

1003.1TM
Standard for Information Technology —
Portable Operating System Interface (POSIXTM)
Base Definitions, Issue 6

Approved 12 September 2001
The Open Group

IEEE Sponsor
Portable Applications Standards Committee
of the
IEEE Computer Society

Approved 6 December 2001
IEEE-SA Standards Board



THE *Open* GROUP

Abstract

This standard defines a standard operating system interface and environment, including a command interpreter (or “shell”), and common utility programs to support applications portability at the source code level. It is the single common revision to IEEE Std 1003.1-1996, IEEE Std 1003.2-1992, and the Base Specifications of The Open Group Single UNIX[®] † Specification, Version 2. This standard is intended to be used by both applications developers and system implementors and comprises four major components (each in an associated volume):

- General terms, concepts, and interfaces common to all volumes of this standard, including utility conventions and C-language header definitions, are included in the Base Definitions volume.
- Definitions for system service functions and subroutines, language-specific system services for the C programming language, function issues, including portability, error handling, and error recovery, are included in the System Interfaces volume.
- Definitions for a standard source code-level interface to command interpretation services (a “shell”) and common utility programs for application programs are included in the Shell and Utilities volume.
- Extended rationale that did not fit well into the rest of the document structure, containing historical information concerning the contents of this standard and why features were included or discarded by the standard developers, is included in the Rationale (Informative) volume.

The following areas are outside the scope of this standard:

- Graphics interfaces
- Database management system interfaces
- Record I/O considerations
- Object or binary code portability
- System configuration and resource availability

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

Keywords

application program interface (API), argument, asynchronous, basic regular expression (BRE), batch job, batch system, built-in utility, byte, child, command language interpreter, CPU, extended regular expression (ERE), FIFO, file access control mechanism, input/output (I/O), job control, network, portable operating system interface (POSIX[™] †), parent, shell, stream, string, synchronous, system, thread, X/Open System Interface (XSI)

† See **Trademarks** (on page xxxv).

All rights reserved. No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission from both the IEEE and The Open Group.

Portions of this standard are derived with permission from copyrighted material owned by Hewlett-Packard Company, International Business Machines Corporation, Novell Inc., The Open Software Foundation, and Sun Microsystems, Inc.

Permissions

Authorization to photocopy portions of this standard for internal or personal use is granted provided that the appropriate fee is paid to the Copyright Clearance Center or the equivalent body outside of the U.S. Permission to make multiple copies for educational purposes in the U.S. requires agreement and a license fee to be paid to the Copyright Clearance Center.

Beyond these provisions, permission to reproduce all or any part of this standard must be with the consent of both copyright holders and may be subject to a license fee. Both copyright holders will need to be satisfied that the other has granted permission. Requests to the copyright holders should be sent by email to austin-group-permissions@opengroup.org.

Feedback

This standard has been prepared by the Austin Group. Feedback relating to the material contained in this standard may be submitted using the Austin Group web site at <http://www.opengroup.org/austin/defectform.html>.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property, or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "AS IS".

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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

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 the IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of 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 for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with the IEEE.¹ Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, U.S.A.

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 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.

A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and non-discriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

The IEEE and its designees are the sole entities that may authorize the use of IEEE-owned certification marks and/or trademarks to indicate compliance with the materials set forth herein. Authorization to photocopy portions of any individual standard for internal or personal use is granted in the U.S. by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to the Copyright Clearance Center.² Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center. To arrange for payment of the licensing fee, please contact:

Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923, U.S.A., Tel.: +1 978 750 8400

Amendments, corrigenda, and interpretations for this standard, or information about the IEEE standards development process, may be found at <http://standards.ieee.org>.

Full catalog and ordering information on all IEEE publications is available from the IEEE Online Catalog & Store at <http://shop.ieee.org/store>.

1. For this standard, please send comments via the Austin Group as requested on page iii.

2. Please refer to the special provisions for this standard on page iii concerning permissions from both copyright holders and arrangements to cover photocopying and reproduction across the world, as well as by commercial organizations wishing to license the material for use in product documentation.

The Open Group

The Open Group, a vendor and technology-neutral consortium, is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost, and risks associated with integrating new technology across the enterprise.

The Open Group's mission is to offer all organizations concerned with open information infrastructures a forum to share knowledge, integrate open initiatives, and certify approved products and processes in a manner in which they continue to trust our impartiality.

In the global eCommerce world of today, no single economic entity can achieve independence while still ensuring interoperability. The assurance that products will interoperate with each other across differing systems and platforms is essential to the success of eCommerce and business workflow. The Open Group, with its proven testing and certification program, is the international guarantor of interoperability in the new century.

The Open Group provides opportunities to exchange information and shape the future of IT. The Open Group's members include some of the largest and most influential organizations in the world. The flexible structure of The Open Groups membership allows for almost any organization, no matter what their size, to join and have a voice in shaping the future of the IT world.

More information is available on The Open Group web site at <http://www.opengroup.org>.

The Open Group has over 15 years' experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification. The Open Group portfolio of test suites includes the *Westwood* family of tests for this standard and the associated certification program for Version 3 of the Single UNIX Specification, as well tests for CDE, CORBA, Motif, Linux, LDAP, POSIX.1, POSIX.2, POSIX Realtime, Sockets, UNIX, XPG4, XNFS, XTI, and X11. The Open Group test tools are essential for proper development and maintenance of standards-based products, ensuring conformance of products to industry-standard APIs, applications portability, and interoperability. In-depth testing identifies defects at the earliest possible point in the development cycle, saving costs in development and quality assurance.

More information is available at <http://www.opengroup.org/testing>.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards and Guides, but which also includes white papers, technical studies, branding and testing documentation, and business titles. Full details and a catalog are available at <http://www.opengroup.org/pubs>.

As with all *live* documents, Technical Standards and Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards compatible and those which are not:

- A new *Version* indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it *replaces* the previous publication.
- A new *Issue* indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Readers should note that Corrigenda may apply to any publication. Corrigenda information is published at <http://www.opengroup.org/corrigenda>.

Full catalog and ordering information on all Open Group publications is available at <http://www.opengroup.org/pubs>.

Contents

Chapter 1	Introduction.....	1
1.1	Scope.....	1
1.2	Conformance	4
1.3	Normative References	4
1.4	Terminology	5
1.5	Portability	6
1.5.1	Codes	6
1.5.2	Margin Code Notation.....	14
 Chapter 2	 Conformance.....	 15
2.1	Implementation Conformance.....	15
2.1.1	Requirements.....	15
2.1.2	Documentation.....	15
2.1.3	POSIX Conformance	16
2.1.3.1	POSIX System Interfaces.....	16
2.1.3.2	POSIX Shell and Utilities.....	18
2.1.4	XSI Conformance	19
2.1.4.1	XSI System Interfaces.....	19
2.1.4.2	XSI Shell and Utilities Conformance	20
2.1.5	Option Groups.....	20
2.1.5.1	Subprofiling Considerations.....	20
2.1.5.2	XSI Option Groups	22
2.1.6	Options.....	26
2.1.6.1	System Interfaces	27
2.1.6.2	Shell and Utilities.....	27
2.2	Application Conformance.....	29
2.2.1	Strictly Conforming POSIX Application.....	29
2.2.2	Conforming POSIX Application.....	30
2.2.2.1	ISO/IEC Conforming POSIX Application.....	30
2.2.2.2	<National Body> Conforming POSIX Application	30
2.2.3	Conforming POSIX Application Using Extensions	30
2.2.4	Strictly Conforming XSI Application	30
2.2.5	Conforming XSI Application Using Extensions.....	31
2.3	Language-Dependent Services for the C Programming Language..	31
2.4	Other Language-Related Specifications	31
 Chapter 3	 Definitions.....	 33
3.1	Abortive Release	33
3.2	Absolute Pathname	33
3.3	Access Mode	33
3.4	Additional File Access Control Mechanism.....	33
3.5	Address Space	33

3.6	Advisory Information	33
3.7	Affirmative Response	34
3.8	Alert	34
3.9	Alert Character (<alert>)	34
3.10	Alias Name	34
3.11	Alignment	34
3.12	Alternate File Access Control Mechanism	34
3.13	Alternate Signal Stack	35
3.14	Ancillary Data	35
3.15	Angle Brackets	35
3.16	Application	35
3.17	Application Address	35
3.18	Application Program Interface (API)	35
3.19	Appropriate Privileges	35
3.20	Argument	36
3.21	Arm (a Timer)	36
3.22	Asterisk	36
3.23	Async-Cancel-Safe Function	36
3.24	Asynchronous Events	36
3.25	Asynchronous Input and Output	36
3.26	Async-Signal-Safe Function	36
3.27	Asynchronously-Generated Signal	37
3.28	Asynchronous I/O Completion	37
3.29	Asynchronous I/O Operation	37
3.30	Authentication	37
3.31	Authorization	37
3.32	Background Job	37
3.33	Background Process	37
3.34	Background Process Group (or Background Job)	37
3.35	Backquote	38
3.36	Backslash	38
3.37	Backspace Character (<backspace>)	38
3.38	Barrier	38
3.39	Base Character	38
3.40	Basename	38
3.41	Basic Regular Expression (BRE)	38
3.42	Batch Access List	38
3.43	Batch Administrator	39
3.44	Batch Client	39
3.45	Batch Destination	39
3.46	Batch Destination Identifier	39
3.47	Batch Directive	39
3.48	Batch Job	39
3.49	Batch Job Attribute	40
3.50	Batch Job Identifier	40
3.51	Batch Job Name	40
3.52	Batch Job Owner	40
3.53	Batch Job Priority	40

3.54	Batch Job State	40
3.55	Batch Name Service.....	40
3.56	Batch Name Space	40
3.57	Batch Node	41
3.58	Batch Operator	41
3.59	Batch Queue.....	41
3.60	Batch Queue Attribute.....	41
3.61	Batch Queue Position.....	41
3.62	Batch Queue Priority.....	41
3.63	Batch Rerunability	41
3.64	Batch Restart	42
3.65	Batch Server	42
3.66	Batch Server Name	42
3.67	Batch Service	42
3.68	Batch Service Request	42
3.69	Batch Submission.....	42
3.70	Batch System.....	42
3.71	Batch Target User.....	43
3.72	Batch User.....	43
3.73	Bind.....	43
3.74	Blank Character (<blank>).....	43
3.75	Blank Line	43
3.76	Blocked Process (or Thread)	43
3.77	Blocking	43
3.78	Block-Mode Terminal.....	43
3.79	Block Special File.....	44
3.80	Braces.....	44
3.81	Brackets.....	44
3.82	Broadcast	44
3.83	Built-In Utility (or Built-In).....	44
3.84	Byte	44
3.85	Byte Input/Output Functions	45
3.86	Carriage-Return Character (<carriage-return>).....	45
3.87	Character	45
3.88	Character Array.....	45
3.89	Character Class.....	45
3.90	Character Set.....	45
3.91	Character Special File.....	46
3.92	Character String	46
3.93	Child Process	46
3.94	Circumflex.....	46
3.95	Clock.....	46
3.96	Clock Jump.....	46
3.97	Clock Tick.....	46
3.98	Coded Character Set.....	46
3.99	Codeset.....	47
3.100	Collating Element	47
3.101	Collation	47

3.102	Collation Sequence	47
3.103	Column Position	47
3.104	Command	48
3.105	Command Language Interpreter	48
3.106	Composite Graphic Symbol	48
3.107	Condition Variable	48
3.108	Connection	48
3.109	Connection Mode	48
3.110	Connectionless Mode	48
3.111	Control Character	49
3.112	Control Operator	49
3.113	Controlling Process	49
3.114	Controlling Terminal	49
3.115	Conversion Descriptor	49
3.116	Core File	49
3.117	CPU Time (Execution Time)	49
3.118	CPU-Time Clock	50
3.119	CPU-Time Timer	50
3.120	Current Job	50
3.121	Current Working Directory	50
3.122	Cursor Position	50
3.123	Datagram	50
3.124	Data Segment	50
3.125	Deferred Batch Service	50
3.126	Device	50
3.127	Device ID	50
3.128	Directory	51
3.129	Directory Entry (or Link)	51
3.130	Directory Stream	51
3.131	Disarm (a Timer)	51
3.132	Display	51
3.133	Display Line	51
3.134	Dollar Sign	51
3.135	Dot	51
3.136	Dot-Dot	52
3.137	Double-Quote	52
3.138	Downshifting	52
3.139	Driver	52
3.140	Effective Group ID	52
3.141	Effective User ID	52
3.142	Eight-Bit Transparency	52
3.143	Empty Directory	52
3.144	Empty Line	53
3.145	Empty String (or Null String)	53
3.146	Empty Wide-Character String	53
3.147	Encoding Rule	53
3.148	Entire Regular Expression	53
3.149	Epoch	53

3.150	Equivalence Class	53
3.151	Era	53
3.152	Event Management.....	54
3.153	Executable File.....	54
3.154	Execute	54
3.155	Execution Time.....	54
3.156	Execution Time Monitoring.....	54
3.157	Expand	54
3.158	Extended Regular Expression (ERE).....	54
3.159	Extended Security Controls	55
3.160	Feature Test Macro	55
3.161	Field	55
3.162	FIFO Special File (or FIFO)	55
3.163	File	55
3.164	File Description	56
3.165	File Descriptor	56
3.166	File Group Class.....	56
3.167	File Mode	56
3.168	File Mode Bits	56
3.169	Filename.....	56
3.170	Filename Portability	56
3.171	File Offset.....	57
3.172	File Other Class	57
3.173	File Owner Class	57
3.174	File Permission Bits.....	57
3.175	File Serial Number	57
3.176	File System	57
3.177	File Type.....	57
3.178	Filter	58
3.179	First Open (of a File).....	58
3.180	Flow Control	58
3.181	Foreground Job.....	58
3.182	Foreground Process	58
3.183	Foreground Process Group (or Foreground Job).....	58
3.184	Foreground Process Group ID	58
3.185	Form-Feed Character (<form-feed>)	58
3.186	Graphic Character.....	59
3.187	Group Database	59
3.188	Group ID.....	59
3.189	Group Name	59
3.190	Hard Limit.....	59
3.191	Hard Link	59
3.192	Home Directory.....	59
3.193	Host Byte Order	60
3.194	Incomplete Line.....	60
3.195	Inf	60
3.196	Instrumented Application.....	60
3.197	Interactive Shell.....	60

3.198	Internationalization.....	60
3.199	Interprocess Communication.....	60
3.200	Invoke.....	60
3.201	Job.....	61
3.202	Job Control	61
3.203	Job Control Job ID.....	61
3.204	Last Close (of a File)	61
3.205	Line	61
3.206	Linger.....	61
3.207	Link	61
3.208	Link Count.....	62
3.209	Local Customs	62
3.210	Local Interprocess Communication (Local IPC)	62
3.211	Locale.....	62
3.212	Localization.....	62
3.213	Login.....	62
3.214	Login Name.....	62
3.215	Map	62
3.216	Marked Message	63
3.217	Matched	63
3.218	Memory Mapped Files.....	63
3.219	Memory Object.....	63
3.220	Memory-Resident	63
3.221	Message.....	63
3.222	Message Catalog	64
3.223	Message Catalog Descriptor.....	64
3.224	Message Queue	64
3.225	Mode.....	64
3.226	Monotonic Clock.....	64
3.227	Mount Point	64
3.228	Multi-Character Collating Element.....	64
3.229	Mutex.....	64
3.230	Name	65
3.231	Named STREAM.....	65
3.232	NaN (Not a Number).....	65
3.233	Native Language.....	65
3.234	Negative Response	65
3.235	Network.....	65
3.236	Network Address.....	65
3.237	Network Byte Order.....	66
3.238	Newline Character (<newline>)	66
3.239	Nice Value	66
3.240	Non-Blocking.....	66
3.241	Non-Spacing Characters	66
3.242	NUL	66
3.243	Null Byte.....	67
3.244	Null Pointer.....	67
3.245	Null String.....	67

3.246	Null Wide-Character Code	67
3.247	Number Sign	67
3.248	Object File	67
3.249	Octet	67
3.250	Offset Maximum	67
3.251	Opaque Address	67
3.252	Open File	68
3.253	Open File Description	68
3.254	Operand	68
3.255	Operator	68
3.256	Option	68
3.257	Option-Argument	68
3.258	Orientation	68
3.259	Orphaned Process Group	68
3.260	Page	69
3.261	Page Size	69
3.262	Parameter	69
3.263	Parent Directory	69
3.264	Parent Process	69
3.265	Parent Process ID	69
3.266	Pathname	70
3.267	Pathname Component	70
3.268	Path Prefix	70
3.269	Pattern	70
3.270	Period	70
3.271	Permissions	70
3.272	Persistence	70
3.273	Pipe	71
3.274	Polling	71
3.275	Portable Character Set	71
3.276	Portable Filename Character Set	71
3.277	Positional Parameter	71
3.278	Preallocation	71
3.279	Preempted Process (or Thread)	72
3.280	Previous Job	72
3.281	Printable Character	72
3.282	Printable File	72
3.283	Priority	72
3.284	Priority Band	72
3.285	Priority Inversion	72
3.286	Priority Scheduling	72
3.287	Priority-Based Scheduling	73
3.288	Privilege	73
3.289	Process	73
3.290	Process Group	73
3.291	Process Group ID	73
3.292	Process Group Leader	73
3.293	Process Group Lifetime	73

3.294	Process ID	74
3.295	Process Lifetime	74
3.296	Process Memory Locking	74
3.297	Process Termination	74
3.298	Process-To-Process Communication	74
3.299	Process Virtual Time	74
3.300	Program.....	75
3.301	Protocol	75
3.302	Pseudo-Terminal	75
3.303	Radix Character.....	75
3.304	Read-Only File System	75
3.305	Read-Write Lock.....	75
3.306	Real Group ID.....	75
3.307	Real Time.....	76
3.308	Realtime Signal Extension.....	76
3.309	Real User ID	76
3.310	Record	76
3.311	Redirection	76
3.312	Redirection Operator	76
3.313	Reentrant Function	76
3.314	Referenced Shared Memory Object	76
3.315	Refresh.....	77
3.316	Regular Expression.....	77
3.317	Region.....	77
3.318	Regular File	77
3.319	Relative Pathname	77
3.320	Relocatable File.....	77
3.321	Relocation	77
3.322	Requested Batch Service.....	77
3.323	(Time) Resolution	77
3.324	Root Directory	78
3.325	Runnable Process (or Thread)	78
3.326	Running Process (or Thread).....	78
3.327	Saved Resource Limits.....	78
3.328	Saved Set-Group-ID	78
3.329	Saved Set-User-ID.....	78
3.330	Scheduling.....	78
3.331	Scheduling Allocation Domain.....	78
3.332	Scheduling Contention Scope	79
3.333	Scheduling Policy.....	79
3.334	Screen	79
3.335	Scroll	79
3.336	Semaphore.....	79
3.337	Session.....	79
3.338	Session Leader	80
3.339	Session Lifetime	80
3.340	Shared Memory Object	80
3.341	Shell.....	80

3.342	Shell, the.....	80
3.343	Shell Script.....	80
3.344	Signal	80
3.345	Signal Stack	81
3.346	Single-Quote	81
3.347	Slash.....	81
3.348	Socket	81
3.349	Socket Address.....	81
3.350	Soft Limit	81
3.351	Source Code	81
3.352	Space Character (<space>).....	82
3.353	Spawn.....	82
3.354	Special Built-In	82
3.355	Special Parameter.....	82
3.356	Spin Lock	82
3.357	Sporadic Server	82
3.358	Standard Error	82
3.359	Standard Input.....	82
3.360	Standard Output	82
3.361	Standard Utilities	83
3.362	Stream.....	83
3.363	STREAM	83
3.364	STREAM End.....	83
3.365	STREAM Head	83
3.366	STREAMS Multiplexor	83
3.367	String	83
3.368	Subshell.....	84
3.369	Successfully Transferred.....	84
3.370	Supplementary Group ID	84
3.371	Suspended Job	84
3.372	Symbolic Link	84
3.373	Synchronized Input and Output	84
3.374	Synchronized I/O Completion.....	84
3.375	Synchronized I/O Data Integrity Completion	85
3.376	Synchronized I/O File Integrity Completion	85
3.377	Synchronized I/O Operation	85
3.378	Synchronous I/O Operation	85
3.379	Synchronously-Generated Signal.....	85
3.380	System	85
3.381	System Crash	86
3.382	System Console	86
3.383	System Databases	86
3.384	System Documentation	86
3.385	System Process	86
3.386	System Reboot	87
3.387	System Trace Event.....	87
3.388	System-Wide	87
3.389	Tab Character (<tab>)	87

3.390	Terminal (or Terminal Device)	87
3.391	Text Column.....	87
3.392	Text File	88
3.393	Thread	88
3.394	Thread ID.....	88
3.395	Thread List	88
3.396	Thread-Safe	88
3.397	Thread-Specific Data Key.....	88
3.398	Tilde	89
3.399	Timeouts	89
3.400	Timer	89
3.401	Timer Overrun.....	89
3.402	Token	89
3.403	Trace Analyzer Process	89
3.404	Trace Controller Process	89
3.405	Trace Event	89
3.406	Trace Event Type.....	89
3.407	Trace Event Type Mapping	90
3.408	Trace Filter.....	90
3.409	Trace Generation Version.....	90
3.410	Trace Log.....	90
3.411	Trace Point.....	90
3.412	Trace Stream.....	90
3.413	Trace Stream Identifier.....	90
3.414	Trace System	90
3.415	Traced Process	90
3.416	Tracing Status of a Trace Stream.....	91
3.417	Typed Memory Name Space	91
3.418	Typed Memory Object	91
3.419	Typed Memory Pool.....	91
3.420	Typed Memory Port	91
3.421	Unbind.....	91
3.422	Unit Data.....	91
3.423	Upshifting.....	91
3.424	User Database	92
3.425	User ID	92
3.426	User Name.....	92
3.427	User Trace Event	92
3.428	Utility.....	92
3.429	Variable	93
3.430	Vertical-Tab Character (<vertical-tab>)	93
3.431	White Space.....	93
3.432	Wide-Character Code (C Language)	93
3.433	Wide-Character Input/Output Functions.....	93
3.434	Wide-Character String	93
3.435	Word	94
3.436	Working Directory (or Current Working Directory)	94
3.437	Worldwide Portability Interface	94

	3.438	Write	94
	3.439	XSI	94
	3.440	XSI-Conformant	94
	3.441	Zombie Process	94
	3.442	± 0	95
Chapter	4	General Concepts	97
	4.1	Concurrent Execution	97
	4.2	Directory Protection	97
	4.3	Extended Security Controls	97
	4.4	File Access Permissions	97
	4.5	File Hierarchy	98
	4.6	Filenames	98
	4.7	File Times Update	98
	4.8	Host and Network Byte Orders	99
	4.9	Measurement of Execution Time	99
	4.10	Memory Synchronization	100
	4.11	Pathname Resolution	100
	4.12	Process ID Reuse	101
	4.13	Scheduling Policy	101
	4.14	Seconds Since the Epoch	102
	4.15	Semaphore	102
	4.16	Thread-Safety	103
	4.17	Tracing	103
	4.18	Treatment of Error Conditions for Mathematical Functions	105
	4.18.1	Domain Error	105
	4.18.2	Pole Error	106
	4.18.3	Range Error	106
	4.18.3.1	Result Overflows	106
	4.18.3.2	Result Underflows	106
	4.19	Treatment of NaN Arguments for the Mathematical Functions	106
	4.20	Utility	107
	4.21	Variable Assignment	107
Chapter	5	File Format Notation	109
Chapter	6	Character Set	113
	6.1	Portable Character Set	113
	6.2	Character Encoding	116
	6.3	C Language Wide-Character Codes	117
	6.4	Character Set Description File	117
	6.4.1	State-Dependent Character Encodings	120
Chapter	7	Locale	121
	7.1	General	121
	7.2	POSIX Locale	122
	7.3	Locale Definition	122
	7.3.1	LC_CTYPE	124

7.3.1.1	LC_CTYPE Category in the POSIX Locale	128
7.3.2	LC_COLLATE	132
7.3.2.1	The collating-element Keyword	133
7.3.2.2	The collating-symbol Keyword	134
7.3.2.3	The order_start Keyword	134
7.3.2.4	Collation Order	135
7.3.2.5	The order_end Keyword	137
7.3.2.6	LC_COLLATE Category in the POSIX Locale	137
7.3.3	LC_MONETARY	140
7.3.3.1	LC_MONETARY Category in the POSIX Locale	142
7.3.4	LC_NUMERIC	143
7.3.4.1	LC_NUMERIC Category in the POSIX Locale	144
7.3.5	LC_TIME	144
7.3.5.1	LC_TIME Locale Definition	144
7.3.5.2	LC_TIME C-Language Access	146
7.3.5.3	LC_TIME Category in the POSIX Locale	148
7.3.6	LC_MESSAGES	150
7.3.6.1	LC_MESSAGES Category in the POSIX Locale	150
7.4	Locale Definition Grammar	151
7.4.1	Locale Lexical Conventions	151
7.4.2	Locale Grammar	152
Chapter 8	Environment Variables	159
8.1	Environment Variable Definition	159
8.2	Internationalization Variables	160
8.3	Other Environment Variables	163
Chapter 9	Regular Expressions	167
9.1	Regular Expression Definitions	167
9.2	Regular Expression General Requirements	168
9.3	Basic Regular Expressions	169
9.3.1	BREs Matching a Single Character or Collating Element	169
9.3.2	BRE Ordinary Characters	169
9.3.3	BRE Special Characters	169
9.3.4	Periods in BREs	170
9.3.5	RE Bracket Expression	170
9.3.6	BREs Matching Multiple Characters	172
9.3.7	BRE Precedence	173
9.3.8	BRE Expression Anchoring	173
9.4	Extended Regular Expressions	173
9.4.1	EREs Matching a Single Character or Collating Element	174
9.4.2	ERE Ordinary Characters	174
9.4.3	ERE Special Characters	174
9.4.4	Periods in EREs	175
9.4.5	ERE Bracket Expression	175
9.4.6	EREs Matching Multiple Characters	175
9.4.7	ERE Alternation	176
9.4.8	ERE Precedence	176

	9.4.9	ERE Expression Anchoring.....	176
	9.5	Regular Expression Grammar.....	177
	9.5.1	BRE/ERE Grammar Lexical Conventions.....	177
	9.5.2	RE and Bracket Expression Grammar.....	178
	9.5.3	ERE Grammar.....	180
Chapter	10	Directory Structure and Devices	183
	10.1	Directory Structure and Files	183
	10.2	Output Devices and Terminal Types.....	183
Chapter	11	General Terminal Interface.....	185
	11.1	Interface Characteristics	185
	11.1.1	Opening a Terminal Device File	185
	11.1.2	Process Groups.....	185
	11.1.3	The Controlling Terminal.....	186
	11.1.4	Terminal Access Control	186
	11.1.5	Input Processing and Reading Data.....	187
	11.1.6	Canonical Mode Input Processing	187
	11.1.7	Non-Canonical Mode Input Processing.....	188
	11.1.8	Writing Data and Output Processing	189
	11.1.9	Special Characters.....	189
	11.1.10	Modem Disconnect.....	190
	11.1.11	Closing a Terminal Device File	190
	11.2	Parameters that Can be Set	191
	11.2.1	The termios Structure	191
	11.2.2	Input Modes.....	191
	11.2.3	Output Modes	192
	11.2.4	Control Modes.....	194
	11.2.5	Local Modes	195
	11.2.6	Special Control Characters	196
Chapter	12	Utility Conventions.....	199
	12.1	Utility Argument Syntax.....	199
	12.2	Utility Syntax Guidelines.....	201
Chapter	13	Headers.....	203
	13.1	Format of Entries.....	203
		< aio.h>	204
		<arpa/inet.h>	206
		<assert.h>	207
		<complex.h>	208
		<cpio.h>.....	211
		<ctype.h>.....	212
		<dirent.h>.....	214
		<dlfcn.h>	216
		<errno.h>.....	217
		<fcntl.h>	221
		<fenv.h>.....	224

<float.h>.....	227
<fmtmsg.h>	231
<fnmatch.h>	233
<ftw.h>.....	234
<glob.h>.....	236
<grp.h>.....	238
<iconv.h>.....	239
<inttypes.h>.....	240
<iso646.h>	242
<langinfo.h>.....	243
<libgen.h>.....	246
<limits.h>	247
<locale.h>	261
<math.h>	263
<monetary.h>.....	270
<mqueue.h>.....	271
<ndbm.h>.....	273
<net/if.h>.....	274
<netdb.h>.....	275
<netinet/in.h>.....	279
<netinet/tcp.h>.....	283
<nl_types.h>.....	284
<poll.h>	285
<pthread.h>	287
<pwd.h>.....	292
<regex.h>.....	293
<sched.h>	295
<search.h>.....	297
<semaphore.h>	299
<setjmp.h>	300
<signal.h>	301
<spawn.h>.....	308
<stdarg.h>	310
<stdbool.h>.....	312
<stddef.h>	313
<stdint.h>	314
<stdio.h>.....	321
<stdlib.h>	325
<string.h>.....	329
<strings.h>.....	331
<stropts.h>	332
<sys/ipc.h>.....	337
<sys/mman.h>.....	339
<sys/msg.h>.....	342
<sys/resource.h>.....	343
<sys/select.h>	345
<sys/sem.h>	347
<sys/shm.h>.....	349

<sys/socket.h>.....	351
<sys/stat.h>	356
<sys/statvfs.h>	360
<sys/time.h>	362
<sys/timeb.h>.....	364
<sys/times.h>	365
<sys/types.h>.....	366
<sys/uio.h>.....	369
<sys/un.h>	370
<sys/utsname.h>.....	371
<sys/wait.h>	372
<syslog.h>	374
<tar.h>.....	376
<termios.h>.....	378
<tgmath.h>.....	384
<time.h>	388
<trace.h>	392
<ucontext.h>.....	396
<ulimit.h>	397
<unistd.h>.....	398
<utime.h>	416
<utmpx.h>.....	417
<wchar.h>	419
<wctype.h>	423
<wordexp.h>	425
Index	427

List of Tables

3-1	Job Control Job ID Formats.....	61
5-1	Escape Sequences and Associated Actions	110
6-1	Portable Character Set.....	113
6-2	Control Character Set.....	118
7-1	Valid Character Class Combinations.....	128
10-1	Control Character Names	184

Introduction

Note: This introduction is not part of IEEE Std 1003.1-2001, Standard for Information Technology — Portable Operating System Interface (POSIX).

This standard has been jointly developed by the IEEE and The Open Group. It is both an IEEE Standard and an Open Group Technical Standard.

The Austin Group

This standard was developed, and is maintained, by a joint working group of members of the IEEE Portable Applications Standards Committee, members of The Open Group, and members of ISO/IEC Joint Technical Committee 1. This joint working group is known as the Austin Group.³ The Austin Group arose out of discussions amongst the parties which started in early 1998, leading to an initial meeting and formation of the group in September 1998. The purpose of the Austin Group has been to revise, combine, and update the following standards: ISO/IEC 9945-1, ISO/IEC 9945-2, IEEE Std 1003.1, IEEE Std 1003.2, and the Base Specifications of The Open Group Single UNIX Specification.

After two initial meetings, an agreement was signed in July 1999 between The Open Group and the Institute of Electrical and Electronics Engineers (IEEE), Inc., to formalize the project with the first draft of the revised specifications being made available at the same time. Under this agreement, The Open Group and IEEE agreed to share joint copyright of the resulting work. The Open Group has provided the chair and secretariat for the Austin Group.

The base document for the revision was The Open Group's Base volumes of its Single UNIX Specification, Version 2. These were selected since they were a superset of the existing POSIX.1 and POSIX.2 specifications and had some organizational aspects that would benefit the audience for the new revision.

The approach to specification development has been one of “write once, adopt everywhere”, with the deliverables being a set of specifications that carry the IEEE POSIX designation and The Open Group's Technical Standard designation, and, if approved, an ISO/IEC designation. This set of specifications forms the core of the Single UNIX Specification, Version 3.

This unique development has combined both the industry-led efforts and the formal standardization activities into a single initiative, and included a wide spectrum of participants. The Austin Group continues as the maintenance body for this document.

Anyone wishing to participate in the Austin Group should contact the chair with their request. There are no fees for participation or membership. You may participate as an observer or as a contributor. You do not have to attend face-to-face meetings to participate; electronic participation is most welcome. For more information on the Austin Group and how to participate, see <http://www.opengroup.org/austin>.

3. The Austin Group is named after the location of the inaugural meeting held at the IBM facility in Austin, Texas in September 1998.

Background

The developers of this standard represent a cross section of hardware manufacturers, vendors of operating systems and other software development tools, software designers, consultants, academics, authors, applications programmers, and others.

Conceptually, this standard describes a set of fundamental services needed for the efficient construction of application programs. Access to these services has been provided by defining an interface, using the C programming language, a command interpreter, and common utility programs that establish standard semantics and syntax. Since this interface enables application writers to write portable applications—it was developed with that goal in mind—it has been designated POSIX,⁴ an acronym for Portable Operating System Interface.

Although originated to refer to the original IEEE Std 1003.1-1988, the name POSIX more correctly refers to a *family* of related standards: IEEE Std 1003.*n* and the parts of ISO/IEC 9945. In earlier editions of the IEEE standard, the term POSIX was used as a synonym for IEEE Std 1003.1-1988. A preferred term, POSIX.1, emerged. This maintained the advantages of readability of the symbol “POSIX” without being ambiguous with the POSIX family of standards.

Audience

The intended audience for this standard is all persons concerned with an industry-wide standard operating system based on the UNIX system. This includes at least four groups of people:

1. Persons buying hardware and software systems
2. Persons managing companies that are deciding on future corporate computing directions
3. Persons implementing operating systems, and especially
4. Persons developing applications where portability is an objective

Purpose

Several principles guided the development of this standard:

- Application-Oriented

The basic goal was to promote portability of application programs across UNIX system environments by developing a clear, consistent, and unambiguous standard for the interface specification of a portable operating system based on the UNIX system documentation. This standard codifies the common, existing definition of the UNIX system.

- Interface, Not Implementation

This standard defines an interface, not an implementation. No distinction is made between library functions and system calls; both are referred to as functions. No details of the implementation of any function are given (although historical practice is sometimes indicated in the RATIONALE section). Symbolic names are given for constants (such as signals and error numbers) rather than numbers.

4. The name POSIX was suggested by Richard Stallman. It is expected to be pronounced *pahz-icks*, as in *positive*, not *poh-six*, or other variations. The pronunciation has been published in an attempt to promulgate a standardized way of referring to a standard operating system interface.

- Source, Not Object, Portability

This standard has been written so that a program written and translated for execution on one conforming implementation may also be translated for execution on another conforming implementation. This standard does not guarantee that executable (object or binary) code will execute under a different conforming implementation than that for which it was translated, even if the underlying hardware is identical.

- The C Language

The system interfaces and header definitions are written in terms of the standard C language as specified in the ISO C standard.

- No Superuser, No System Administration

There was no intention to specify all aspects of an operating system. System administration facilities and functions are excluded from this standard, and functions usable only by the superuser have not been included. Still, an implementation of the standard interface may also implement features not in this standard. This standard is also not concerned with hardware constraints or system maintenance.

- Minimal Interface, Minimally Defined

In keeping with the historical design principles of the UNIX system, the mandatory core facilities of this standard have been kept as minimal as possible. Additional capabilities have been added as optional extensions.

- Broadly Implementable

The developers of this standard endeavored to make all specified functions implementable across a wide range of existing and potential systems, including:

1. All of the current major systems that are ultimately derived from the original UNIX system code (Version 7 or later)
2. Compatible systems that are not derived from the original UNIX system code
3. Emulations hosted on entirely different operating systems
4. Networked systems
5. Distributed systems
6. Systems running on a broad range of hardware

No direct references to this goal appear in this standard, but some results of it are mentioned in the Rationale (Informative) volume.

- Minimal Changes to Historical Implementations

When the original version of IEEE Std 1003.1 was published, there were no known historical implementations that did not have to change. However, there was a broad consensus on a set of functions, types, definitions, and concepts that formed an interface that was common to most historical implementations.

The adoption of the 1988 and 1990 IEEE system interface standards, the 1992 IEEE shell and utilities standard, the various Open Group (formerly X/Open) specifications, and the subsequent revisions and addenda to all of them have consolidated this consensus, and this revision reflects the significantly increased level of consensus arrived at since the original versions. The earlier standards and their modifications specified a number of areas where consensus had not been reached before, and these are now reflected in this revision. The authors of the original versions tried, as much as possible, to follow the principles below

when creating new specifications:

1. By standardizing an interface like one in an historical implementation; for example, directories
2. By specifying an interface that is readily implementable in terms of, and backwards-compatible with, historical implementations, such as the extended *tar* format defined in the *pax* utility
3. By specifying an interface that, when added to an historical implementation, will not conflict with it; for example, the *sigaction()* function

This revision tries to minimize the number of changes required to implementations which conform to the earlier versions of the approved standards to bring them into conformance with the current standard. Specifically, the scope of this work excluded doing any “new” work, but rather collecting into a single document what had been spread across a number of documents, and presenting it in what had been proven in practice to be a more effective way. Some changes to prior conforming implementations were unavoidable, primarily as a consequence of resolving conflicts found in prior revisions, or which became apparent when bringing the various pieces together.

However, since it references the 1999 version of the ISO C standard, and no longer supports “Common Usage C”, there are a number of unavoidable changes. Applications portability is similarly affected.

This standard is specifically not a codification of a particular vendor’s product.

It should be noted that implementations will have different kinds of extensions. Some will reflect “historical usage” and will be preserved for execution of pre-existing applications. These functions should be considered “obsolescent” and the standard functions used for new applications. Some extensions will represent functions beyond the scope of this standard. These need to be used with careful management to be able to adapt to future extensions of this standard and/or port to implementations that provide these services in a different manner.

- Minimal Changes to Existing Application Code

A goal of this standard was to minimize additional work for the developers of applications. However, because every known historical implementation will have to change at least slightly to conform, some applications will have to change.

This Standard

This standard defines the Portable Operating System Interface (POSIX) requirements and consists of the following volumes:

- Base Definitions (this volume)
- Shell and Utilities
- System Interfaces
- Rationale (Informative)

This Volume

The Base Definitions volume provides common definitions for this standard, therefore readers should be familiar with it before using the other volumes.

This volume is structured as follows:

- Chapter 1 is an introduction.
- Chapter 2 defines the conformance requirements.
- Chapter 3 defines general terms used.
- Chapter 4 describes general concepts used.
- Chapter 5 describes the notation used to specify file input and output formats in this volume and the Shell and Utilities volume.
- Chapter 6 describes the portable character set and the process of character set definition.
- Chapter 7 describes the syntax for defining internationalization locales as well as the POSIX locale provided on all systems.
- Chapter 8 describes the use of environment variables for internationalization and other purposes.
- Chapter 9 describes the syntax of pattern matching using regular expressions employed by many utilities and matched by the *regcomp()* and *regex()* functions.
- Chapter 10 describes files and devices found on all systems.
- Chapter 11 describes the asynchronous terminal interface for many of the functions in the System Interfaces volume and the *stty* utility in the Shell and Utilities volume.
- Chapter 12 describes the policies for command line argument construction and parsing.
- Chapter 13 defines the contents of headers which declare constants, macros, and data structures that are needed by programs using the services provided by the System Interfaces volume.

Comprehensive references are available in the index.

Typographical Conventions

The following typographical conventions are used throughout this standard. In the text, this standard is referred to as IEEE Std 1003.1-2001, which is technically identical to The Open Group Base Specifications, Issue 6.

The typographical conventions listed here are for ease of reading only. Editorial inconsistencies in the use of typography are unintentional and have no normative meaning in this standard.

Reference	Example	Notes
C-Language Data Structure	aio	
C-Language Data Structure Member	<i>aio_lio_opcode</i>	
C-Language Data Type	long	
C-Language External Variable	<i>errno</i>	
C-Language Function	<i>system()</i>	

Reference	Example	Notes
C-Language Function Argument	<i>arg1</i>	
C-Language Function Family	<i>exec</i>	
C-Language Header	<sys/stat.h>	
C-Language Keyword	return	
C-Language Macro with Argument	<i>assert()</i>	
C-Language Macro with No Argument	INET_ADDRSTRLEN	
C-Language Preprocessing Directive	#define	
Commands within a Utility	a, c	
Conversion Specification, Specifier/Modifier Character	%A, g, E	1
Environment Variable	PATH	
Error Number	[EINTR]	
Example Output	Hello, World	
Filename	/tmp	
Literal Character	'c', '\r', '\'	2
Literal String	"abcde"	2
Optional Items in Utility Syntax	[]	
Parameter	<directory pathname>	
Special Character	<newline>	3
Symbolic Constant	_POSIX_VDISABLE	
Symbolic Limit, Configuration Value	{LINE_MAX}	4
Syntax	#include <sys/stat.h>	
User Input and Example Code	echo Hello, World	5
Utility Name	<i>awk</i>	
Utility Operand	<i>file_name</i>	
Utility Option	-c	
Utility Option with Option-Argument	-w width	

Notes:

1. Conversion specifications, specifier characters, and modifier characters are used primarily in date-related functions and utilities and the *fprintf* and *fscanf* formatting functions.
2. Unless otherwise noted, the quotes shall not be used as input or output. When used in a list item, the quotes are omitted. For literal characters, '**' (or any of the other sequences such as '*''*') is the same as the C constant '**' (or '*\\''*').
3. The style selected for some of the special characters, such as <newline>, matches the form of the input given to the *localedef* utility. Generally, the characters selected for this special treatment are those that are not visually distinct, such as the control characters <tab> or <newline>.
4. Names surrounded by braces represent symbolic limits or configuration values which may be declared in appropriate headers by means of the C **#define** construct.
5. Brackets shown in this font, "[]", are part of the syntax and do *not* indicate optional items. In syntax the '| ' symbol is used to separate alternatives, and ellipses ("...") are used to show that additional arguments are optional.

Shading is used to identify extensions and options; see Section 1.5.1 (on page 6).

Footnotes and notes within the body of the normative text are for information only (informative).

Informative sections (such as Rationale, Change History, Application Usage, and so on) are denoted by continuous shading bars in the margins.

Ranges of values are indicated with parentheses or brackets as follows:

- (a,b) means the range of all values from a to b , including neither a nor b .
- $[a,b]$ means the range of all values from a to b , including a and b .
- $[a,b)$ means the range of all values from a to b , including a , but not b .
- $(a,b]$ means the range of all values from a to b , including b , but not a .

Note: A symbolic limit beginning with POSIX is treated differently, depending on context. In a C-language header, the symbol `POSIXstring` (where *string* may contain underscores) is represented by the C identifier `_POSIXstring`, with a leading underscore required to prevent ISO C standard name space pollution. However, in other contexts, such as languages other than C, the leading underscore is not used because this requirement does not exist.

Participants

The Austin Group

This standard was prepared by the Austin Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society, The Open Group, and ISO/SC22 WG15. At the time of approval, the membership of the Austin Group was as follows.

Andrew Josey, Chair

Donald W. Cragun, Organizational Representative, IEEE PASC

Nicholas Stoughton, Organizational Representative, ISO/SC22 WG15

Mark Brown, Organizational Representative, The Open Group

Cathy Hughes, Technical Editor

Austin Group Technical Reviewers

Peter Anvin

Bouazza Bachar

Theodore P. Baker

Walter Briscoe

Mark Brown

Dave Butenhof

Geoff Clare

Donald W. Cragun

Lee Damico

Ulrich Drepper

Paul Eggert

Joanna Farley

Clive D.W. Feather

Andrew Gollan

Michael Gonzalez

Joseph M. Gwinn

Jon Hitchcock

Yvette Ho Sang

Cathy Hughes

Lowell G. Johnson

Andrew Josey

Michael Kavanaugh

David Korn

Marc Aurele La France

Jim Meyering

Gary Miller

Finnbarr P. Murphy

Joseph S. Myers

Sandra O'Donnell

Frank Prindle

Curtis Royster Jr.

Glen Seeds

Keld Jorn Simonsen

Raja Srinivasan

Nicholas Stoughton

Donn S. Terry

Fred Tydeman

Peter Van Der Veen

James Youngman

Jim Zepeda

Jason Zions

Austin Group Working Group Members

Harold C. Adams	Michael Gonzalez	Sandra O'Donnell
Peter Anvin	Karen D. Gordon	Frank Prindle
Pierre-Jean Arcos	Joseph M. Gwinn	Francois Riche
Jay Ashford	Steven A. Haaser	John D. Riley
Bouazza Bachar	Charles E. Hammons	Andrew K. Roach
Theodore P. Baker	Chris J. Harding	Helmut Roth
Robert Barned	Barry Hedquist	Jaideep Roy
Joel Berman	Vincent E. Henley	Curtis Royster Jr.
David J. Blackwood	Karl Heubaum	Stephen C. Schwarm
Shirley Bockstahler-Brandt	Jon Hitchcock	Glen Seeds
James Bottomley	Yvette Ho Sang	Richard Seibel
Walter Briscoe	Niklas Holsti	David L. Shroads Jr.
Andries Brouwer	Thomas Hosmer	W. Olin Sibert
Mark Brown	Cathy Hughes	Keld Jorn Simonsen
Eric W. Burger	Jim D. Isaak	Curtis Smith
Alan Burns	Lowell G. Johnson	Raja Srinivasan
Andries Brouwer	Michael B. Jones	Nicholas Stoughton
Dave Butenhof	Andrew Josey	Marc J. Teller
Keith Chow	Michael J. Karels	Donn S. Terry
Geoff Clare	Michael Kavanaugh	Fred Tydeman
Donald W. Cragun	David Korn	Mark-Rene Uchida
Lee Damico	Steven Kramer	Scott A. Valcourt
Juan Antonio De La Puente	Thomas M. Kurihara	Peter Van Der Veen
Ming De Zhou	Marc Aurele La France	Michael W. Vannier
Steven J. Dovich	C. Douglass Locke	Eric Vought
Richard P. Draves	Nick Maclaren	Frederick N. Webb
Ulrich Drepper	Roger J. Martin	Paul A.T. Wolfgang
Paul Eggert	Craig H. Meyer	Garrett Wollman
Philip H. Enslow	Jim Meyering	James Youngman
Joanna Farley	Gary Miller	Oren Yuen
Clive D.W. Feather	Finnbarr P. Murphy	Janusz Zalewski
Pete Forman	Joseph S. Myers	Jim Zepeda
Mark Funkenhauser	John Napier	Jason Zions
Lois Goldthwaite	Peter E. Obermayer	
Andrew Gollan	James T. Oblinger	

The Open Group

When The Open Group approved the Base Specifications, Issue 6 on 12 September 2001, the membership of The Open Group Base Working Group was as follows.

Andrew Josey, Chair

Finnbarr P. Murphy, Vice-Chair

Mark Brown, Austin Group Liaison

Cathy Hughes, Technical Editor

Base Working Group Members

Bouazza Bachar

Mark Brown

Dave Butenhof

Donald W. Cragun

Larry Dwyer

Joanna Farley

Andrew Gollan

Karen D. Gordon

Gary Miller

Finnbarr P. Murphy

Frank Prindle

Andrew K. Roach

Curtis Royster Jr.

Nicholas Stoughton

Kenjiro Tsuji

IEEE

When the IEEE Standards Board approved IEEE Std 1003.1-2001 on 6 December 2001, the membership of the committees was as follows.

Portable Applications Standards Committee (PASC)

Lowell G. Johnson, Chair
Joseph M. Gwinn, Vice-Chair
Jay Ashford, Functional Chair
Andrew Josey, Functional Chair
Curtis Royster Jr., Functional Chair
Nicholas Stoughton, Secretary

Balloting Committee

The following members of the balloting committee voted on IEEE Std 1003.1-2001. Balloters may have voted for approval, disapproval, or abstention.

Harold C. Adams	Steven A. Haaser	Frank Prindle
Pierre-Jean Arcos	Charles E. Hammons	Francois Riche
Jay Ashford	Chris J. Harding	John D. Riley
Theodore P. Baker	Barry Hedquist	Andrew K. Roach
Robert Barned	Vincent E. Henley	Helmut Roth
David J. Blackwood	Karl Heubaum	Jaideep Roy
Shirley Bockstahler-Brandt	Niklas Holsti	Curtis Royster Jr.
James Bottomley	Thomas Hosmer	Stephen C. Schwarm
Mark Brown	Jim D. Isaak	Richard Seibel
Eric W. Burger	Lowell G. Johnson	David L. Shroads Jr.
Alan Burns	Michael B. Jones	W. Olin Sibert
Dave Butenhof	Andrew Josey	Keld Jorn Simonsen
Keith Chow	Michael J. Karels	Nicholas Stoughton
Donald W. Cragun	Steven Kramer	Donn S. Terry
Juan Antonio De La Puente	Thomas M. Kurihara	Mark-Rene Uchida
Ming De Zhou	C. Douglass Locke	Scott A. Valcourt
Steven J. Dovich	Roger J. Martin	Michael W. Vannier
Richard P. Draves	Craig H. Meyer	Frederick N. Webb
Philip H. Enslow	Finnbarr P. Murphy	Paul A.T. Wolfgang
Michael Gonzalez	John Napier	Oren Yuen
Karen D. Gordon	Peter E. Obermayer	Janusz Zalewski
Joseph M. Gwinn	James T. Oblinger	

The following organizational representative voted on this standard:

Andrew Josey, X/Open Company Ltd.

IEEE-SA Standards Board

When the IEEE-SA Standards Board approved IEEE Std 1003.1-2001 on 6 December 2001, it had the following membership:

Donald N. Heirman, Chair

James T. Carlo, Vice-Chair

Judith Gorman, Secretary

Satish K. Aggarwal

Mark D. Bowman

Gary R. Engmann

Harold E. Epstein

H. Landis Floyd

Jay Forster*

Howard M. Frazier

Ruben D. Garzon

James H. Gurney

Richard J. Holleman

Lowell G. Johnson

Robert J. Kennelly

Joseph L. Koepfinger*

Peter H. Lips

L. Bruce McClung

Daleep C. Mohla

James W. Moore

Robert F. Munzner

Ronald C. Petersen

Gerald H. Peterson

John B. Posey

Gary S. Robinson

Akio Tojo

Donald W. Zipse

Also included are the following non-voting IEEE-SA Standards Board liaisons:

Alan Cookson, NIST Representative

Donald R. Volzka, TAB Representative

Yvette Ho Sang, **Don Messina**, **Savoula Amanatidis**, IEEE Project Editors

* Member Emeritus

Trademarks

The following information is given for the convenience of users of this standard and does not constitute endorsement of these products by The Open Group or the IEEE. There may be other products mentioned in the text that might be covered by trademark protection and readers are advised to verify them independently.

1003.1[™] is a trademark of the Institute of Electrical and Electronic Engineers, Inc.

AIX[®] is a registered trademark of IBM Corporation.

AT&T[®] is a registered trademark of AT&T in the U.S.A. and other countries.

BSD[™] is a trademark of the University of California, Berkeley, U.S.A.

Hewlett-Packard[®], HP[®], and HP-UX[®] are registered trademarks of Hewlett-Packard Company.

IBM[®] is a registered trademark of International Business Machines Corporation.

Motif[®], OSF/1[®], UNIX[®], and the “X Device” are registered trademarks and IT DialTone[™] and

The Open Group[™] are trademarks of The Open Group in the U.S. and other countries.

POSIX[™] is a trademark of the Institute of Electrical and Electronic Engineers, Inc.

Sun[®] and Sun Microsystems[®] are registered trademarks of Sun Microsystems, Inc.

/usr/group[®] is a registered trademark of UniForum, the International Network of UNIX System Users.

Acknowledgements

The contributions of the following organizations to the development of IEEE Std 1003.1-2001 are gratefully acknowledged:

- AT&T for permission to reproduce portions of its copyrighted System V Interface Definition (SVID) and material from the UNIX System V Release 2.0 documentation.
- The SC22 WG14 Committees.

This standard was prepared by the Austin Group, a joint working group of the IEEE, The Open Group, and ISO SC22 WG15.

Referenced Documents

Normative References

Normative references for this standard are defined in Section 1.3 (on page 4).

Informative References

The following documents are referenced in this standard:

1984 /usr/group Standard

/usr/group Standards Committee, Santa Clara, CA, UniForum 1984.

Almasi and Gottlieb

George S. Almasi and Allan Gottlieb, *Highly Parallel Computing*, The Benjamin/Cummings Publishing Company, Inc., 1989, ISBN: 0-8053-0177-1.

ANSI C

American National Standard for Information Systems: Standard X3.159-1989, Programming Language C.

ANSI X3.226-1994

American National Standard for Information Systems: Standard X3.226-1994, Programming Language Common LISP.

Brawer

Steven Brawer, *Introduction to Parallel Programming*, Academic Press, 1989, ISBN: 0-12-128470-0.

DeRemer and Pennello Article

DeRemer, Frank and Pennello, Thomas J., *Efficient Computation of LALR(1) Look-Ahead Sets*, SigPlan Notices, Volume 15, No. 8, August 1979.

Draft ANSI X3J11.1

IEEE Floating Point draft report of ANSI X3J11.1 (NCEG).

FIPS 151-1

Federal Information Procurement Standard (FIPS) 151-1. Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language].

FIPS 151-2

Federal Information Procurement Standards (FIPS) 151-2, Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language].

HP-UX Manual

Hewlett-Packard HP-UX Release 9.0 Reference Manual, Third Edition, August 1992.

IEC 60559: 1989

IEC 60559: 1989, Binary Floating-Point Arithmetic for Microprocessor Systems (previously designated IEC 559: 1989).

IEEE Std 754-1985

IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic.

IEEE Std 854-1987

IEEE Std 854-1987, IEEE Standard for Radix-Independent Floating-Point Arithmetic.

- IEEE Std 1003.9-1992
IEEE Std 1003.9-1992, IEEE Standard for Information Technology — POSIX FORTRAN 77 Language Interfaces — Part 1: Binding for System Application Program Interface API.
- IETF RFC 791
Internet Protocol, Version 4 (IPv4), September 1981.
- IETF RFC 819
The Domain Naming Convention for Internet User Applications, Z. Su, J. Postel, August 1982.
- IETF RFC 822
Standard for the Format of ARPA Internet Text Messages, D.H. Crocker, August 1982.
- IETF RFC 919
Broadcasting Internet Datagrams, J. Mogul, October 1984.
- IETF RFC 920
Domain Requirements, J. Postel, J. Reynolds, October 1984.
- IETF RFC 921
Domain Name System Implementation Schedule, J. Postel, October 1984.
- IETF RFC 922
Broadcasting Internet Datagrams in the Presence of Subnets, J. Mogul, October 1984.
- IETF RFC 1034
Domain Names — Concepts and Facilities, P. Mockapetris, November 1987.
- IETF RFC 1035
Domain Names — Implementation and Specification, P. Mockapetris, November 1987.
- IETF RFC 1123
Requirements for Internet Hosts — Application and Support, R. Braden, October 1989.
- IETF RFC 1886
DNS Extensions to Support Internet Protocol, Version 6 (IPv6), C. Huitema, S. Thomson, December 1995.
- IETF RFC 2045
Multipurpose Internet Mail Extensions (MIME), Part 1: Format of Internet Message Bodies, N. Freed, N. Borenstein, November 1996.
- IETF RFC 2373
Internet Protocol, Version 6 (IPv6) Addressing Architecture, S. Deering, R. Hinden, July 1998.
- IETF RFC 2460
Internet Protocol, Version 6 (IPv6), S. Deering, R. Hinden, December 1998.
- Internationalisation Guide
Guide, July 1993, Internationalisation Guide, Version 2 (ISBN: 1-859120-02-4, G304), published by The Open Group.
- ISO C (1990)
ISO/IEC 9899:1990, Programming Languages — C, including Amendment 1:1995 (E), C Integrity (Multibyte Support Extensions (MSE) for ISO C).
- ISO 2375:1985
ISO 2375:1985, Data Processing — Procedure for Registration of Escape Sequences.

ISO 8652:1987

ISO 8652:1987, Programming Languages — Ada (technically identical to ANSI standard 1815A-1983).

ISO/IEC 1539:1990

ISO/IEC 1539:1990, Information Technology — Programming Languages — Fortran (technically identical to the ANSI X3.9-1978 standard [FORTRAN 77]).

ISO/IEC 4873:1991

ISO/IEC 4873:1991, Information Technology — ISO 8-bit Code for Information Interchange — Structure and Rules for Implementation.

ISO/IEC 6429:1992

ISO/IEC 6429:1992, Information Technology — Control Functions for Coded Character Sets.

ISO/IEC 6937:1994

ISO/IEC 6937:1994, Information Technology — Coded Character Set for Text Communication — Latin Alphabet.

ISO/IEC 8802-3:1996

ISO/IEC 8802-3:1996, Information Technology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks — Specific Requirements — Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

ISO/IEC 8859

ISO/IEC 8859, Information Technology — 8-Bit Single-Byte Coded Graphic Character Sets:

Part 1: Latin Alphabet No. 1

Part 2: Latin Alphabet No. 2

Part 3: Latin Alphabet No. 3

Part 4: Latin Alphabet No. 4

Part 5: Latin/Cyrillic Alphabet

Part 6: Latin/Arabic Alphabet

Part 7: Latin/Greek Alphabet

Part 8: Latin/Hebrew Alphabet

Part 9: Latin Alphabet No. 5

Part 10: Latin Alphabet No. 6

Part 13: Latin Alphabet No. 7

Part 14: Latin Alphabet No. 8

Part 15: Latin Alphabet No. 9

ISO POSIX-1:1996

ISO/IEC 9945-1:1996, Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language] (identical to ANSI/IEEE Std 1003.1-1996). Incorporating ANSI/IEEE Stds 1003.1-1990, 1003.1b-1993, 1003.1c-1995, and 1003.1i-1995.

ISO POSIX-2:1993

ISO/IEC 9945-2:1993, Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities (identical to ANSI/IEEE Std 1003.2-1992, as amended by ANSI/IEEE Std 1003.2a-1992).

Issue 1

X/Open Portability Guide, July 1985 (ISBN: 0-444-87839-4).

Issue 2

X/Open Portability Guide, January 1987:

- Volume 1: XVS Commands and Utilities (ISBN: 0-444-70174-5)
- Volume 2: XVS System Calls and Libraries (ISBN: 0-444-70175-3)

Issue 3

X/Open Specification, 1988, 1989, February 1992:

- Commands and Utilities, Issue 3 (ISBN: 1-872630-36-7, C211); this specification was formerly X/Open Portability Guide, Issue 3, Volume 1, January 1989, XSI Commands and Utilities (ISBN: 0-13-685835-X, XO/XPG/89/002)
- System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this specification was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003)
- Curses Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapters 9 to 14 inclusive; this specification was formerly X/Open Portability Guide, Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)
- Headers Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapter 19, Cpio and Tar Headers; this specification was formerly X/Open Portability Guide Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)

Issue 4

CAE Specification, July 1992, published by The Open Group:

- System Interface Definitions (XBD), Issue 4 (ISBN: 1-872630-46-4, C204)
- Commands and Utilities (XCU), Issue 4 (ISBN: 1-872630-48-0, C203)
- System Interfaces and Headers (XSH), Issue 4 (ISBN: 1-872630-47-2, C202)

Issue 4, Version 2

CAE Specification, August 1994, published by The Open Group:

- System Interface Definitions (XBD), Issue 4, Version 2 (ISBN: 1-85912-036-9, C434)
- Commands and Utilities (XCU), Issue 4, Version 2 (ISBN: 1-85912-034-2, C436)
- System Interfaces and Headers (XSH), Issue 4, Version 2 (ISBN: 1-85912-037-7, C435)

Issue 5

Technical Standard, February 1997, published by The Open Group:

- System Interface Definitions (XBD), Issue 5 (ISBN: 1-85912-186-1, C605)
- Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)
- System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)

Knuth Article

Knuth, Donald E., *On the Translation of Languages from Left to Right*, Information and Control, Volume 8, No. 6, October 1965.

KornShell

Bolsky, Morris I. and Korn, David G., *The New KornShell Command and Programming Language*, March 1995, Prentice Hall.

MSE Working Draft

Working draft of ISO/IEC 9899:1990/Add3:Draft, Addendum 3 — Multibyte Support Extensions (MSE) as documented in the ISO Working Paper SC22/WG14/N205 dated 31 March 1992.

POSIX.0:1995

IEEE Std 1003.0-1995, IEEE Guide to the POSIX Open System Environment (OSE) (identical to ISO/IEC TR 14252).

POSIX.1:1988

IEEE Std 1003.1-1988, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].

POSIX.1:1990

IEEE Std 1003.1-1990, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].

POSIX.1a

P1003.1a, Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — (C Language) Amendment

POSIX.1d:1999

IEEE Std 1003.1d-1999, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 4: Additional Realtime Extensions [C Language].

POSIX.1g:2000

IEEE Std 1003.1g-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 6: Protocol-Independent Interfaces (PII).

POSIX.1j:2000

IEEE Std 1003.1j-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 5: Advanced Realtime Extensions [C Language].

POSIX.1q:2000

IEEE Std 1003.1q-2000, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 7: Tracing [C Language].

POSIX.2b

P1003.2b, Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities — Amendment

POSIX.2d:-1994

IEEE Std 1003.2d:1994, IEEE Standard for Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities — Amendment 1: Batch Environment.

POSIX.13:-1998

IEEE Std 1003.13:1998, IEEE Standard for Information Technology — Standardized Application Environment Profile (AEP) — POSIX Realtime Application Support.

Sarwate Article

Sarwate, Dilip V., *Computation of Cyclic Redundancy Checks via Table Lookup*, Communications of the ACM, Volume 30, No. 8, August 1988.

Sprunt, Sha, and Lehoczky

Sprunt, B., Sha, L., and Lehoczky, J.P., *Aperiodic Task Scheduling for Hard Real-Time Systems*, The Journal of Real-Time Systems, Volume 1, 1989, Pages 27-60.

SVID, Issue 1

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 1; Morristown, NJ, UNIX Press, 1985.

SVID, Issue 2

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 2; Morristown, NJ, UNIX Press, 1986.

SVID, Issue 3

American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue 3; Morristown, NJ, UNIX Press, 1989.

The AWK Programming Language

Aho, Alfred V., Kernighan, Brian W., and Weinberger, Peter J., *The AWK Programming Language*, Reading, MA, Addison-Wesley 1988.

UNIX Programmer's Manual

American Telephone and Telegraph Company, *UNIX Time-Sharing System: UNIX Programmer's Manual*, 7th Edition, Murray Hill, NJ, Bell Telephone Laboratories, January 1979.

XNS, Issue 4

CAE Specification, August 1994, Networking Services, Issue 4 (ISBN: 1-85912-049-0, C438), published by The Open Group.

XNS, Issue 5

CAE Specification, February 1997, Networking Services, Issue 5 (ISBN: 1-85912-165-9, C523), published by The Open Group.

XNS, Issue 5.2

Technical Standard, January 2000, Networking Services (XNS), Issue 5.2 (ISBN: 1-85912-241-8, C808), published by The Open Group.

X/Open Curses, Issue 4, Version 2

CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3, C610), published by The Open Group.

Yacc

Yacc: Yet Another Compiler Compiler, Stephen C. Johnson, 1978.

Source Documents

Parts of the following documents were used to create the base documents for this standard:

AIX 3.2 Manual

AIX Version 3.2 For RISC System/6000, Technical Reference: Base Operating System and Extensions, 1990, 1992 (Part No. SC23-2382-00).

OSF/1

OSF/1 Programmer's Reference, Release 1.2 (ISBN: 0-13-020579-6).

OSF AES

Application Environment Specification (AES) Operating System Programming Interfaces
Volume, Revision A (ISBN: 0-13-043522-8).

System V Release 2.0

- UNIX System V Release 2.0 Programmer's Reference Manual (April 1984 - Issue 2).
- UNIX System V Release 2.0 Programming Guide (April 1984 - Issue 2).

System V Release 4.2

Operating System API Reference, UNIX SVR4.2 (1992) (ISBN: 0-13-017658-3).

Introduction

1.1 Scope

IEEE Std 1003.1-2001 defines a standard operating system interface and environment, including a command interpreter (or “shell”), and common utility programs to support applications portability at the source code level. It is intended to be used by both applications developers and system implementors.

IEEE Std 1003.1-2001 comprises four major components (each in an associated volume):

1. General terms, concepts, and interfaces common to all volumes of IEEE Std 1003.1-2001, including utility conventions and C-language header definitions, are included in the Base Definitions volume of IEEE Std 1003.1-2001.
2. Definitions for system service functions and subroutines, language-specific system services for the C programming language, function issues, including portability, error handling, and error recovery, are included in the System Interfaces volume of IEEE Std 1003.1-2001.
3. Definitions for a standard source code-level interface to command interpretation services (a “shell”) and common utility programs for application programs are included in the Shell and Utilities volume of IEEE Std 1003.1-2001.
4. Extended rationale that did not fit well into the rest of the document structure, containing historical information concerning the contents of IEEE Std 1003.1-2001 and why features were included or discarded by the standard developers, is included in the Rationale (Informative) volume of IEEE Std 1003.1-2001.

The following areas are outside of the scope of IEEE Std 1003.1-2001:

- Graphics interfaces
- Database management system interfaces
- Record I/O considerations
- Object or binary code portability
- System configuration and resource availability

IEEE Std 1003.1-2001 describes the external characteristics and facilities that are of importance to applications developers, rather than the internal construction techniques employed to achieve these capabilities. Special emphasis is placed on those functions and facilities that are needed in a wide variety of commercial applications.

The facilities provided in IEEE Std 1003.1-2001 are drawn from the following base documents:

- IEEE Std 1003.1-1996 (POSIX-1) (incorporating IEEE Std 1003.1-1990, 1003.1b-1993, 1003.1c-1995, and 1003.1i-1995)
- The following amendments to the POSIX.1-1990 standard:
 - IEEE P1003.1a draft standard (Additional System Services)
 - IEEE Std 1003.1d-1999 (Additional Realtime Extensions)

- IEEE Std 1003.1g-2000 (Protocol-Independent Interfaces (PII))
- IEEE Std 1003.1j-2000 (Advanced Realtime Extensions)
- IEEE Std 1003.1q-2000 (Tracing)
- IEEE Std 1003.2-1992 (POSIX-2) (includes IEEE Std 1003.2a-1992)
- The following amendments to the ISO POSIX-2: 1993 standard:
 - IEEE P1003.2b draft standard (Additional Utilities)
 - IEEE Std 1003.2d-1994 (Batch Environment)
- Open Group Technical Standard, February 1997, System Interface Definitions, Issue 5 (XBD5) (ISBN: 1-85912-186-1, C605)
- Open Group Technical Standard, February 1997, Commands and Utilities, Issue 5 (XCU5) (ISBN: 1-85912-191-8, C604)
- Open Group Technical Standard, February 1997, System Interfaces and Headers, Issue 5 (XSH5) (in 2 Volumes) (ISBN: 1-85912-181-0, C606)
- Note:** XBD5, XCU5, and XSH5 are collectively referred to as the *Base Specifications*.
- Open Group Technical Standard, January 2000, Networking Services, Issue 5.2 (XNS5.2) (ISBN: 1-85912-241-8, C808)
- ISO/IEC 9899: 1999, Programming Languages — C.

IEEE Std 1003.1-2001 uses the Base Specifications as its organizational basis and adds the following additional functionality to them, drawn from the base documents above:

- Normative text from the ISO POSIX-1: 1996 standard and the ISO POSIX-2: 1993 standard not included in the Base Specifications
- The amendments to the POSIX.1-1990 standard and the ISO POSIX-2: 1993 standard listed above, except for parts of IEEE Std 1003.1g-2000
- Portability Considerations
- Additional rationale and notes

The following features, marked legacy or obsolescent in the base documents, are not carried forward into IEEE Std 1003.1-2001. Other features from the base documents marked legacy or obsolescent are carried forward unless otherwise noted.

From XSH5, the following legacy interfaces, headers, and external variables are not carried forward:

advance(), *brk()*, *chroot()*, *compile()*, *cuserid()*, *gamma()*, *getdtablesize()*, *getpagesize()*, *getpass()*, *getw()*, *putw()*, *re_comp()*, *re_exec()*, *regcmp()*, *sbrk()*, *sigstack()*, *step()*, *wait3()*, **<re_comp.h>**, **<regexp.h>**, **<varargs.h>**, *loc1*, *__loc1*, *loc2*, *locs*

From XCU5, the following legacy utilities are not carried forward:

calendar, *cancel*, *cc*, *col*, *cpio*, *cu*, *dircmp*, *dis*, *egrep*, *fgrep*, *line*, *lint*, *lpstat*, *mail*, *pack*, *pcat*, *pg*, *spell*, *sum*, *tar*, *unpack*, *uulog*, *uuname*, *uupick*, *uuto*

In addition, legacy features within non-legacy reference pages (for example, headers) are not carried forward.

From the ISO POSIX-1: 1996 standard, the following obsolescent features are not carried forward:

- 78 Page 112, *CLK_TCK*
 79 Page 197 *tcgetattr()* rate returned option
- 80 From the ISO POSIX-2: 1993 standard, obsolescent features within the following pages are not
 81 carried forward:
- 82 Page 75, zero-length prefix within *PATH*
 83 Page 156, 159 *set*
 84 Page 178, *awk*, use of no argument and no parentheses with length
 85 Page 259, *ed*
 86 Page 272, *env*
 87 Page 282, *find* **perm**[-]onum
 88 Page 295-296, *egrep*
 89 Page 299-300, *head*
 90 Page 305-306, *join*
 91 Page 309-310, *kill*
 92 Page 431-433, 435-436, *sort*
 93 Page 444-445, *tail*
 94 Page 453, 455-456, *touch*
 95 Page 464-465, *tty*
 96 Page 472, *uniq*
 97 Page 515-516, *ex*
 98 Page 542-543, *expand*
 99 Page 563-565, *more*
 100 Page 574-576, *newgrp*
 101 Page 578, *nice*
 102 Page 594-596, *renice*
 103 Page 597-598, *split*
 104 Page 600-601, *strings*
 105 Page 624-625, *vi*
 106 Page 693, *lex*
- 107 The *c89* utility (which specified a compiler for the C Language specified by the
 108 ISO/IEC 9899: 1990 standard) has been replaced by a *c99* utility (which specifies a compiler for
 109 the C Language specified by the ISO/IEC 9899: 1999 standard).
- 110 From XSH5, text marked OH (Optional Header) has been reviewed on a case-by-case basis and
 111 removed where appropriate. The XCU5 text marked OF (Output Format Incompletely Specified)
 112 and UN (Possibly Unsupportable Feature) has been reviewed on a case-by-case basis and
 113 removed where appropriate.
- 114 For the networking interfaces, the base document is the XNS, Issue 5.2 specification. The
 115 following parts of the XNS, Issue 5.2 specification are out of scope and not included in
 116 IEEE Std 1003.1-2001:
- 117 • Part 3 (XTI)
 - 118 • Part 4 (Appendixes)
- 119 Since there is much duplication between the XNS, Issue 5.2 specification and
 120 IEEE Std 1003.1g-2000, material only from the following sections of IEEE Std 1003.1g-2000 has
 121 been included:
- 122 • General terms related to sockets (Section 2.2.2)
 - 123 • Socket concepts (Sections 5.1 through 5.3, inclusive)

- The *pselect()* function (Sections 6.2.2.1 and 6.2.3)
- The *socketmark()* function (Section 5.4.13)
- The `<sys/select.h>` header (Section 6.2)

Emphasis is placed on standardizing existing practice for existing users, with changes and additions limited to correcting deficiencies in the following areas:

- Issues raised by IEEE or ISO/IEC Interpretations against IEEE Std 1003.1 and IEEE Std 1003.2
- Issues raised in corrigenda for the Base Specifications and working group resolutions from The Open Group
- Corrigenda and resolutions passed by The Open Group for the XNS, Issue 5.2 specification
- Changes to make the text self-consistent with the additional material merged
- A reorganization of the options in order to facilitate profiling, both for smaller profiles such as IEEE Std 1003.13, and larger profiles such as the Single UNIX Specification
- Alignment with the ISO/IEC 9899:1999 standard

1.2 Conformance

Conformance requirements for IEEE Std 1003.1-2001 are defined in Chapter 2 (on page 15).

1.3 Normative References

The following standards contain provisions which, through references in IEEE Std 1003.1-2001, constitute provisions of IEEE Std 1003.1-2001. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on IEEE Std 1003.1-2001 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.

ANS X3.9-1978

(Reaffirmed 1989) American National Standard for Information Systems: Standard X3.9-1978, Programming Language FORTRAN.¹

ISO/IEC 646:1991

ISO/IEC 646:1991, Information Processing — ISO 7-Bit Coded Character Set for Information Interchange.²

ISO 4217:1995

ISO 4217:1995, Codes for the Representation of Currencies and Funds.

ISO 8601:2000

ISO 8601:2000, Data Elements and Interchange Formats — Information Interchange —

- 161 Representation of Dates and Times.
- 162 ISO C (1999)
- 163 ISO/IEC 9899: 1999, Programming Languages — C, including Technical Corrigendum No. 1.
- 164 ISO/IEC 10646-1:2000
- 165 ISO/IEC 10646-1:2000, Information Technology — Universal Multiple-Octet Coded
- 166 Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.

167 1.4 Terminology

168 For the purposes of IEEE Std 1003.1-2001, the following terminology definitions apply:

169 **can**

170 Describes a permissible optional feature or behavior available to the user or application. The

171 feature or behavior is mandatory for an implementation that conforms to

172 IEEE Std 1003.1-2001. An application can rely on the existence of the feature or behavior.

173 **implementation-defined**

174 Describes a value or behavior that is not defined by IEEE Std 1003.1-2001 but is selected by

175 an implementor. The value or behavior may vary among implementations that conform to

176 IEEE Std 1003.1-2001. An application should not rely on the existence of the value or

177 behavior. An application that relies on such a value or behavior cannot be assured to be

178 portable across conforming implementations.

179 The implementor shall document such a value or behavior so that it can be used correctly

180 by an application.

181 **legacy**

182 Describes a feature or behavior that is being retained for compatibility with older

183 applications, but which has limitations which make it inappropriate for developing portable

184 applications. New applications should use alternative means of obtaining equivalent

185 functionality.

186 **may**

187 Describes a feature or behavior that is optional for an implementation that conforms to

188 IEEE Std 1003.1-2001. An application should not rely on the existence of the feature or

189 behavior. An application that relies on such a feature or behavior cannot be assured to be

190 portable across conforming implementations.

191 To avoid ambiguity, the opposite of *may* is expressed as *need not*, instead of *may not*.

192 **shall**

193 For an implementation that conforms to IEEE Std 1003.1-2001, describes a feature or

194 behavior that is mandatory. An application can rely on the existence of the feature or

195 behavior.

196 For an application or user, describes a behavior that is mandatory.

197 **should**

198 For an implementation that conforms to IEEE Std 1003.1-2001, describes a feature or

199 behavior that is recommended but not mandatory. An application should not rely on the

200 existence of the feature or behavior. An application that relies on such a feature or behavior

201 cannot be assured to be portable across conforming implementations.

202 For an application, describes a feature or behavior that is recommended programming

203 practice for optimum portability.

undefined

Describes the nature of a value or behavior not defined by IEEE Std 1003.1-2001 which results from use of an invalid program construct or invalid data input.

The value or behavior may vary among implementations that conform to IEEE Std 1003.1-2001. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

unspecified

Describes the nature of a value or behavior not specified by IEEE Std 1003.1-2001 which results from use of a valid program construct or valid data input.

The value or behavior may vary among implementations that conform to IEEE Std 1003.1-2001. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be assured to be portable across conforming implementations.

1.5 Portability

Some of the utilities in the Shell and Utilities volume of IEEE Std 1003.1-2001 and functions in the System Interfaces volume of IEEE Std 1003.1-2001 describe functionality that might not be fully portable to systems meeting the requirements for POSIX conformance (see the Base Definitions volume of IEEE Std 1003.1-2001, Chapter 2, Conformance).

Where optional, enhanced, or reduced functionality is specified, the text is shaded and a code in the margin identifies the nature of the option, extension, or warning (see Section 1.5.1). For maximum portability, an application should avoid such functionality.

Unless the primary task of a utility is to produce textual material on its standard output, application developers should not rely on the format or content of any such material that may be produced. Where the primary task *is* to provide such material, but the output format is incompletely specified, the description is marked with the OF margin code and shading. Application developers are warned not to expect that the output of such an interface on one system is any guide to its behavior on another system.

1.5.1 Codes

The codes and their meanings are as follows. See also Section 1.5.2 (on page 14).

ADV

The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the ADV margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the ADV margin legend.

AIO

The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the AIO margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the AIO margin legend.