6936: 1988, *Information processing—Conversion between the two coded character sets of ISO 646 and ISO 6937-2 and the CCITT international telegraph alphabet No. 2 (ITA2).*
- {15} ISO/IEC 6937: 1994, *Information technology—Coded graphic character set for text communication—Latin alphabet.*
- {16} ISO/IEC 7185: 1990, *Information technology—Programming languages—Pascal.*

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

2) Please note that this standard is currently withdrawn.

- 58 {17} ISO/IEC 7350:1991, *Information technology—Registration of repertoires of*
59 *graphic characters from ISO 10367.*
- 60 {18} ISO/IEC 7498:1994, *Information processing systems—Open Systems Inter-*
61 *connection—Basic Reference Model: The Basic Model.*
- 62 {19} ISO 7498-2:1989, *Information processing systems—Open Systems Inter-*
63 *connection—Basic Reference Model—Part 2: Security Architecture.*
- 64 {20} ISO 7942:1985, *Information processing systems—Computer graphics—*
65 *Graphical Kernel System (GKS) functional description.*
- 66 {21} ISO/IEC 8072:1994, *Information processing systems—Open Systems Inter-*
67 *connection—Transport service definition.*
- 68 {22} ISO/IEC 8073:1992, *Information technology—Telecommunications and*
69 *information exchange between systems—Open Systems Interconnection—*
70 *Protocol for providing the connection-mode transport service.*
- 71 {23} ISO 8327:1987, *Information processing systems—Open Systems Inter-*
72 *connection—Basic connection oriented session protocol specification.*
- 73 {24} ISO 8485:1989, *Programming languages—APL.*
- 74 {25} ISO 8571, *Information processing systems—Open Systems Inter-*
75 *connection—File Transfer, Access and Management.*
- 76 {26} ISO 8601:1988, *Data elements and interchange formats—Information*
77 *interchange—Representation of dates and times.*
- 78 {27} ISO 8602:1987, *Information processing systems—Open Systems Inter-*
79 *connection—Protocol for providing the connectionless-mode transport*
80 *service.*
- 81 {28} ISO 8613, *Information processing—Text and office systems—Office Docu-*
82 *ment Architecture (ODA) and interchange format.*
- 83 {29} ISO/IEC 8632:1992, *Information technology—Computer graphics—Metafile*
84 *for the storage and transfer of picture description information.*
- 85 {30} ISO 8649:1988, *Information processing systems—Open Systems Inter-*
86 *connection—Service definition for the Association Control Service Element.*
- 87 {31} ISO 8650:1988, *Information processing systems—Open Systems Inter-*
88 *connection—Protocol specification for the Association Control Service*
89 *Element.*
- 90 {32} ISO 8651, *Information processing systems—Computer graphics—Graphical*
91 *Kernel System (GKS) language bindings.*
- 92 {33} ISO 8652:1987 (ANSI/MIL 1815A-1983), *Programming languages—Ada.*
- 93 {34} ISO/IEC 8802-3:1993 (ANSI/IEEE Std 802.3, 1993 Edition), *Information*
94 *technology—Local and metropolitan area networks—Part 3: Carrier sense*
95 *multiple access with collision detection (CSMA/CD) access method and phy-*
96 *sical layer specifications.*

1.2 Normative References

- {35} ISO/IEC 8802-4:1990 (ANSI/IEEE Std 802.4-1990), *Information processing systems—Local area networks—Part 4: Token-passing bus access method and physical layer specifications.*
- {36} ISO/IEC 8802-5:1995 (ANSI/IEEE Std 802.5-1995), *Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications.*
- {37} ISO 8805:1988, *Information processing systems—Computer graphics—Graphical Kernel System for Three Dimensions (GKS-3D) functional description.*
- {38} ISO/IEC 8806-1: ..., ³⁾ *Information processing systems—Computer graphics—Graphical Kernel System for Three Dimensions (GKS-3D) language bindings—Part 1: FORTRAN.*
- {39} ISO/IEC 8806-4:1991, *Information technology—Computer graphics—Graphical Kernel System for Three Dimensions (GKS-3D) language bindings—Part 4: C.*
- {40} ISO/IEC 8823-1:1994, *Information technology—Open Systems Interconnection—Connection-oriented presentation protocol: Protocol specification.*
- {41} ISO/IEC 8824:1990, *Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1).*
- {42} ISO/IEC 8825:1990, *Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
- {43} ISO 8859-1:1987, *Information processing—8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1.*
- {44} ISO 8879:1986, *Information processing—Text and office systems—Standard Generalized Markup Language (SGML).*
- {45} ISO 9040:1990, *Information technology—Open Systems Interconnection—Virtual Terminal Basic Class Service.*
- {46} ISO 9041, *Information technology—Open Systems Interconnection—Virtual Terminal Basic Class Protocol.*
- {47} ISO 9069:1988, *Information processing—SGML support facilities—SGML Document Interchange Format (SDIF).*
- {48} ISO/IEC 9075:1992 (ANSI X3.135-1992), *Information technology—Database languages—SQL.*
- {49} ISO 9241, *Ergonomic requirements for office work with visual display terminals (VDTs).*

³⁾ Presently at the state of Draft International Standard.

- 136 {50} ISO/IEC 9314, *Information technology—Fibre Distributed Data Interface*
137 *(FDDI)*.
- 138 {51} ISO 9316:1989, *Information processing systems—Small Computer System*
139 *Interface (SCSI)*.
- 140 {52} ISO/IEC 9541, *Information technology—Font information interchange*.
- 141 {53} ISO/IEC 9548-1: ...³⁾ *Information processing systems—Open Systems Inter-*
142 *connection—Session connectionless protocol to provide the connectionless-*
143 *mode session service*.
- 144 {54} ISO/IEC 9576:1991, *Information technology—Open Systems Inter-*
145 *connection—Connectionless presentation protocol specification*.
- 146 {55} ISO/IEC 9579, *Information technology—Open Systems Interconnection—*
147 *Remote Database Access*.
- 148 {56} ISO/IEC 9592, *Information processing systems—Computer graphics—*
149 *Programmer's Hierarchical Interactive Graphics System (PHIGS)*.
- 150 {57} ISO/IEC 9593, *Information processing systems—Computer graphics—*
151 *Programmer's Hierarchical Interactive Graphics System (PHIGS) language*
152 *bindings*.
- 153 {58} ISO/IEC 9594-1:1990, *Information technology—Open Systems Inter-*
154 *connection—The Directory—Part 1: Overview of concepts, models and*
155 *services*.
- 156 {59} ISO/IEC 9594-8:1990, *Information technology—Open Systems Inter-*
157 *connection—The Directory—Part 8: Authentication framework*.
- 158 {60} ISO/IEC 9596-1:1991, *Information technology—Open Systems Inter-*
159 *connection—Common management information protocol—Part 1:*
160 *Specification*.
- 161 {61} ISO/IEC 9636:1991, *Information technology—Computer graphics—*
162 *Interfacing techniques for dialogues with graphical devices (CGI)—*
163 *Functional specification*.
- 164 {62} ISO/IEC 9637-2:1992, *Information technology—Computer graphics—*
165 *Interfacing techniques for dialogues with graphical devices (CGI)—Data*
166 *stream binding—Part 2: Binary encoding*.
- 167 {63} ISO/IEC 9638-3:1994, *Information technology—Computer Graphics—*
168 *Interfacing techniques for dialogues with graphical devices (CGI)—*
169 *Language bindings—Part 3: Ada*.
- 170 {64} ISO 9735:1988, *Electronic data interchange for administration, commerce*
171 *and transport (EDIFACT)—Application level syntax rules (Amended and*
172 *reprinted 1990)*.

3) Presently at the state of Draft International Standard.

- 174 {65} ISO/IEC 9804:1994, *Information technology—Open Systems Inter-*
175 *connection—Service definition for the commitment, concurrency and*
176 *recovery service element.*
- 177 {66} ISO/IEC 9805-1:1994, *Information technology—Open Systems Inter-*
178 *connection—Protocol for the Commitment, Concurrency and Recovery ser-*
179 *vice element—Protocol Specification.*
- 180 {67} ISO/IEC 9899:1990, *Programming languages—C.*
- 181 {68} ISO/IEC 9945-1:1990 (IEEE Std 1003.1-1990), *Information technology—*
182 *Portable Operating System Interface (POSIX®)—Part 1: System Application*
183 *Program Interface (API) [C Language].*
- 184 {69} ISO/IEC 9945-2:1993 (IEEE Std 1003.2-1992), *Information technology—*
185 *Portable Operating System Interface (POSIX®)—Part 2: Shell and Utilities.*
- 186 {70} ISO/IEC 9995:1994, *Information technology—Keyboard layouts for text and*
187 *office systems.*
- 188 {71} ISO/IEC 10021, *Information technology—Text Communication—Message-*
189 *Oriented Text Interchange Systems (MOTIS).*
- 190 {72} ISO/IEC 10026-1:1992, *Information technology—Open Systems*
191 *Interconnection—Distributed Transaction Processing—Part 1: OSI TP*
192 *Model.*
- 193 {73} ISO/IEC 10027:1990, *Information technology—Information Resource Dic-*
194 *tionary System (IRDS) framework.*
- 195 {74} ISO/IEC 10164-4:1992 [ITU-T X.733 (1992)], *Information technology—Open*
196 *Systems Interconnection—Systems management: Alarm reporting function.*
- 197 {75} ISO/IEC 10164-5:1993 [ITU-T X.734 (1992)], *Information technology—Open*
198 *Systems Interconnection—Systems management: Event Report Management*
199 *Function.*
- 200 {76} ISO/IEC 10164-6:1993 [ITU-T X.735 (1992)], *Information technology—Open*
201 *Systems Interconnection—Systems Management: Log control function.*
- 202 {77} ISO/IEC 10164-7:1992 [ITU-T X.736 (1992)], *Information technology—Open*
203 *Systems Interconnection—Systems Management: Security alarm reporting*
204 *function.*
- 205 {78} ISO/IEC 10164-10:1995 [ITU-T X.742 (1993)], *Information processing*
206 *systems—Open Systems Interconnection—Systems Management: Usage*
207 *Metering Function.*
- 208 {79} ISO/IEC 10179: ..., ²⁾ *Information technology—Processing languages—*
209 *Document Style Semantics and Specification Language (DSSSL).*
- 210 {80} ISO/IEC 10180:1995 *Information technology—Text communication—*
211 *Standard Page Description Language (SPDL).*

212 2) Presently at the state of Draft International Standard.

- 213 {81} ISO/IEC 10206:1991, *Information technology—Programming languages—*
214 *Extended Pascal.*
- 215 {82} ISO/IEC 10279:1991, *Information technology—Programming languages—*
216 *Full BASIC.*
- 217 {83} ISO/IEC 10303, *Industrial automation systems and integration—Product*
218 *data representation and exchange.*
- 219 {84} ISO/IEC 10367:1991 *Information technology—Standardized coded graphic*
220 *character sets for use in 8-bit codes.*
- 221 {85} ISO/IEC 10514-1: ..., ¹⁾ *Information technology—Programming languages—*
222 *Modula-2.*
- 223 {86} ISO/IEC ISP 10607, *Information technology—International Standardized*
224 *Profiles AFTnn—File Transfer, Access and Management.*
- 225 {87} ISO/IEC 10641:1993, *Information technology—Computer graphics and*
226 *image processing—Conformance testing of implementations of graphics*
227 *standards.*
- 228 {88} ISO/IEC 10646-1:1993, *Information technology—Universal Multiple-Octet*
229 *Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual*
230 *Plane.*
- 231 {89} ISO/IEC 10728:1993, *Information technology—Information Resource Dic-*
232 *tionary System (IRDS) Services Interface.*
- 233 {90} ISO/IEC 10744:1992, *Information technology—Hypermedia / Time-based*
234 *Structuring Language (HyTime).*
- 235 {91} ISO/IEC 11072:1992, *Information technology—Computer graphics—*
236 *Computer Graphics Reference Model.*
- 237 {92} ISO/IEC 11730:1994, *Information technology—Programming languages—*
238 *Form Interface Management System (FIMS).*
- 239 {93} ISO/IEC 12087, *Information technology—Computer graphics and image*
240 *processing—Image Processing and Interchange (IPI)—Functional*
241 *specification.*
- 242 {94} ISO/IEC 12088-4:1995 *Information technology—Computer graphics and*
243 *image processing—Image processing and interchange (IPI)—Application*
244 *program interface language bindings—Part 4: C.*
- 245 {95} ISO/IEC 12089: ..., ⁰⁾ *Information technology—Computer graphics and*
246 *image processing—Encoding for the Image Processing and Interchange*
247 *Standard (IPI)—Encoding for the Image Interchange Facility (IIF).*
- 248 {96} ISO/IEC 12227:1995, *Information technology—Programming languages—*
249 *SQL/Ada Module Description Language (SAMeDL).*

250 1) Presently at the state of Draft International Standard.

- 251 {97} ISO/IEC 13210: 1994 (IEEE Std 1003.3-1991), *Information technology—Test*
252 *methods for measuring conformance to POSIX®.*
- 253 {98} ISO/IEC 13211-1: 1995, *Information technology—Programming languages—*
254 *Prolog—Part 1: General core.*
- 255 {99} ITU-T Recommendation I.120 (1993),¹⁾ *Integrated services digital networks*
256 *(ISDNs).*
- 257 {100} ITU-T Recommendation T.61 (1993), *Character repertoire and coded charac-*
258 *ter sets for the international teletex service.*²⁾
- 259 {101} ITU-T Recommendation X.25 (1993), *Interface between data terminal equip-*
260 *ment (DTE) and data circuit-terminating equipment (DCE) for terminals*
261 *operating in the packet mode and connected to public data networks by dedi-*
262 *cated circuit.*
- 263 {102} ITU-T Recommendation X.400 (1993), *Message handling services: Message*
264 *handling system and service overview.*
- 265 {103} ANSI X3.9-1978 (Reaff. 1989),³⁾ *American National Standard Programming*
266 *Language FORTRAN.*
- 267 {104} ANSI X3.62-1987 (Reaff. 1993), *Information Systems—Optical Character*
268 *Recognition (OCR)—Paper Used in OCR Systems.*
- 269 {105} ANSI X3.102-1992, *Information Systems—Data Communication Systems*
270 *and Services—User-Oriented Performance Parameters.*
- 271 {106} ANSI X3.138-1988, *Information Systems—Information Resource Dictionary*
272 *System (IRDS).*
- 273 {107} ANSI X3.131-1994, *Information Systems—Small Computer Systems*
274 *Interface-2 (SCSI-2).*
- 275 {108} ANSI X3.141-1987 (Reaff. 1992), *Information Systems—Data Communica-*
276 *tion Systems and Services—Measurement Methods for User-Oriented Perfor-*
277 *mance Evaluation.*
- 278 {109} ANSI X3.168-1989, *Information Systems—Database Language—Embedded*
279 *SQL.*
- 280 {110} ANSI X3.185-1992, *Information Systems—Information Resource Dictionary*
281 *Systems—IRDS Services Interface.*
- 282 {111} ANSI X3.195-1991, *Information Systems—Information Resource Dictionary*
283 *Systems (IRDS)—Export / Import File Format.*

284 1) ITU-T documents can be obtained from the ITU-T General Secretariat, International
285 Telecommunications Union, Sales Section, Place des Nations, CH-1211, Genève 20,
286 Switzerland/Suisse.

287 2) Please note that this recommendation is currently withdrawn.

288 3) ANSI documents can be obtained from the Sales Department, American National Standards
289 Institute, 11 West 42nd Street New York, NY 10036, USA.

- 290 {112} ANSI/ASME Y14.26M-1989, *Digital Representation for Communication of*
291 *Product Definition Data.*
- 292 {113} ECMA TR-47,⁴⁾ *Configuration Management Service Definition.*
- 293 {114} ECMA 138, *Security in Open Systems—Data Elements and Service*
294 *Definitions.*
- 295 {115} GB 2312-1980, China State Bureau of Standards.⁵⁾ *Coded Chinese Graphic*
296 *Character Set for Information Interchange.*
- 297 {116} ISO/IEC 9945-1:1996 [ANSI/IEEE Std 1003.1, 1996 Edition],⁶⁾ *Information*
298 *Technology—Portable Operating System Interface (POSIX[®])—Part 1: System*
299 *Application Program Interface (API) [C Language]*
- 300 NOTE: This standard was formerly called draft P1003.4.
- 301 {117} IEEE Std 1003.1c-1995, (*see ISO/IEC 9945-1:1996*)
- 302 {118} IEEE Std 1003.2d-1994, *IEEE Standard for Information Technology—*
303 *Portable Operating System Interface (POSIX[®])—Part 2: Shell and*
304 *Utilities—Amendment 1: Batch Environment.*
- 305 NOTE: This standard was formerly called draft P1003.15.
- 306 {119} IEEE Std 1003.5-1992, *IEEE Standard for Information Technology—POSIX[®]*
307 *Ada Language Interfaces—Part 1: Binding for System Application Program*
308 *Interface (API).*
- 309 {120} IEEE Std 1003.9-1992, *IEEE Standard for Information Technology—POSIX[®]*
310 *FORTTRAN 77 Language Interfaces—Part 1: Binding for System Application*
311 *Program Interface (API).*
- 312 {121} IEEE Std 1003.10-1995, *IEEE Standard for Information Technology—*
313 *POSIX[®] Supercomputing Application Environment Profile.*
- 314 {122} IEEE Std 1076-1993, *IEEE Standard VHDL Language Reference Manual.*
- 315 {123} IEEE Std 1224-1993, *IEEE Standard for Information Technology—Open*
316 *Systems Interconnection (OSI) Abstract Data Manipulation—Application*
317 *Program Interface (API) [Language Independent].*
- 318 {124} IEEE Std 1224.1-1993, *IEEE Standard for Information Technology—X.400-*
319 *Based Electronic Messaging—Application Program Interface (API)*
320 *[Language Independent].*
- 321 {125} IEEE Std 1224.2-1993, *IEEE Standard for Information Technology—*
322 *Directory Services—Application Program Interface (API) [Language*
323 *Independent].*

324 4) ECMA documents can be obtained from ECMA, 114 rue du Rhône, CH-1204 Genève,
325 Switzerland/Suisse.

326 5) CSBS documents can be obtained from the China State Bureau of Standards, P.O. Box 8010, 42
327 Hi Chun Road, Haidian District, Beijing 100088, China.

328 6) This standard is available from the IEEE, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ
329 08855-1331, USA. Telephone: 1 (800) 678-IEEE or +1 (908) 981-1393 (outside USA).

- {126} IEEE Std 1238.1-1995, *IEEE Standard for Information Technology—File Transfer, Access, and Management Services—Application Program Interface (API) [C Language Binding]*.
- {127} IEEE Std 1295-1993, *IEEE Standard for Information Technology—X Window System—Modular Toolkit Environment*.
- {128} IEEE Std 1327-1993, *IEEE Standard for Information Technology—Open Systems Interconnection (OSI) Abstract Data Manipulation C Language Interfaces—Binding for Application Program Interface (API)*.
- {129} IEEE Std 1327.1-1993, *IEEE Standard for Information Technology—X.400-Based Electronic Messaging C Language Interfaces—Binding for Application Program Interface (API)*.
- {130} IEEE Std 1327.2-1993, *IEEE Standard for Information Technology—Directory Services C Language Interfaces—Binding for Application Program Interface (API)*.
- {131} IEEE Std 1387.2-1995, *IEEE Standard for Information Technology—Portable Operating System Interface (POSIX[®]) System Administration—Part 2: Software*.
- {132} JIS X0208-1990, Japanese Industrial Standard.⁷⁾ *Code of the Japanese graphic character set for information interchange*.
- {133} JIS X0212-1990, Japanese Industrial Standard. *Code of the supplementary Japanese graphic character set for information interchange*.
- {134} KS C 5601-1987, Korean Bureau of Standards.⁸⁾ *Korean Graphic Character Set for Information Interchange*.

1.3 Conformance

It is not appropriate to claim conformance to this guide because it contains no mandatory requirements. This guide is intended to be used only as a source of reference material.

7) JIS documents can be obtained from the Japanese Standards Association, 1-24, Akasaka 4-chome, Minato-Ku, Tokyo, Japan 107.

8) KBS documents can be obtained from the Korean Bureau of Standards, 2 Chung-ang-dong Kwach'on-city, Kyonggi-do 171-11, Republic of Korea.

361 **1.4 Test Methods**

362 Not applicable.

Section 2: Terminology

1 **2.1 Conventions**

2 **2.1.1 Editorial Conventions**

3 This guide uses the following editorial and typographical conventions. A sum-
4 mary of typographical conventions is shown in Table 2-1.

5 **Table 2-1 – Typographical Conventions**

6	Reference	Example
7	C-Language function	<i>system()</i>
8	Cross-Reference: Annex	Annex A
9	Cross-Reference: Clause	2.3
10	Cross-Reference: Other standard	ISO/IEC 9999-1 { <i>n</i> }
11	Cross-Reference: Section	Section 2
12	Cross-Reference: Subclause	2.3.4, 2.3.4.5, 2.3.4.5.6
13	Defined term	(see text)
14	Figure reference	Figure 7-1
15	Table reference	Table 6-1
16	Utility name	awk

17 Numbers within braces, such as “ISO/IEC 9945-1:1990 {68},” represent cross-
18 references to the normative references clause (see 1.2). If the number is preceded
19 by a B, it represents a bibliographic entry (see Annex A). See 2.1.2 for a further
20 description of the references to POSIX working groups and standards.

21 Defined terms are shown in three styles, depending on context:

- 22 (1) Terms defined in 2.2.1 and 2.2.2 are expressed as subclause titles. Alter-
23 native forms of the terms appear in [brackets].
- 24 (2) The initial appearances of other terms, applying to a limited portion of
25 the text, are in *italics*.
- 26 (3) Subsequent appearances of the term are in the Roman font.

2.1.2 POSIX

The term “POSIX” has been evolving into a term with a number of different meanings. This subclause attempts to define the word and some related terms. The intent is to ensure that the term POSIX is used in a useful and predictable manner in this guide.

As background, note that POSIX is sometimes used to denote the formal standard ISO/IEC 9945-1:1990 {68}, sometimes to denote that standard plus related standards and drafts emerging from IEEE PASC working groups (such as P1003, P1201, P1224, P1278, P1387, etc.), and sometimes to denote the groups themselves. In all those cases, POSIX is used as a noun.

This guide uses the term “POSIX” only as an adjective, and uses it only in well-defined ways. This subclause serves as a preview of the usages in this guide of POSIX terms. (These terms are defined, formally or informally, in subsequent clauses, and the reader will be referred to those clauses as appropriate.)

This guide refers to the original POSIX standard by its standard designation, ISO/IEC 9945-1:1990 {68}, and not by the term POSIX.

The IEEE groups developing standards related to IEEE P1003 are called, in this guide, *IEEE P1003.n working groups*. Examples are the IEEE working groups P1003.2, P1003.3, etc. The names of the groups are sometimes abbreviated POSIX.2, POSIX.3, etc., but this convention is not used by this guide; confusion could result when the IEEE P1003 decimal number does not match the ISO/IEC 9945 part number. Furthermore, there are other IEEE working groups, such as P1224, that do not use the POSIX prefix. Therefore, all IEEE projects and working groups are referred to uniformly as IEEE Pnnnn.

The standards emerging out of the POSIX working groups are referred to by their formal names (e.g., IEEE Std 1003.2-1992 or IEEE P1003.10/D9) and are called either *POSIX base standards* or *POSIX SPs*.

2.2 Definitions

2.2.1 Terminology

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

2.2.1.1 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.

2.2.1.2 informative: Providing or disclosing information; instructive.

Used in standards to indicate a portion of the text that poses no requirements; the opposite of *normative*.

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 guide, but can be provided.

2.2.1.4 normative: Of, pertaining to, or prescribing a norm or a standard.

Used in standards to indicate a portion of the text that poses requirements.

2.2.1.5 should: With respect to implementations, an indication of an implementation recommendation, but not a requirement.

2.2.2 General Terms

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

2.2.2.1 accredited standards development organization: An organization recognized as a standards development organization by ISO, IEC, ITU-T, or recognized as a standards development organization by one of the member bodies of one of these three organizations.

2.2.2.2 application: The use of capabilities provided by an information system specific to the satisfaction of a set of user requirements.

NOTE: These capabilities include hardware, software, and data.

2.2.2.3 application environment profile (AEP): A profile specifying a complete and coherent specification of the Open System Environment (OSE), in which the standards, options, and parameters chosen are necessary to support a class of applications.

2.2.2.4 application platform: A set of resources, including hardware and software, that support the services on which application software will run.

The application platform provides services at its interfaces that, as much as possible, make the specific characteristics of the platform transparent to the application software.

2.2.2.5 application program interface (API): The interface between the application software and the application platform, across which all services are provided.

2.2.2.6 application software: Software that is specific to an application and is composed of programs, data, and documentation.

2.2.2.7 base standard: An approved international standard, technical report, ITU-T Recommendation, or national standard.

2.2.2.8 communication interface: That part of the API devoted to communications with other application software, external data transport facilities, and devices.

2.2.2.9 communication services interface (CSI): The boundary across which access to services for interaction between internal application software entities and application platform external entities is provided.

2.2.2.10 component profile: A profile that is made up of a formally defined subset of a single standard.

2.2.2.11 cross-category services: A set of tools or features or both that has a direct effect on the operation of one or more components of the OSE, but is not in and of itself a stand-alone component.

2.2.2.12 emerging standard: A specification that is under consideration by an accredited standards development organization, but that has not completed the process of approval by the sponsoring body.

Emerging standards are often subject to significant change prior to approval.

2.2.2.13 external environment: A set of entities external to the application platform with which services are provided.

External entities include people, exchangeable media that is not mounted in the platform, communication wiring, and other platforms.

2.2.2.14 external environment interface (EEI): The interface between the application platform and the external environment across which services are provided.

The EEI is defined primarily in support of system and application interoperability.

The primary services present at the EEI comprise

- Human/Computer interaction services
- Information services
- Communication services

2.2.2.15 hardware: Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation.

2.2.2.16 harmonization: The process of ensuring that profiles do not overlap or conflict.

2.2.2.17 human/computer interface (HCI): The boundary across which physical interaction between a human being and the application platform takes place.

2.2.2.18 information services interface (ISI): The boundary across which external, persistent storage service is provided.

2.2.2.19 interface: A shared boundary between two functional entities.

A standard specifies the services in terms of the functional characteristics and behavior observed at the interface. The standard is a contract in the sense that it documents a mutual obligation between the service user and provider and assures a stable definition of that obligation.

2.2.2.20 internationalization: The process of designing and developing an implementation with a set of features, functions, and options intended to satisfy a variety of cultural environments.

(See also *localization* in 2.2.2.26.)

2.2.2.21 interoperability: The ability of two or more systems to exchange information and to use the information that has been exchanged mutually.

2.2.2.22 language-binding API specification: A specification that documents the source code method, consistent with a specific programming language, used by an application to access services provided by an application platform.

2.2.2.23 language-independent service specification: A specification that defines a set of required functional semantics independent of the syntax and semantics of a programming language.

2.2.2.24 local adaptation: The process of modifying a product that is specific to one culture to make it specific to another culture.

2.2.2.25 locale: The definition of the user environment that depends on language and cultural conventions.

2.2.2.26 localization: The process of utilizing internationalization features to adapt an internationalized product to a specific cultural environment.

(See also *internationalization* in 2.2.2.20.)

2.2.2.27 open specifications: Specifications that are maintained by an organization that uses an open, public consensus process to accommodate new technologies and user requirements over time.

2.2.2.28 open system: A system that implements sufficient open specifications or standards for interfaces, services, and supporting formats to enable properly engineered application software

— To be ported with minimal changes across a wide range of systems from one or more suppliers

— To interoperate with other applications on local and remote systems

— To interact with people in a style that facilitates user portability

2.2.2.29 open system API: A combination of standards-based interfaces specifying a complete interface between an application program and the underlying application platform.

2.2.2.30 open system environment (OSE): A comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of applications, data, or people, as specified by information technology standards and profiles.

2.2.2.31 operating system software: Application-independent software that supports the running of application software and manages the resources of the application platform.

2.2.2.32 performance: A measure of a computer system or subsystem to perform its functions; for example, response time, throughput, or number of transactions per second.

The efficiency of a system in accomplishing pieces of work is an attribute of performance.

2.2.2.33 performance requirement: A requirement that specifies a performance characteristic that a system or system component must possess; for example, speed, accuracy, frequency.

2.2.2.34 platform internal interface (PII): The interface between application platform service components within that platform.

2.2.2.35 platform profile: A profile whose focus is on functionality and interfaces for a particular type of platform, which may be a single processor shared by a group of applications or a large distributed system with each application dedicated to a single processor.

2.2.2.36 portability (application software): The ease with which application software and data can be transferred from one application platform to another.

2.2.2.37 POSIX Standardized Profile (POSIX SP): A standardized profile that specifies the application of certain POSIX base standards in support of a class of applications and does not require any departure from the structure defined by the reference model for POSIX systems in this guide.

2.2.2.38 process: An address space and one or more threads of control that execute within that address space, and their required system resources.

196 **2.2.2.39 profile:** A set of one or more base standards and, where applicable, the
197 identification of chosen classes, subsets, options, and parameters of those base
198 standards that are necessary for accomplishing a particular function.

199 **2.2.2.40 programming language API specification:** The interface between
200 applications and application platforms traditionally associated with programming
201 language specifications, such as program control, math functions, string manipu-
202 lation, etc.

203 **2.2.2.41 protocol:** A set of semantic and syntactic rules that determine the
204 behavior of entities that interact.

205 **2.2.2.42 public specifications:** Specifications that are available, without res-
206 triction, to anyone for implementation, sublicensing, and distribution (i.e., sale) of
207 that implementation.

208 **2.2.2.43 reference model:** A structured collection of concepts and their rela-
209 tionships that scope a subject and enable the partitioning of the relationships into
210 topics relevant to the overall subject and that can be expressed by a common
211 means of description.

212 **2.2.2.44 scalability:** The ability to provide functionality up and down a gra-
213 duated series of application platforms that differ in speed and capacity.

214 **2.2.2.45 security:** The protection of computer resources (e.g., hardware,
215 software, and data) from accidental or malicious access, use, modification, des-
216 truction, or disclosure.

217 Tools for the maintenance of security are focused on availability, authentication,
218 accountability, confidentiality, and integrity.

219 **2.2.2.46 service:** A distinct part of the functionality that is provided by an
220 entity on one side of an interface to an entity on the other side of the interface.

221 **2.2.2.47 software:** The programs, procedures, rules, and any associated docu-
222 mentation pertaining to the operation of an information processing system.

223 **2.2.2.48 specification:** A document that prescribes, in a complete, precise,
224 verifiable manner, the requirements, design, behavior, or characteristics of a sys-
225 tem or system component.

226 **2.2.2.49 standard:** A document, established by consensus and approved by an
227 accredited standards development organization, that provides, for common and
228 repeated use, rules, guidelines, or characteristics for activities or their results,
229 aimed at the achievement of the optimum degree of order and consistency in a
230 given context.

231 **2.2.2.50 standardized profile:** A balloted, formal, harmonized document that
232 specifies a profile.

233 **2.2.2.51 standard development organization:** An accredited organization
234 that formally develops and coordinates standards for use by a community.

235 **2.2.2.52 thread:** A single flow of control within a process.

236 **2.2.2.53 transaction:** A unit of work consisting of an arbitrary number of indivi-
237 dual operations, all of which will either complete successfully or abort with no
238 effect on the intended resources.

239 A transaction has well-defined boundaries. A transaction starts with a request
240 from the application program and either completes successfully (commits) or has
241 no effect (abort). Both the commit and abort signify completion of a transaction.

242 **2.2.2.54 validation:** The process of testing an application or system to ensure
243 that it conforms to its specification.

244 **2.2.3 Abbreviations**

245 For the purposes of this guide, the following abbreviations apply:

246 **2.2.3.1 AEP:** Application Environment Profile.

247 **2.2.3.2 API:** Application Program Interface.

248 **2.2.3.3 ASC:** Accredited Standards Committee.

249 **2.2.3.4 CSI:** Communications Service Interface.

250 **2.2.3.5 EEI:** External Environment Interface.

251 **2.2.3.6 HCI:** Human/Computer Interface.

252 **2.2.3.7 ISI:** Information Service Interface.

253 **2.2.3.8 ISP:** International Standardized Profile.

254 **2.2.3.9 OSE:** Open System Environment.

255 **2.2.3.10 OSI:** Open System Interconnect.

256 **2.2.3.11 PASC:** The Portable Applications Standards Committee of the IEEE
257 Computer Society.

258 **2.2.3.12 PII:** Platform Internal Interface.

259 **2.2.3.13 SP:** Standardized Profile.

Section 3: POSIX Open System Environment (OSE)

The POSIX OSE provides a context for user services and standards specification. It provides a minimum standard set of conceptual information-system building blocks with associated interfaces and functionality. The POSIX OSE consists of a reference model, service definitions, standards, and profiles.

These POSIX OSE concepts are intended to be conventional within computer science. The intention is not to break new ground, but to establish a minimum and unambiguous terminology and a set of concepts for identification and resolution of portability and interoperability issues.

The POSIX OSE is defined in five parts:

- (1) General objectives that apply to the POSIX OSE as a whole are identified in 3.1.
- (2) A reference model is developed that unambiguously identifies the system under consideration for purposes of specification. The POSIX OSE reference model described in 3.2 defines system elements and key interfaces between them that are related to application software portability and interoperability.
- (3) Using the interfaces identified in the reference model, each clause of Section 4 categorizes and describes the basic services available to users across each interface. The services are defined in a generic way, based on the reference model, user requirements, and current industry practice, rather than any given implementation.

Each clause of Section 4 begins with a more detailed and specialized version of the reference model to provide a context for service specification. After defining the interfaces and services, each clause of Section 4 concludes with a discussion of standards that are related to the services.

Definition of the services is not constrained by the availability of standards. Services that are not currently satisfied by standards are discussed in the Additional Specifications and Unaddressed Services subclauses.
- (4) Section 5 discusses issues and services that directly affect all of the service categories, such as internationalization, security, and administration.
- (5) Section 6 provides guidelines for creating profiles that address various application domains. This is a brief description of how the reference model and services are applied to a variety of existing types of systems. Section 7 describes current POSIX profiles and profiling activities.

3.1 POSIX OSE — General Objectives

The POSIX OSE described in this guide should provide services to satisfy the following objectives. These objectives may be achieved to a greater or lesser degree for any given usage.

3.1.1 Application Software Portability at the Source-Code Level

The POSIX OSE should enable application software portability at the source-code level.

Rationale: Comprehensive and consistent source-code level service specifications allow porting of applications among application platform implementations (ideally without modification). Binary portability requires too tight a coupling with the application platform implementation.

Although standards-based systems improve application portability, the standards do not guarantee that an application will be portable. Applications still have to be properly engineered to ensure maximum application portability. This allows an organization to protect its investment in existing software.

Application portability is often associated with porting an entire application at one time. *Software reuse* is a term used to describe porting only a subset of a working program into a new application. The new application may or may not be executed on the same application platform. Software reuse is an important element in achieving the benefits of application portability.

3.1.2 Data Portability

The POSIX OSE should enable transfer of stored data among application platforms.

Rationale: The ability to move stored data from one application platform to another is fundamental to achieving the objective of application portability. Since application software is defined to include both the source code and application related data (see 2.2.2.6 and 3.2.1), porting of the application requires both the application source code and the data to be moved to the destination platform.

Data portability also directly supports the application software and application platform interoperability objective. Transfer of stored data is an important method for application software entities to exchange and make mutual use of data.

3.1.3 Application Software Interoperability and Application Platform Interoperability

The POSIX OSE should enable application software and application platform interoperability.

Rationale: Communication services and format specifications allow two entities participating in a distributed system to exchange and make mutual use of data, including

- 76 — Homogeneous systems
- 77 — Heterogeneous systems (i.e., a wide variety of hardware/software
- 78 platforms)
- 79 — POSIX OSE-based and non-POSIX OSE-based systems

80 3.1.4 User Portability

81 The POSIX OSE should enable human users to operate on a wide range of applica-
82 tion platform implementations without retraining.

83 *Rationale:* Standard methods and services for supporting human/computer
84 interaction are a key aspect of the definition of an open system (see 2.2.2.28).
85 Elimination of gratuitous differences in the interface that the application plat-
86 form presents to the user via standards is a significant aspect of this task.

87 3.1.5 Accommodation of Standards

88 The POSIX OSE should accommodate existing, imminent, and new standards that
89 provide interfaces, services, and formats for open systems.

90 *Rationale:* In order to accommodate standards, the POSIX OSE has to define a
91 clear set of interfaces, upon which portability and interoperability are defined.
92 The definition of these interfaces needs to be driven by user requirements and be
93 as independent of the implementation technology as possible.

94 If the POSIX OSE were constrained to current standards and technology, it would
95 quickly become obsolete. It would also not be capable of providing a complete set
96 of applicable standards and profiles, as efforts to date have not yet provided a full
97 suite of applicable standards. The POSIX OSE will evolve as standards emerge
98 and technology changes.

99 3.1.6 Accommodation of New Technology

100 The POSIX OSE should accommodate new information system technology.

101 *Rationale:* POSIX OSE standards specify interfaces, rather than implementations.
102 However, where the interface needs to refer to specific technology, an element of
103 judgment is required, due to the tension between the conflicting needs for stable
104 standards and provision for technology enhancement. The POSIX OSE should be
105 sufficiently general to allow for technology growth and variety, and yet specific
106 enough to act as a guide for standards development.

107 3.1.7 Application Platform Scalability

108 The POSIX OSE should be scalable to allow inclusion of platforms of varying
109 speed, throughput, and implementation architecture (e.g., multiprocessing).

110 *Rationale:* This reflects the realities faced by the potential users of the POSIX
111 OSE. This objective affects individual standards as well as the conditions under

which various standards can or should be combined into profiles. For example, where similar services are provided by both workstation-type and supercomputer-type application platforms, the same standards should be applied to each if possible. This would enable a greater degree of portability across these specialized implementations of the application platform.

3.1.8 Distributed System Scalability

The POSIX OSE should provide for distributed system scalability.

Rationale: The number of distributed system components connected should not be limited by any structural aspects of the POSIX OSE. For example, in the area of network services, the OSE standards should be such that it is possible to construct profiles (and therefore systems) in which remote and local operation and utilization of information system resources are indistinguishable, with the exception of unavoidable message transit delay. In other words, it should be possible for applications to be unaware of whether the application platform on which they are executing is local or distributed and that lack of awareness should not affect their proper operation.

3.1.9 Implementation Transparency

The POSIX OSE should provide consistent and standard interfaces to the application regardless of the underlying implementation technology.

Rationale: The mechanism for implementation of services is not visible to the service user; i.e., only the service is visible to the service user. The POSIX OSE calls for interfaces that are specified in such a way as to be independent of the implementation technology. The interface specifications are the complete definition of the application platform. The specifications should be complete enough that one does not need to see the source code to program.

3.1.10 User Functional Requirements

The POSIX OSE should reflect the full scope of the functional requirements of the user, within the context of the preceding objectives.

Rationale: The POSIX OSE will provide the context within which application software portability and interoperability requirements objectives are satisfied. The user of the information system is the final authority with respect to completeness of the services required to achieve full portability. In particular, the range of services required is broader than those traditionally identified as operating system services.

3.2 POSIX OSE Reference Model

The POSIX OSE is based on a reference model with the full (potentially distributed) information system as its scope. As such, it spans the gap between requirement specification and the design of any specific system that uses information technology. The reference model provides a set of conventions and concepts, mutually agreed upon between the information system user and provider communities. This common understanding is key to achieving application software portability and system interoperability, and it may encourage software reuse. It may lead to more compact and correct procurement specifications.

An information system reference model addresses conflicting requirements similar to those encountered in traditional architectural disciplines. The reference model has to be structured enough to encourage the generation and use of standards and standard components, yet it also has to be flexible enough to accommodate tailored and special purpose components necessary to meet real user requirements.

The POSIX OSE reference model is a set of concepts, interfaces, entities, and diagrams that provides a basis for specification of standards. It provides guidance and direction for future standardization and integration efforts. In order for the POSIX OSE to evolve and mature, it is necessary for the reference model to provide insights into those services and capabilities for which standards do not currently exist and for which appropriate standardization activities cannot be identified.

The POSIX OSE reference model is described from the user perspective; i.e., the reference model records the perception of the application platform user (mental model) of the overall large distributed system used to support the user enterprise. This point of view assures that

- Information technology users will have the proper services to meet their requirements
- Information technology vendor implementations will not be constrained unnecessarily

Figure 3-1 depicts the POSIX OSE reference model. The model identifies three entities (Application Software, Application Platform, and External Environment) and two interfaces between them, identified as the API and the EEI. The application platform provides API and EEI services across the associated interfaces.

This model has been generalized to such a degree that it can accommodate a wide variety of general- and special-purpose systems. A more detailed discussion of each service category is provided in Section 4. The service specification has been defined to be flexible enough to allow subsets or extensions for each category as needed. As a result, the POSIX OSE reference model is able to accommodate a variety of architectures and standardization approaches. It should be possible to show where any relevant standard fits within the reference model.

For the purposes of this guide, the standards discussed address only interfaces between entities, including services and supporting formats offered across those interfaces. The interface specification defines a convention adopted to represent the function offered across the interface in both directions. Note that no set of standards can, by itself, assure portability of specific applications. Applications

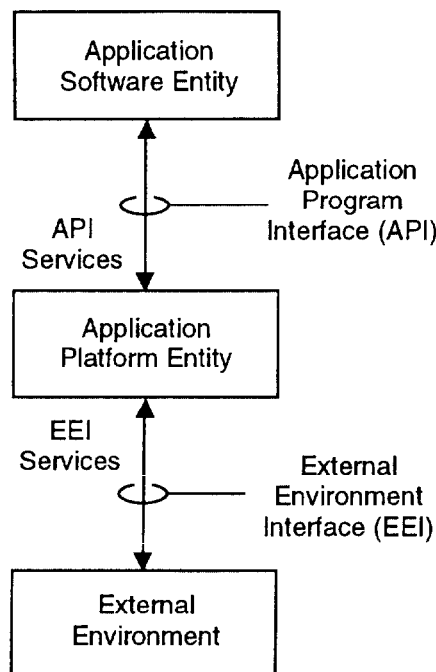


Figure 3-1 – POSIX OSE Reference Model

need to be properly engineered with an explicit portability objective in order to achieve it.

The reference model is not a layered model, although the graphical representation may give that impression.⁹⁾ It is the definitions of, and relationships among, the parts of the model that are important, rather than the graphical representation. Each of the parts interacts with the others across the interfaces, and no other relationship (for example dependence, importance, superiority, or primacy) is implied or intended.

The application platform provides services to a variety of users across both platform interfaces. The EEI is actually composed of three types of interfaces, all of which share the common characteristic of being externally visible. A more complete discussion of EEIs is found in 3.2.2.1. A human user invokes the platform services at the EEI. A programmer invokes application platform services at the API by writing source code, which accesses the service when compiled and executed.

All of these features may be available locally or remotely if the system is connected to a larger distributed system. All other resources can be conceptualized as being contained within the application platform.

⁹⁾ In fact, many other representations were considered (for example, various rotations, circular symmetry, etc.).

Note that the actual implementation of any given system element may differ greatly from the reference model presented. The intention is to define a conceptual reference model that the widespread design, implementation, and integration communities may assume in executing their activities. Partitioning of function for purposes of discussion or specification does not imply or endorse similar partitioning for design or implementation.

3.2.1 Reference Model Entities and Elements

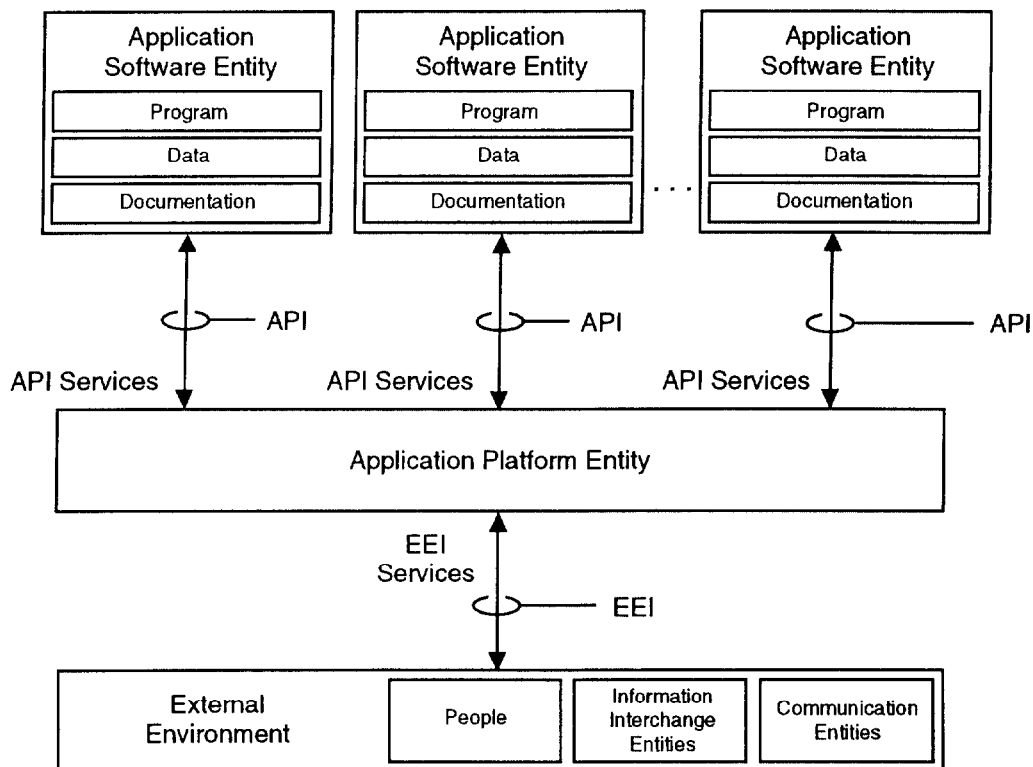


Figure 3-2 – POSIX OSE Reference Model — Entities

Figure 3-2 expands Figure 3-1 to identify elements of the reference model entities. For the purposes of this discussion, the term *entities* is used when discussing the classification of items related to application portability. The term *entity* is used in the dictionary sense, with no hierarchy implied. The only distinction needed in this discussion is the distinction between a *thing* (or entity), and an *interface* (or boundary between things).

Application software is defined (see 2.2.2.6) as software specific to an application. It is composed of one or more of

- Programs (source code, command/script files, etc.)
- Data (user data, application parameters, screen definitions, etc.)

— Documentation (online documentation only; hardcopy not included)

An application program is represented by source code, produced according to one or more specific programming languages and a set of language bindings (i.e., API specifications) for the required services. These specifications may be standards, public specifications (see 2.2.2.42), or proprietary specifications.

An application program may be divided into two parts:

— An *invariant* portion of source code, requiring no change when ported

— A *variant* portion of source code, which requires changes when ported

The objective of any effective application software portability method or project should be to minimize the variant portion of the application software via creation and use of API standards. This would ideally allow application software components to be moved to a different (but standards conforming) application platform and run without source code modification.

Separate, but related, standards may be required to support the portability of each of the elements listed above. Examples of application software are the familiar word-processing, spreadsheet, or accounting packages, as developed by the consumer or a commercial application software developer. Each of these packages appears as an application software entity when executed on an application platform.

One or more applications may run on a given application platform simultaneously, as represented by the boxes at the top of Figure 3-2. Each application can be thought of as an independent application entity, communicating and synchronizing with other applications, if necessary, via a variety of communication mechanisms.

The application platform is defined (see 2.2.2.4) as the set of resources that support the services on which an application or application software will run. It provides services at its interfaces that, as much as possible, make the implementation-specific characteristics of the platform transparent to the application software.

In order to assure system integrity and consistency, application software entities competing for application platform resources need to access all resources via service requests across the API.

The application platform concept does not imply or constrain any specific implementation beyond the basic requirement to supply services at the interfaces. For example, the platform might be a single processor shared by a group of applications, or it might be a large distributed system with each application dedicated to a single processor. (See 3.2.4.)

The application platform implementations that use the POSIX OSE may differ greatly depending upon the requirements of each system and its intended use. It is expected that application platforms defined to be consistent with the POSIX OSE will not necessarily provide all the features discussed here, but will use tailored subsets for a particular set of application software.

The external environment contains the external entities with which the application platform exchanges information. These entities are classified into the

general categories of human users, information interchange entities, and communication entities.

Human users are not further classified, but are treated as typical persons. Information interchange entities are physical data storage media, including, for example, CD-ROM, magnetic tapes, floppy disks, and security badges. Communication entities include phone lines, local area networks, and packet switching equipment.

3.2.2 Reference Model Interfaces

Figure 3-3 expands on Figure 3-1 to identify categories of services available at the reference model interfaces. Between the general model entities there are two types of interfaces, labeled as the API and the EEI.

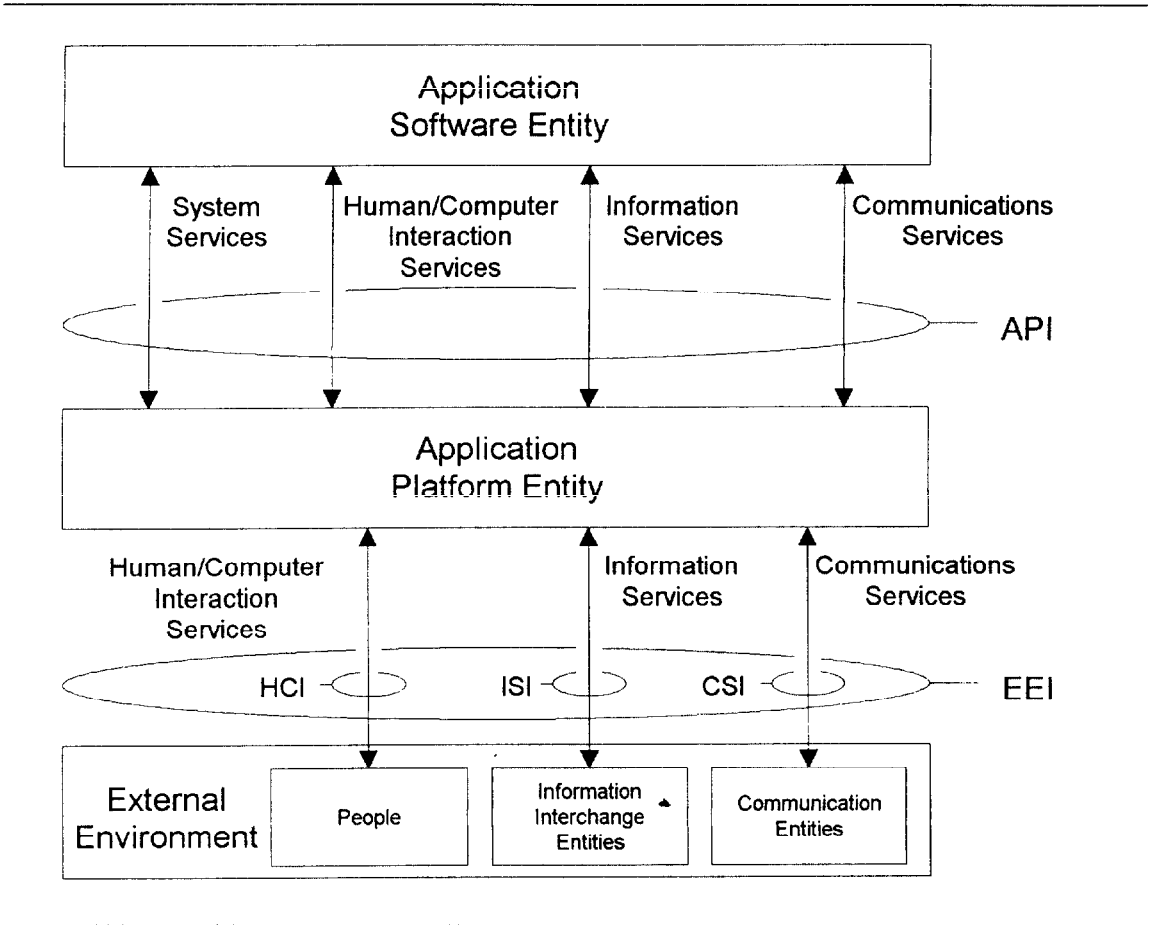


Figure 3-3 – POSIX OSE Reference Model — Interfaces

3.2.2.1 EEI

The EEI is defined (see 2.2.2.14) as the interface between the application platform and the external environment across which information is exchanged. It is defined primarily in support of system and application software interoperability.

User and data portability are directly provided by the EEI, but application software portability is indirectly supported by reference to common concepts linking specifications at both interfaces.

An example of a common concept linking both the human/computer interaction API and the human/computer interaction EEI is a “window.” At the API, a hypothetical service

```
Open_Window(x1, y1, x2, y2)
```

could be specified to draw a rectangular region on a display, with its upper left corner at screen coordinates (x1, y1) and its lower right corner at screen coordinates (x2, y2). At the EEI, conventions for the appearance and behavior of a window may be documented independently of the API. For instance, a convention may be adopted that all windows will have borders and “handles” on the corners that a user can use to resize the window.

Changes to the specifications for the API and EEI can be made independently to some degree, but the concept of a window is fundamental to both. To achieve portability for these services, a common concept for a window has to be adopted across the set of platforms for which portability is desired. Other common concepts are associated with each of the external interfaces.

The following three interfaces are EEI type interfaces:

- HCI
- ISI
- CSI

The HCI is the boundary across which physical interaction between the human user and the application platform takes place. Examples of this type of interface include CRT displays, keyboards, mice (or other pointing devices), and audio I/O devices. Standardization at this interface will allow users to access the services of POSIX OSE-based systems without costly retraining.

The ISI defines a boundary across which external, persistent storage service is provided, where only the format and syntax are required to be specified for data portability and interoperability.

The CSI provides access to services for interaction between internal application software entities and application platform external entities, such as application software entities on other application platforms, external data transport facilities, and devices. The services provided are those where protocol state, syntax, and format all should be standardized for application interoperability.

3.2.2.2 API

The API is defined (see 2.2.2.5) as the interface between the application software and the application platform across which all services are provided. The API is defined primarily in support of application portability, but system and application software interoperability also are supported via the communication services and information services APIs.

The POSIX OSE API is a combination of a number of standards-based interfaces. It can be thought of as a bookshelf containing several standards-based APIs, with each API a separate book on the bookshelf.

The POSIX OSE API describes a complete set of services provided at the interface between the application software and the underlying application platform. The services provided at the API may be divided into the following categories:

- System Services, including core system services and language services
- Communication Services
- Information Services, including database services, data interchange services, and transaction processing services
- Human/Computer Interaction Services, including user command interface services, character-based user interface services, windowing system services, graphics services, and application software development support services

The first API grouping provides access to services associated with resources that are internal to the application platform. The last three API groupings provide the application software with access to services associated with external environment entities.

These four categories of API services describe the range of function needed at the API. The specifications (that is, the documents themselves) may take the form of programming language specifications, language-independent service specifications, and language bindings for the service specifications. These specifications may be described as follows:

- Specifications traditionally associated with languages, such as program control (if ... then ... else), math functions, string manipulation, concurrency, low-level programming mechanisms, etc., are defined as *the programming language API specifications*.
- Specifications for services provided by the underlying application platform defined independent of any programming language, such as interprocess communications, interobject messages, access to the user interface, and data storage, are identified as *language-independent service specifications*.
- Language-independent service specifications are translated into language-specific specifications used by programmers in writing applications. These specifications provide access to the services using methods consistent with a specific programming language. Such language-specific specifications are called *language-binding API specifications*.

While it is acknowledged that language-independent specifications are not yet common, their creation is expected to facilitate the management and development of consistent language-binding standards. The language-binding specifications are used directly by programmers and application platform suppliers in implementing application software and platforms.

The “programming language”/“language binding” dichotomy may be a result of the way information technology standards are currently developed. Programming language specifications are developed with the goal of being “system independent”

(e.g., C, COBOL, Fortran, etc.). Language-binding specifications (e.g., ISO/IEC 9945-1:1990 [68], MOSI, GKS, SQL, etc.) are being translated into “language-independent” specifications, with one or more bindings for specific languages. Revisiting the “bookshelf” concept described at the beginning of the subclause, the following question can now be answered: “What books (standards) does the programmer need to produce application software source code that is fully portable?” Once the programmer decides (or is told) which language to use, the programmer needs the programming language manual (e.g., C or Ada) and the language-binding specifications that provide all the required services (e.g., C bindings for graphics, networking, and indexed files). In this context, it is clear that the base language specification itself is a major component of the API. In fact, many useful programs can be written using only the base language specifications.

3.2.3 EEI-API Service Relationships

The relationships between services provided at the API and similarly named services at the EEI are not simple one-to-one relationships. For example, a data storage service interface may provide an application with transparent access to a remote file via network services. In this case, the completion of the data storage service provided at the API is dependent upon (“translated” into) communication services provided at the EEI.

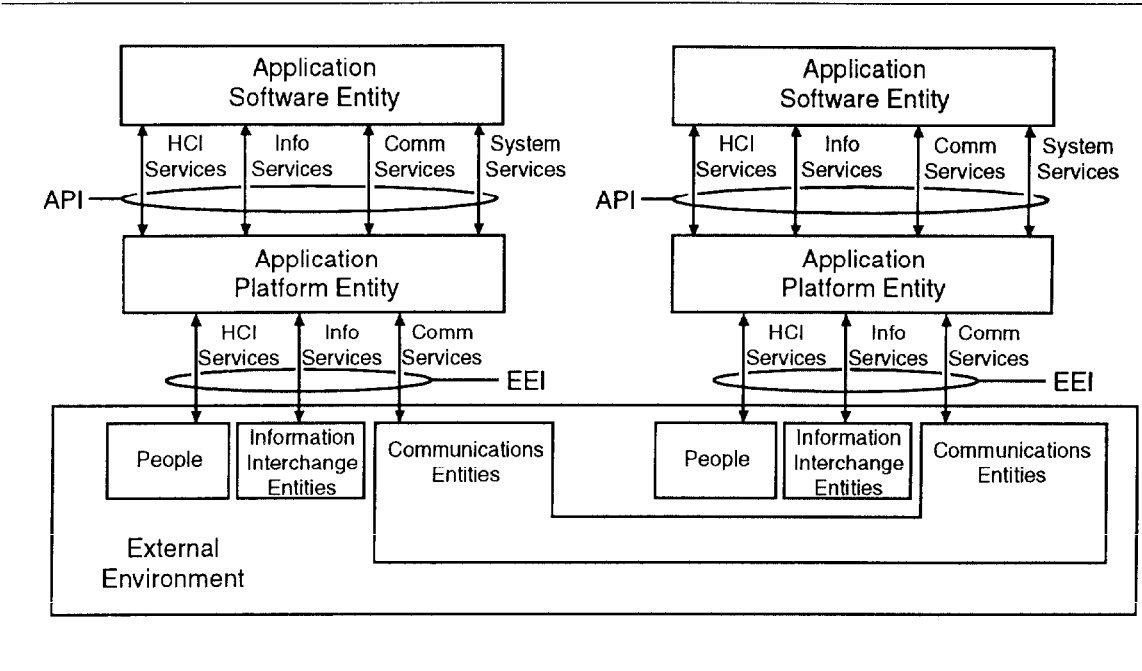
In the general case, application software entities never access the EEI directly, although service requests at the API will often result in service invocations at the EEI. This usually happens using mechanisms that the application programmer never observes or cares about, as long as the service request is satisfied. Likewise, the information services that are accessible at the API may sometimes be performed internally, without invoking an EEI function.

Fortunately, it is not essential for the purpose of satisfying the requirements of the POSIX OSE to specify these relationships in detail. In fact, too detailed a definition could unnecessarily constrain the implementation. A given implementation of the application platform will define the relationship between the API and EEI in different ways.

3.2.4 POSIX OSE-Based Distributed Systems

In a distributed environment, multiple application platforms may interact by way of a communication mechanism external to the platforms. Application platforms interact with the communication services EEI, as in Figure 3-4. When an application software entity requests communication with another entity on a different platform, the request is made at the API. The implementation of the application platform translates these API requests into appropriate actions at the EEI.

Communication occurs between application platforms via external entities that implement the data transport function at the communications services EEI. These can use a wide variety of implementation methods and protocols, providing access to distributed data and services via a network.



417 **Figure 3-4 – POSIX OSE Reference Model — Distributed Systems**

418 This treatment of the communications services EEI is also used to support the
 419 implementation transparency of the application platform. Figure 3-5 sketches an
 420 application platform implementation where the platform is actually a distributed
 421 system composed of multiple platforms. In this scenario, the application platform
 422 is labeled as a “perceived” platform. This distinction is important, for example,
 423 during procurement. The procuring authority specifies requirements of a per-
 424 ceived platform, and the bidders respond with proposals that describe alternative
 425 implementations of the platform. One proposal could implement the platform
 426 using a single monolithic processor, such as a classic mainframe. Another propo-
 427 sal might provide a variety of specialized processors connected by a network. A
 428 wide variety of alternatives is possible, each with its own advantages and disad-
 429 vantages. By specifying the procurement requirements as a perceived platform,
 430 the buyer can consider the widest range of alternative implementations and more
 431 easily tailor the solution to fit the needs of the buyer.

432 Note that the treatment of distributed systems within this guide is limited to
 433 those aspects directly related to application portability and interoperability. A
 434 complete treatment of distributed systems would require a considerably expanded
 435 discussion, which is beyond the scope of this guide.

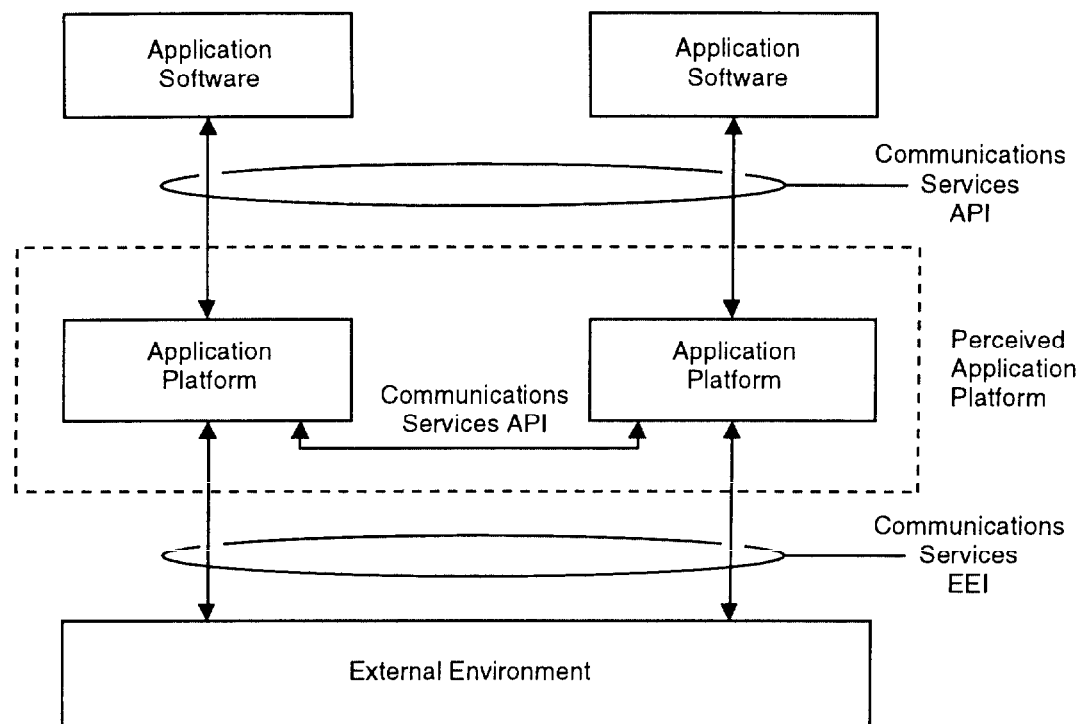


Figure 3-5 – Distributed Application Platform Implementation

3.3 POSIX OSE Services

Applications may use services that are distributed among a number of different computers. This implies that one application may use many POSIX services. This approach becomes even more valuable when a distributed environment has to be designed to support many applications using a single resource like a database. Note that the database itself may be implemented across multiple machines. Similarly, the notion of cooperative work will eventually require support of many users accessing a single application that arbitrates both screen and database usage.

This guide describes services provided to users of application platforms in support of POSIX objectives of application portability and system interoperability. These services are available to users across specified interfaces keyed to the POSIX OSE reference model defined in 3.2.

The POSIX OSE services are divided into categories described by the clauses in Section 4. Each category begins by defining a more detailed and specialized version of the POSIX OSE reference model to provide context for service specification. Services and associated standards are then described for each category. Finally, POSIX OSE cross-category services affecting each category are discussed.

Further refinement, based on evolving standards and technology, will lead to further evolution of this guide.