

Draft Standard for Information Technology— Portable Operating System Interface (POSIX[®])

Prepared by the Austin Group
(<http://www.opengroup.org/austin/>)

Copyright © 2001 The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA.

Copyright © 2001 The Open Group
Apex Plaza, Forbury Road, Reading, Berkshire RG11AX, UK.

All rights reserved.

Except as permitted below, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners. This is an unapproved draft, subject to change. Permission is hereby granted for Austin Group participants to reproduce this document for purposes of IEEE, the Open Group, and JTC1 standardization activities. Other entities seeking permission to reproduce this document for standardization purposes or other activities must contact the copyright owners for an appropriate license. Use of information contained within this unapproved draft is at your own risk.

Portions of this document 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.

1 / *Technical Standard*

2 **Shell and Utilities, Issue 6**

3 *The Open Group*
4 *The Institute of Electrical and Electronics Engineers, Inc.*

5



6 © 2001, *The Open Group*

7 © 2001, *The Institute of Electrical and Electronics Engineers, Inc.*

8 All rights reserved.

9 Except as permitted below, no part of this publication may be reproduced, stored in a retrieval
10 system, or transmitted, in any form or by any means, electronic, mechanical, photocopying,
11 recording or otherwise, without the prior permission of the copyright owners.

This is an unapproved draft, subject to change. Permission is hereby granted for Austin Group participants to reproduce IEEE Std 1003.1-200x for purposes of IEEE, The Open Group, and JTC1 standardization activities.

Other entities seeking permission to reproduce IEEE Std 1003.1-200x for standardization purposes or activities must contact the copyright owners for an appropriate license. Use of information contained within this unapproved draft is at your own risk.

Portions of IEEE Std 1003.1-200x 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.

12 Technical Standard

13 Shell and Utilities, Issue 6

14 Document Number:

15 Published in the U.K. by The Open Group, 2001.

16 See <http://www.opengroup.org/austin/bugreport.html> for instructions on commenting on this
17 unapproved draft document.

Contents

19	Chapter 1	Introduction.....	2201	
20	1.1	Scope.....	2201	
21	1.2	Conformance	2201	
22	1.3	Normative References	2201	
23	1.4	Change History	2201	
24	1.5	Terminology	2201	
25	1.6	Definitions.....	2203	
26	1.7	Relationship to Other Documents.....	2203	
27	1.7.1	System Interfaces	2203	
28	1.7.1.1	Process Attributes.....	2203	
29	1.7.1.2	Concurrent Execution of Processes.....	2203	
30	1.7.1.3	File Access Permissions.....	2204	
31	1.7.1.4	File Read, Write, and Creation	2204	
32	1.7.1.5	File Removal	2206	
33	1.7.1.6	File Time Values.....	2206	
34	1.7.1.7	File Contents	2206	
35	1.7.1.8	Pathname Resolution	2207	
36	1.7.1.9	Changing the Current Working Directory.....	2207	
37	1.7.1.10	Establish the Locale	2207	
38	1.7.1.11	Actions Equivalent to Functions.....	2207	
39	1.7.2	Concepts Derived from the ISO C Standard.....	2207	
40	1.7.2.1	Arithmetic Precision and Operations	2207	
41	1.7.2.2	Mathematical Functions	2210	
42	1.8	Portability	2210	
43	1.8.1	Codes.....	2210	
44	1.9	Utility Limits.....	2218	
45	1.10	Grammar Conventions.....	2220	
46	1.11	Utility Description Defaults.....	2221	
47	1.12	Considerations for Utilities in Support of Files of Arbitrary Size	2228	
48	1.13	Built-In Utilities.....	2229	
49	Chapter 2	Shell Command Language	2231	
50	2.1	Shell Introduction	2231	
51	2.2	Quoting.....	2232	
52	2.2.1	Escape Character (Backslash).....	2232	
53	2.2.2	Single-Quotes.....	2232	
54	2.2.3	Double-Quotes	2232	
55	2.3	Token Recognition.....	2233	
56	2.3.1	Alias Substitution	2234	
57	2.4	Reserved Words	2235	
58	2.5	Parameters and Variables.....	2235	
59	2.5.1	Positional Parameters.....	2235	

60	2.5.2	Special Parameters.....	2235
61	2.5.3	Shell Variables.....	2236
62	2.6	Word Expansions.....	2238
63	2.6.1	Tilde Expansion.....	2239
64	2.6.2	Parameter Expansion.....	2239
65	2.6.3	Command Substitution.....	2242
66	2.6.4	Arithmetic Expansion.....	2243
67	2.6.5	Field Splitting.....	2243
68	2.6.6	Pathname Expansion.....	2244
69	2.6.7	Quote Removal.....	2244
70	2.7	Redirection.....	2244
71	2.7.1	Redirecting Input.....	2245
72	2.7.2	Redirecting Output.....	2245
73	2.7.3	Appending Redirected Output.....	2245
74	2.7.4	Here-Document.....	2246
75	2.7.5	Duplicating an Input File Descriptor.....	2246
76	2.7.6	Duplicating an Output File Descriptor.....	2247
77	2.7.7	Open File Descriptors for Reading and Writing.....	2247
78	2.8	Exit Status and Errors.....	2247
79	2.8.1	Consequences of Shell Errors.....	2247
80	2.8.2	Exit Status for Commands.....	2248
81	2.9	Shell Commands.....	2248
82	2.9.1	Simple Commands.....	2248
83	2.9.1.1	Command Search and Execution.....	2249
84	2.9.2	Pipelines.....	2250
85	2.9.3	Lists.....	2251
86	2.9.3.1	Asynchronous Lists.....	2252
87	2.9.3.2	Sequential Lists.....	2252
88	2.9.3.3	AND Lists.....	2252
89	2.9.3.4	OR Lists.....	2253
90	2.9.4	Compound Commands.....	2253
91	2.9.4.1	Grouping Commands.....	2253
92	2.9.4.2	For Loop.....	2253
93	2.9.4.3	Case Conditional Construct.....	2254
94	2.9.4.4	If Conditional Construct.....	2254
95	2.9.4.5	While Loop.....	2255
96	2.9.4.6	Until Loop.....	2255
97	2.9.5	Function Definition Command.....	2256
98	2.10	Shell Grammar.....	2257
99	2.10.1	Shell Grammar Lexical Conventions.....	2257
100	2.10.2	Shell Grammar Rules.....	2257
101	2.11	Signals and Error Handling.....	2262
102	2.12	Shell Execution Environment.....	2263
103	2.13	Pattern Matching Notation.....	2264
104	2.13.1	Patterns Matching a Single Character.....	2264
105	2.13.2	Patterns Matching Multiple Characters.....	2264
106	2.13.3	Patterns Used for Filename Expansion.....	2265
107	2.14	Special Built-In Utilities.....	2266

108		<i>break</i>	2267
109		<i>colon</i>	2269
110		<i>continue</i>	2271
111		<i>dot</i>	2273
112		<i>eval</i>	2275
113		<i>exec</i>	2277
114		<i>exit</i>	2279
115		<i>export</i>	2281
116		<i>readonly</i>	2283
117		<i>return</i>	2285
118		<i>set</i>	2287
119		<i>shift</i>	2293
120		<i>times</i>	2295
121		<i>trap</i>	2297
122		<i>unset</i>	2300
123	Chapter 3	Batch Environment Services	2303
124	3.1	General Concepts.....	2303
125	3.1.1	Batch Client-Server Interaction.....	2303
126	3.1.2	Batch Queues	2303
127	3.1.3	Batch Job Creation	2304
128	3.1.4	Batch Job Tracking	2304
129	3.1.5	Batch Job Routing	2304
130	3.1.6	Batch Job Execution.....	2305
131	3.1.7	Batch Job Exit.....	2305
132	3.1.8	Batch Job Abort	2305
133	3.1.9	Batch Authorization	2305
134	3.1.10	Batch Administration.....	2306
135	3.1.11	Batch Notification.....	2306
136	3.2	Batch Services	2306
137	3.2.1	Batch Job States	2307
138	3.2.2	Deferred Batch Services.....	2308
139	3.2.2.1	Batch Job Execution.....	2308
140	3.2.2.2	Batch Job Routing	2315
141	3.2.2.3	Batch Job Exit.....	2315
142	3.2.2.4	Batch Server Restart	2316
143	3.2.2.5	Batch Job Abort	2316
144	3.2.3	Requested Batch Services.....	2317
145	3.2.3.1	Delete Batch Job Request.....	2317
146	3.2.3.2	Hold Batch Job Request.....	2318
147	3.2.3.3	Batch Job Message Request.....	2318
148	3.2.3.4	Batch Job Status Request	2319
149	3.2.3.5	Locate Batch Job Request	2319
150	3.2.3.6	Modify Batch Job Request.....	2319
151	3.2.3.7	Move Batch Job Request.....	2320
152	3.2.3.8	Queue Batch Job Request	2320
153	3.2.3.9	Batch Queue Status Request.....	2321
154	3.2.3.10	Release Batch Job Request.....	2321

155	3.2.3.11	Rerun Batch Job Request	2322
156	3.2.3.12	Select Batch Jobs Request	2322
157	3.2.3.13	Server Shutdown Request.....	2322
158	3.2.3.14	Server Status Request.....	2323
159	3.2.3.15	Signal Batch Job Request	2323
160	3.2.3.16	Track Batch Job Request	2323
161	3.3	Common Behavior for Batch Environment Utilities	2324
162	3.3.1	Batch Job Identifier	2324
163	3.3.2	Destination	2325
164	3.3.3	Multiple Keyword-Value Pairs	2325
165	Chapter 4	Utilities.....	2327
166		<i>admin</i>	2328
167		<i>alias</i>	2333
168		<i>ar</i>	2336
169		<i>asa</i>	2343
170		<i>at</i>	2346
171		<i>awk</i>	2355
172		<i>basename</i>	2389
173		<i>batch</i>	2392
174		<i>bc</i>	2395
175		<i>bg</i>	2410
176		<i>c99</i>	2413
177		<i>cal</i>	2422
178		<i>cat</i>	2424
179		<i>cd</i>	2428
180		<i>cflow</i>	2432
181		<i>chgrp</i>	2435
182		<i>chmod</i>	2438
183		<i>chown</i>	2444
184		<i>cksum</i>	2448
185		<i>cmp</i>	2453
186		<i>comm</i>	2456
187		<i>command</i>	2459
188		<i>compress</i>	2465
189		<i>cp</i>	2468
190		<i>crontab</i>	2476
191		<i>csplit</i>	2480
192		<i>ctags</i>	2484
193		<i>cut</i>	2489
194		<i>cxref</i>	2493
195		<i>date</i>	2496
196		<i>dd</i>	2503
197		<i>delta</i>	2512
198		<i>df</i>	2516
199		<i>diff</i>	2520
200		<i>dirname</i>	2527
201		<i>du</i>	2530

Contents

202	<i>echo</i>	2534
203	<i>ed</i>	2537
204	<i>env</i>	2553
205	<i>ex</i>	2556
206	<i>expand</i>	2627
207	<i>expr</i>	2630
208	<i>false</i>	2635
209	<i>fc</i>	2637
210	<i>fg</i>	2643
211	<i>file</i>	2645
212	<i>find</i>	2652
213	<i>fold</i>	2660
214	<i>fort77</i>	2663
215	<i>fuser</i>	2669
216	<i>gencat</i>	2672
217	<i>get</i>	2675
218	<i>getconf</i>	2683
219	<i>getopts</i>	2689
220	<i>grep</i>	2694
221	<i>hash</i>	2699
222	<i>head</i>	2702
223	<i>iconv</i>	2705
224	<i>id</i>	2708
225	<i>ipcrm</i>	2712
226	<i>ipcs</i>	2714
227	<i>jobs</i>	2720
228	<i>join</i>	2724
229	<i>kill</i>	2729
230	<i>lex</i>	2734
231	<i>link</i>	2746
232	<i>ln</i>	2748
233	<i>locale</i>	2752
234	<i>localedef</i>	2757
235	<i>logger</i>	2761
236	<i>logname</i>	2763
237	<i>lp</i>	2765
238	<i>ls</i>	2770
239	<i>m4</i>	2778
240	<i>mailx</i>	2785
241	<i>make</i>	2809
242	<i>man</i>	2830
243	<i>mesg</i>	2834
244	<i>mkdir</i>	2837
245	<i>mkfifo</i>	2840
246	<i>more</i>	2842
247	<i>mv</i>	2854
248	<i>newgrp</i>	2859
249	<i>nice</i>	2863

250	<i>nl</i>	2866	
251	<i>nm</i>	2870	
252	<i>nohup</i>	2875	
253	<i>od</i>	2878	
254	<i>paste</i>	2886	
255	<i>patch</i>	2890	
256	<i>pathchk</i>	2896	
257	<i>pax</i>	2900	
258	<i>pr</i>	2935	
259	<i>printf</i>	2940	
260	<i>prs</i>	2945	
261	<i>ps</i>	2950	
262	<i>pwd</i>	2957	
263	<i>qalter</i>	2959	
264	<i>qdel</i>	2968	
265	<i>qhold</i>	2971	
266	<i>qmove</i>	2974	
267	<i>qmsg</i>	2977	
268	<i>qrerun</i>	2980	
269	<i>qrls</i>	2982	
270	<i>qselect</i>	2985	
271	<i>qsig</i>	2994	
272	<i>qstat</i>	2997	
273	<i>qsub</i>	3002	
274	<i>read</i>	3015	
275	<i>renice</i>	3018	
276	<i>rm</i>	3022	
277	<i>rmdel</i>	3027	
278	<i>rmdir</i>	3029	
279	<i>sact</i>	3031	
280	<i>sccs</i>	3034	
281	<i>sed</i>	3039	
282	<i>sh</i>	3048	
283	<i>sleep</i>	3065	
284	<i>sort</i>	3068	
285	<i>split</i>	3074	
286	<i>strings</i>	3077	
287	<i>strip</i>	3080	
288	<i>stty</i>	3082	
289	<i>tabs</i>	3091	
290	<i>tail</i>	3095	
291	<i>talk</i>	3098	
292	<i>tee</i>	3102	
293	<i>test</i>	3105	
294	<i>time</i>	3113	
295	<i>touch</i>	3117	
296	<i>tput</i>	3121	
297	<i>tr</i>	3124	

Contents

298		<i>true</i>	3130
299		<i>tsort</i>	3132
300		<i>tty</i>	3134
301		<i>type</i>	3136
302		<i>ulimit</i>	3138
303		<i>umask</i>	3140
304		<i>unalias</i>	3144
305		<i>uname</i>	3146
306		<i>uncompress</i>	3149
307		<i>unexpand</i>	3152
308		<i>unget</i>	3155
309		<i>uniq</i>	3157
310		<i>unlink</i>	3161
311		<i>uucp</i>	3163
312		<i>uudecode</i>	3167
313		<i>uuencode</i>	3170
314		<i>uustat</i>	3175
315		<i>uux</i>	3178
316		<i>val</i>	3182
317		<i>vi</i>	3185
318		<i>wait</i>	3239
319		<i>wc</i>	3243
320		<i>what</i>	3246
321		<i>who</i>	3248
322		<i>write</i>	3252
323		<i>xargs</i>	3255
324		<i>yacc</i>	3261
325		<i>zcat</i>	3277
326		Index	3279
327	List of Figures		
328	4-1	pax Format Archive Example	2912
329	List of Tables		
330	1-1	Actions when Creating a File that Already Exists	2205
331	1-2	ISO C Standard Operators and Functions	2209
332	1-3	Utility Limit Minimum Values	2218
333	1-4	Symbolic Utility Limits	2219
334	1-5	Regular Built-in Utilities	2229
335	3-1	Batch Utilities	2303
336	3-2	Environment Variable Summary	2307
337	3-3	Next State Table	2309
338	3-4	Results/Output Table	2310
339	3-5	Batch Services Summary	2317
340	4-1	Expressions in Decreasing Precedence in <i>awk</i>	2358

341	4-2	Escape Sequences in <i>awk</i>	2364	
342	4-3	Operators in <i>bc</i>	2400	
343	4-4	Programming Environments: Type Sizes	2417	
344	4-5	Programming Environments: c99 and cc Arguments	2418	
345	4-6	ASCII to EBCDIC Conversion.....	2506	
346	4-7	ASCII to IBM EBCDIC Conversion.....	2507	
347	4-8	File Utility Output Strings	2647	
348	4-9	Table Size Declarations in <i>lex</i>	2737	
349	4-10	Escape Sequences in <i>lex</i>	2739	
350	4-11	ERE Precedence in <i>lex</i>	2740	
351	4-12	Named Characters in <i>od</i>	2881	
352	4-13	ustar Header Block	2917	
353	4-14	ustar <i>mode</i> Field	2918	
354	4-15	Octet-Oriented <i>cpio</i> Archive Entry	2920	
355	4-16	Values for <i>cpio c_mode</i> Field	2921	
356	4-17	Variable Names and Default Headers in <i>ps</i>	2954	
357	4-18	Environment Variable Values (Utilities)	3003	
358	4-19	Control Character Names in <i>stty</i>	3087	
359	4-20	Circumflex Control Characters in <i>stty</i>	3087	
360	4-21	uuencode Base64 Values.....	3171	
361	4-22	Internal Limits in <i>yacc</i>	3273	
362				

364 IEEE Std 1003.1-200x has been jointly developed by the IEEE and The Open Group. It is both an
365 IEEE standard and an Open Group Technical Standard.

366 **Background**

367 The developers of IEEE Std 1003.1-200x represent a cross-section of hardware manufacturers,
368 vendors of operating systems and other software development tools, software designers,
369 consultants, academics, authors, applications programmers, and others.

370 Conceptually, IEEE Std 1003.1-200x describes a set of fundamental services needed for the
371 efficient construction of application programs. Access to these services has been provided by
372 defining an interface, using the C programming language, a command interpreter, and common
373 utility programs that establish standard semantics and syntax. Since this interface enables
374 application writers to write portable applications—it was developed with that goal in mind—it
375 has been designated POSIX,¹ an acronym for Portable Operating System Interface.

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

381 **Audience**

382 The intended audience for IEEE Std 1003.1-200x is all persons concerned with an industry-wide
383 standard operating system based on the UNIX system. This includes at least four groups of
384 people:

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

389 **Purpose**

390 Several principles guided the development of IEEE Std 1003.1-200x:

- 391 • Application-Oriented

392 The basic goal was to promote portability of application programs across UNIX system
393 environments by developing a clear, consistent, and unambiguous standard for the interface
394 specification of a portable operating system based on the UNIX system documentation.
395 IEEE Std 1003.1-200x codifies the common, existing definition of the UNIX system.

396

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

- 400 • Interface, Not Implementation
- 401 IEEE Std 1003.1-200x defines an interface, not an implementation. No distinction is made
- 402 between library functions and system calls; both are referred to as functions. No details of the
- 403 implementation of any function are given (although historical practice is sometimes
- 404 indicated in the RATIONALE section). Symbolic names are given for constants (such as
- 405 signals and error numbers) rather than numbers.
- 406 • Source, Not Object, Portability
- 407 IEEE Std 1003.1-200x has been written so that a program written and translated for execution
- 408 on one conforming implementation may also be translated for execution on another
- 409 conforming implementation. IEEE Std 1003.1-200x does not guarantee that executable (object
- 410 or binary) code will execute under a different conforming implementation than that for
- 411 which it was translated, even if the underlying hardware is identical.
- 412 • The C Language
- 413 The system interfaces and header definitions are written in terms of the standard C language
- 414 as specified in the ISO C standard.
- 415 • No Superuser, No System Administration
- 416 There was no intention to specify all aspects of an operating system. System administration
- 417 facilities and functions are excluded from IEEE Std 1003.1-200x, and functions usable only by
- 418 the superuser have not been included. Still, an implementation of the standard interface may
- 419 also implement features not in IEEE Std 1003.1-200x. IEEE Std 1003.1-200x is also not
- 420 concerned with hardware constraints or system maintenance.
- 421 • Minimal Interface, Minimally Defined
- 422 In keeping with the historical design principles of the UNIX system, the mandatory core
- 423 facilities of IEEE Std 1003.1-200x have been kept as minimal as possible. Additional
- 424 capabilities have been added as optional extensions.
- 425 • Broadly Implementable
- 426 The developers of IEEE Std 1003.1-200x endeavored to make all specified functions
- 427 implementable across a wide range of existing and potential systems, including:
- 428 1. All of the current major systems that are ultimately derived from the original UNIX
- 429 system code (Version 7 or later)
- 430 2. Compatible systems that are not derived from the original UNIX system code
- 431 3. Emulations hosted on entirely different operating systems
- 432 4. Networked systems
- 433 5. Distributed systems
- 434 6. Systems running on a broad range of hardware
- 435 No direct references to this goal appear in IEEE Std 1003.1-200x, but some results of it are
- 436 mentioned in the Rationale (Informative) volume of IEEE Std 1003.1-200x.
- 437 • Minimal Changes to Historical Implementations
- 438 When the original version of IEEE Std 1003.1 was published, there were no known historical
- 439 implementations that did not have to change. However, there was a broad consensus on a set
- 440 of functions, types, definitions, and concepts that formed an interface that was common to
- 441 most historical implementations.

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

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

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

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

467 IEEE Std 1003.1-200x is specifically not a codification of a particular vendor’s product.

468 It should be noted that implementations will have different kinds of extensions. Some will
469 reflect “historical usage” and will be preserved for execution of pre-existing applications.
470 These functions should be considered “obsolescent” and the standard functions used for
471 new applications. Some extensions will represent functions beyond the scope of
472 IEEE Std 1003.1-200x. These need to be used with careful management to be able to adapt to
473 future IEEE Std 1003.1-200x extensions and/or port to implementations that provide these
474 services in a different manner.

475 • Minimal Changes to Existing Application Code

476 A goal of IEEE Std 1003.1-200x was to minimize additional work for the developers of
477 applications. However, because every known historical implementation will have to change
478 at least slightly to conform, some applications will have to change.

479 **IEEE Std 1003.1-200x**

480 IEEE Std 1003.1-200x defines the Portable Operating System Interface (POSIX) requirements and
481 consists of the following volumes:

- 482 • Base Definitions
- 483 • Shell and Utilities (this volume)
- 484 • System Interfaces

- 485 • Rationale (Informative)

486 **This Volume**

487 The Shell and Utilities volume of IEEE Std 1003.1-200x describes the commands and utilities
488 offered to application programs on POSIX-conformant systems. Readers are expected to be
489 familiar with the Base Definitions volume of IEEE Std 1003.1-200x.

490 This volume of IEEE Std 1003.1-200x is structured as follows:

- 491 • Chapter 1 explains the status of this volume of IEEE Std 1003.1-200x and its relationship to
492 other formal standards. It also describes the defaults used by the utility descriptions in
493 Chapter 4.
- 494 • Chapter 2 describes the command language used in POSIX-conformant systems.
- 495 • Chapter 4 consists of reference pages for all utilities available on POSIX-conformant systems.

496 Comprehensive references are available in the index.

497 **Typographical Conventions**

498 The following typographical conventions are used throughout IEEE Std 1003.1-200x.

499 The typographical conventions listed here are for ease of reading only. Editorial inconsistencies
500 in the use of typography are unintentional and have no normative meaning in
501 IEEE Std 1003.1-200x.

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

Reference	Example	Notes
C-Language Data Structure	aiocb	
C-Language Data Structure Member	<i>aio_lio_opcode</i>	
C-Language Data Type	long	
C-Language Function	<i>system()</i>	
C-Language Function Family	<i>exec</i>	
C-Language Function Argument	<i>arg1</i>	
C-Language External Variable	<i>errno</i>	
C-Language Header	<sys/stat.h>	
C-Language Keyword	#define	
C-Language Macro with Argument	<i>assert()</i>	
C-Language Macro with No Argument	INET_ADDRSTRLEN	
Commands within a Utility	a, c	
Conversion Specification, Specifier/Modifier Character	<i>%A, g, E</i>	1
Environment Variable	<i>PATH</i>	
Error Number	[EINTR]	
Example Output	Hello, World	
Filename	/tmp	
Literal Character	<i>'c'</i>	2
Literal String	<i>"abcde"</i>	2
Optional Items in Utility Syntax	[]	
Parameter	<directory pathname>	
Special Character	<newline>	3
Symbolic Limit, Configuration Value	{LINE_MAX}	4

527
528
529
530
531
532
533
534
535

Reference	Example	Notes
Symbolic Constant	<code>_POSIX_VDISABLE</code>	
Syntax	<code>#include <sys/stat.h></code>	
User Input and Example Code	<code>echo Hello, World</code>	5
Utility Name	<code>awk</code>	
Utility Operand	<code>file_name</code>	
Utility Option	<code>-c</code>	
Utility Option with Option-Argument	<code>-w width</code>	

536
537
538
539
540
541
542
543
544
545
546
547
548
549

Notes:

1. Conversion specifications, specifier characters, and modifier characters are used primarily in date-related functions and utilities and the *fprintf* and *scanf* 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.
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.

550
551
552
553
554
555
556
557
558
559

Shading is used to identify extensions and options; see Section 1.8.1 (on page 2210).

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*.

Trademarks

561

562 The following information is given for the convenience of users of IEEE Std 1003.1-200x and
563 does not constitute an endorsement by The Open Group or IEEE of these products. There may be
564 other products mentioned in the text that might be covered by trademark protection and readers
565 are advised to verify them independently.

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

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

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

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

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

571 Motif[®], OSF/1[®], UNIX[®], and the “X Device” are registered trademarks and IT DialTone[™] and
572 The Open Group[™] are trademarks of The Open Group in the U.S. and other countries.

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

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

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

Acknowledgements

577

578 The contributions of the following organizations to the development of IEEE Std 1003.1-200x are
579 gratefully acknowledged:

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

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

585 **Austin Group Common Standards Committee**

586 At the time IEEE Std 1003.1-200x was approved, the membership was as follows:

587 Chair

588 Andrew Josey

589 Organizational Representatives

590 Donald W. Cragun [IEEE PASC]

591 Nicholas Stoughton [ISO/SC22 WG15]

592 Mark Brown [The Open Group]

593 Editor

594 Cathy Hughes

595 Technical Reviewers

596 Peter Anvin

597 Bouazza Bachar

598 Theodore P. Baker

599 Walter Briscoe

600 Dave Butenhof

601 Mark Brown

602 Geoff Clare

603 Donald W. Cragun

604 Lee Damico

605 Ulrich Drepper

606 Paul Eggert

607 Joanna Farley

608 Clive D.W. Feather

609 Andrew Gollan

Michael Gonzalez

Joseph M. Gwinn

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 Simonsen

Raja Srinivasan

Nicholas Stoughton

Donn S. Terry

Fred Tydeman

Peter Van Der Veen

James Youngman

Jim Zepeda

Jason Zions

610 Working Group

611 Harold C. Adams

612 Pierre-Jean Arcos

613 Jay Ashford

614 Peter Anvin

Karen D. Gordon

Joseph M. Gwinn

Steven A. Haaser

Charles E. Hammons

Frank Prindle

Francois Riche

John D. Riley

Andrew K. Roach

Acknowledgements

615	Bouazza Bachar	Chris J. Harding	Helmut Roth
616	Theodore P. Baker	Barry Hedquist	Jaideep Roy
617	Robert Barned	Vincent E. Henley	Curtis Royster Jr.
618	Joel Berman	Karl Heubaum	Stephen C. Schwarm
619	Andries Brouwer	Jon Hitchcock	Richard Seibel
620	David J. Blackwood	Yvette Ho Sang	David L. Shroads Jr.
621	Shirley Bockstahler-Brandt	Niklas Holsti	Curtis Smith
622	James Bottomley	Thomas Hosmer	W. Olin Sibert
623	Walter Briscoe	Cathy Hughes	Glen Seeds
624	Mark Brown	Jim D Isaak	Keld Jorn Simonsen
625	Eric W. Burger	Lowell G. Johnson	Raja Srinivasan
626	Alan Burns	Michael B. Jones	Nicholas Stoughton
627	Dave Butenhof	Andrew Josey	Donn S. Terry
628	Keith Chow	Michael J. Karels	Marc J. Teller
629	Geoff Clare	Michael Kavanaugh	Fred Tydeman
630	Donald W. Cragun	David Korn	Mark-Rene Uchida
631	Lee Damico	Steven Kramer	Scott A. Valcourt
632	Juan Antonio De La Puente	Thomas M. Kurihara	Michael W. Vannier
633	Ming De Zhou	Marc Aurele La France	Frederick N. Webb
634	Steven J. Dovich	C. Douglass Locke	Paul A.T. Wolfgang
635	Richard P. Draves	Roger J Martin	Garrett Wollman
636	Ulrich Drepper	Nick Maclaren	Peter Van Der Veen
637	Paul Eggert	Craig H. Meyer	Eric Vought
638	Philip H. Enslow	Jim Meyering	X/Open Company Ltd.
639	Joanna Farley	Gary Miller	James Youngman
640	Clive D.W. Feather	Finnbarr P. Murphy	Oren Yuen
641	Pete Forman	Joseph S. Myers	Janusz Zalewski
642	Mark Funkenhauser	John Napier	Jim Zepeda
643	Lois Goldthwaite	Peter E. Obermayer	Jason Zions
644	Andrew Gollan	James T. Oblinger	
645	Michael Gonzalez	Sandra O'Donnell	

Referenced Documents

646

647 **Normative References**

648 Normative references for IEEE Std 1003.1-200x are defined in the Base Definitions volume of
649 IEEE Std 1003.1-200x.

650 **Informative References**

651 The following documents are referenced in IEEE Std 1003.1-200x:

652 1984 /usr/group Standard
653 /usr/group Standards Committee, Santa Clara, CA, UniForum 1984.

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

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

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

663 Brawer
664 Steven Brawer, *Introduction to Parallel Programming*, Academic Press, 1989, ISBN:
665 0-12-128470-0.

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

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

671 FIPS 151-1
672 Federal Information Procurement Standard (FIPS) 151-1.

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

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

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

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

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

Referenced Documents

- 685 IEEE Std 1003.9-1992
686 Standard for Information Technology — POSIX FORTRAN 77 Language Interfaces — Part
687 1: Binding for System Application Program Interface API.
- 688 IETF RFC 791
689 Internet Protocol, Version 4 (IPv4), September 1981.
- 690 IETF RFC 819
691 The Domain Naming Convention for Internet User Applications, Z. Su, J. Postel, August
692 1982.
- 693 IETF RFC 822
694 Standard for the Format of ARPA Internet Text Messages, D.H. Crocker, August 1982.
- 695 IETF RFC 919
696 Broadcasting Internet Datagrams, J. Mogul, October 1984.
- 697 IETF RFC 920
698 Domain Requirements, J. Postel, J. Reynolds, October 1984.
- 699 IETF RFC 921
700 Domain Name System Implementation Schedule, J. Postel, October 1984.
- 701 IETF RFC 922
702 Broadcasting Internet Datagrams in the Presence of Subnets, J. Mogul, October 1984.
- 703 IETF RFC 1034
704 Domain Names — Concepts and Facilities, P. Mockapetris, November 1987.
- 705 IETF RFC 1035
706 Domain Names — Implementation and Specification, P. Mockapetris, November 1987.
- 707 IETF RFC 1123
708 Requirements for Internet Hosts — Application and Support, R. Braden, October 1989.
- 709 IETF RFC 1886
710 DNS Extensions to Support Internet Protocol, Version 6 (IPv6), C. Huitema, S. Thomson,
711 December 1995.
- 712 IETF RFC 2045
713 Multipurpose Internet Mail Extensions (MIME), Part 1: Format of Internet Message Bodies,
714 N. Freed, N. Borenstein, November 1996.
- 715 IETF RFC 2373
716 Internet Protocol, Version 6 (IPv6) Addressing Architecture, S. Deering, R. Hinden, July
717 1998.
- 718 IETF RFC 2460
719 Internet Protocol, Version 6 (IPv6), S. Deering, R. Hinden, December 1998.
- 720 Internationalisation Guide
721 Guide, July 1993, Internationalisation Guide, Version 2 (ISBN: 1-859120-02-4, G304),
722 published by The Open Group.
- 723 ISO C (1990)
724 ISO/IEC 9899:1990: Programming Languages — C, including Amendment 1:1995 (E), C
725 Integrity (Multibyte Support Extensions (MSE) for ISO C).
- 726 ISO 2375:1985
727 ISO 2375:1985, Data Processing — Procedure for Registration of Escape Sequences.

- 728 ISO 8652:1987
 729 ISO 8652:1987, Programming Languages — Ada (technically identical to ANSI standard
 730 1815A-1983).
- 731 ISO/IEC 1539:1990
 732 ISO/IEC 1539:1990, Information Technology — Programming Languages — Fortran
 733 (technically identical to the ANSI X3.9-1978 standard [FORTRAN 77]).
- 734 ISO/IEC 4873:1991
 735 ISO/IEC 4873:1991, Information Technology — ISO 8-bit Code for Information Interchange
 736 — Structure and Rules for Implementation.
- 737 ISO/IEC 6429:1992
 738 ISO/IEC 6429:1992, Information Technology — Control Functions for Coded Character
 739 Sets.
- 740 ISO/IEC 6937:1994
 741 ISO/IEC 6937:1994, Information Technology — Coded Character Set for Text
 742 Communication — Latin Alphabet.
 743 ISO 7-bit or 8-bit coded character set for text communication using public communication
 744 networks, private communication networks, or interchange media, such as magnetic tapes
 745 and discs.
- 746 ISO/IEC 8802-3:1996
 747 ISO/IEC 8802-3:1996, Information Technology — Telecommunications and Information
 748 Exchange Between Systems — Local and Metropolitan Area Networks — Specific
 749 Requirements — Part 3: Carrier Sense Multiple Access with Collision Detection
 750 (CSMA/CD) Access Method and Physical Layer Specifications.
- 751 ISO/IEC 8859
 752 ISO/IEC 8859, Information Technology — 8-Bit Single-Byte Coded Graphic Character Sets:
 753 Part 1: Latin Alphabet No. 1
 754 Part 2: Latin Alphabet No. 2
 755 Part 3: Latin Alphabet No. 3
 756 Part 4: Latin Alphabet No. 4
 757 Part 5: Latin/Cyrillic Alphabet
 758 Part 6: Latin/Arabic Alphabet
 759 Part 7: Latin/Greek Alphabet
 760 Part 8: Latin/Hebrew Alphabet
 761 Part 9: Latin Alphabet No. 5
 762 Part 10: Latin Alphabet No. 6
 763 Part 13: Latin Alphabet No. 7
 764 Part 14: Latin Alphabet No. 8
 765 Part 15: Latin Alphabet No. 9
- 766 This standard character set comprises 191 graphic characters covering the requirements of
 767 most of Western Europe.
- 768 ISO POSIX-1:1996
 769 ISO/IEC 9945-1:1996, Information Technology — Portable Operating System Interface
 770 (POSIX) — Part 1: System Application Program Interface (API) [C Language] (identical to
 771 ANSI/IEEE Std 1003.1-1996). Incorporating ANSI/IEEE Stds 1003.1-1990, 1003.1b-1993,
 772 1003.1c-1995, and 1003.1i-1995.
- 773 ISO POSIX-2:1993
 774 ISO/IEC 9945-2:1993, Information Technology — Portable Operating System Interface

Referenced Documents

- 775 (POSIX) — Part 2: Shell and Utilities (identical to ANSI/IEEE Std 1003.2-1992, as amended
776 by ANSI/IEEE Std 1003.2a-1992).
- 777 Issue 1
778 X/Open Portability Guide, July 1985 (ISBN: 0-444-87839-4).
- 779 Issue 2
780 X/Open Portability Guide, January 1987:
- 781 • Volume 1: XVS Commands and Utilities (ISBN: 0-444-70174-5)
 - 782 • Volume 2: XVS System Calls and Libraries (ISBN: 0-444-70175-3)
- 783 Issue 3
784 X/Open Specification, 1988, 1989, February 1992:
- 785 • Commands and Utilities, Issue 3 (ISBN: 1-872630-36-7, C211); this specification was
786 formerly X/Open Portability Guide, Issue 3, Volume 1, January 1989, XSI Commands
787 and Utilities (ISBN: 0-13-685835-X, XO/XPG/89/002)
 - 788 • System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this specification
789 was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System
790 Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003)
 - 791 • Curses Interface, Issue 3, contained in Supplementary Definitions, Issue 3
792 (ISBN: 1-872630-38-3, C213), Chapters 9 to 14 inclusive; this specification was formerly
793 X/Open Portability Guide, Issue 3, Volume 3, January 1989, XSI Supplementary
794 Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)
 - 795 • Headers Interface, Issue 3, contained in Supplementary Definitions, Issue 3
796 (ISBN: 1-872630-38-3, C213), Chapter 19, Cpio and Tar Headers; this specification was
797 formerly X/Open Portability Guide Issue 3, Volume 3, January 1989, XSI Supplementary
798 Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)
- 799 Issue 4
800 CAE Specification, July 1992, published by The Open Group:
- 801 • System Interface Definitions (XBD), Issue 4 (ISBN: 1-872630-46-4, C204)
 - 802 • Commands and Utilities (XCU), Issue 4 (ISBN: 1-872630-48-0, C203)
 - 803 • System Interfaces and Headers (XSH), Issue 4 (ISBN: 1-872630-47-2, C202)
- 804 Issue 4, Version 2
805 CAE Specification, August 1994, published by The Open Group:
- 806 • System Interface Definitions (XBD), Issue 4, Version 2 (ISBN: 1-85912-036-9, C434)
 - 807 • Commands and Utilities (XCU), Issue 4, Version 2 (ISBN: 1-85912-034-2, C436)
 - 808 • System Interfaces and Headers (XSH), Issue 4, Version 2 (ISBN: 1-85912-037-7, C435)
- 809 Issue 5
810 Technical Standard, February 1997, published by The Open Group:
- 811 • System Interface Definitions (XBD), Issue 5 (ISBN: 1-85912-186-1, C605)
 - 812 • Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)
 - 813 • System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)
- 814 Knuth Article
815 Knuth, Donald E., *On the Translation of Languages from Left to Right*, Information and Control,

- 816 Volume 8, No. 6, October 1965.
- 817 KornShell
- 818 Bolsky, Morris I. and Korn, David G., *The New KornShell Command and Programming*
- 819 *Language*, March 1995, Prentice Hall.
- 820 MSE working draft
- 821 Working draft of ISO/IEC 9899:1990/Add3:draft, Addendum 3 — Multibyte Support
- 822 Extensions (MSE) as documented in the ISO Working Paper SC22/WG14/N205 dated 31
- 823 March 1992.
- 824 POSIX.0: 1995
- 825 IEEE Std 1003.0-1995, Guide to the POSIX Open System Environment (OSE) (identical to
- 826 ISO/IEC TR 14252).
- 827 POSIX.1: 1988
- 828 IEEE Std 1003.1-1988, Standard for Information Technology — Portable Operating System
- 829 Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].
- 830 POSIX.1: 1990
- 831 IEEE Std 1003.1-1990, Standard for Information Technology — Portable Operating System
- 832 Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].
- 833 POSIX.1a: 2000
- 834 IEEE Std 1003.1a-2000, Standard for Information Technology — Portable Operating System
- 835 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment
- 836 ????: TITLE?? [C Language].
- 837 POSIX.1d: 1999
- 838 IEEE Std 1003.1d-1999, Standard for Information Technology — Portable Operating System
- 839 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 4:
- 840 Additional Realtime Extensions [C Language].
- 841 POSIX.1g: 2000
- 842 IEEE Std 1003.1g-2000, Standard for Information Technology — Portable Operating System
- 843 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 6:
- 844 Protocol-Independent Interfaces (PII).
- 845 POSIX.1j: 2000
- 846 IEEE Std 1003.1j-2000, Standard for Information Technology — Portable Operating System
- 847 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 5:
- 848 Advanced Realtime Extensions [C Language].
- 849 POSIX.1q: 2000
- 850 IEEE Std 1003.1q-2000, Standard for Information Technology — Portable Operating System
- 851 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment 7:
- 852 Tracing [C Language].
- 853 POSIX.2b:-2000
- 854 IEEE Std 1003.2b: 2000, Standard for Information Technology — Portable Operating System
- 855 Interface (POSIX) — Part 2: Shell and Utilities — Amendment ????: TITLE??.
- 856 POSIX.2d:-1994
- 857 IEEE Std 1003.2d: 1994, Standard for Information Technology — Portable Operating System
- 858 Interface (POSIX) — Part 2: Shell and Utilities — Amendment 1: Batch Environment.
- 859 POSIX.13:-1998
- 860 IEEE Std 1003.13: 1998, Standard for Information Technology — Standardized Application
- 861 Environment Profile (AEP) — POSIX Realtime Application Support.

Referenced Documents

- 862 Sarwate Article
863 Sarwate, Dilip V., *Computation of Cyclic Redundancy Checks via Table Lookup*, Communications
864 of the ACM, Volume 30, No. 8, August 1988.
- 865 Sprunt, Sha, and Lehoczky
866 Sprunt, B., Sha, L., and Lehoczky, J.P., *Aperiodic Task Scheduling for Hard Real-Time Systems*,
867 The Journal of Real-Time Systems, Volume 1, 1989, Pages 27-60.
- 868 SVID, Issue 1
869 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
870 1; Morristown, NJ, UNIX Press, 1985.
- 871 SVID, Issue 2
872 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
873 2; Morristown, NJ, UNIX Press, 1986.
- 874 SVID, Issue 3
875 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
876 3; Morristown, NJ, UNIX Press, 1989.
- 877 The AWK Programming Language
878 Aho, Alfred V., Kernighan, Brian W., and Weinberger, Peter J., *The AWK Programming
879 Language*, Reading, MA, Addison-Wesley 1988.
- 880 UNIX Programmer's Manual
881 American Telephone and Telegraph Company, *UNIX Time-Sharing System: UNIX
882 Programmer's Manual*, 7th Edition, Murray Hill, NJ, Bell Telephone Laboratories, January
883 1979.
- 884 XNS, Issue 4
885 CAE Specification, August 1994, Networking Services, Issue 4 (ISBN: 1-85912-049-0, C438),
886 published by The Open Group.
- 887 XNS, Issue 5
888 CAE Specification, February 1997, Networking Services, Issue 5 (ISBN: 1-85912-165-9, C523),
889 published by The Open Group.
- 890 XNS, Issue 5.2
891 Technical Standard, January 2000, Networking Services (XNS), Issue 5.2
892 (ISBN: 1-85912-241-8, C808), published by The Open Group.
- 893 X/Open Curses, Issue 4, Version 2
894 CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3,
895 C610), published by The Open Group.
- 896 Yacc
897 *Yacc: Yet Another Compiler Compiler*, Stephen C. Johnson, 1978.
- 898 Parts of the following documents were used to create the base documents for
899 IEEE Std 1003.1-200x:
- 900 AIX 3.2 Manual
901 AIX Version 3.2 For RISC System/6000, Technical Reference: Base Operating System and
902 Extensions, 1990, 1992 (Part No. SC23-2382-00).
- 903 OSF/1
904 OSF/1 Programmer's Reference, Release 1.2 (ISBN: 0-13-020579-6).
- 905 OSF AES
906 Application Environment Specification (AES) Operating System Programming Interfaces

- 907 Volume, Revision A (ISBN: 0-13-043522-8).
- 908 System V Release 2.0
- 909 — UNIX System V Release 2.0 Programmer's Reference Manual (April 1984 - Issue 2).
- 910 — UNIX System V Release 2.0 Programming Guide (April 1984 - Issue 2).
- 911 System V Release 4.2
- 912 Operating System API Reference, UNIX SVR4.2 (1992) (ISBN: 0-13-017658-3).