



**INTERNATIONAL STANDARD ISO/IEC/IEEE 9945:2009
TECHNICAL CORRIGENDUM 1**

Published 2013-08-15



INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Portable Operating System Interface (POSIX®) Base Specifications, Issue 7

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Spécifications de base de l'interface pour la portabilité des systèmes (POSIX®), Issue 7

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9945:2009 was prepared by The Open Group and the Portable Applications Standards Committee of the Computer Society of the IEEE (as IEEE Std 1003.1-2008/Cor 1-2013). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*, in parallel with its approval by the ISO/IEC national bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

IEEE Std 1003.1-2008 (Corrigendum to IEEE Std 1003.1-2008)

IEEE Std 1003.1-2008 (Corrigendum to IEEE Std 1003.1-2008)

The Open Group Technical Standard
Base Specifications, Issue 7

IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)

Base Specifications, Issue 7— Technical Corrigendum 1

Sponsor

Portable Applications Standards Committee
of the
IEEE Computer Society

and
The Open Group

Approved 6 February 2013

IEEE-SA Standards Board

Approved 25 October 2012

The Open Group

STANDARDS100.COM: Click to view the full PDF of ISO/IEC/JE945:2009/COR1:2013

Abstract: This Technical Corrigendum addresses problems discovered since the approval of IEEE Std 1003.1™ -2008.

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, IEEE 1003.1™, input/output (I/O), job control, network, portable operating system interface (POSIX®)

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

Copyright © 2013 by The Institute of Electrical and Electronics Engineers, Inc. and The Open Group.
All rights reserved. Published 15 March 2013. Printed in the United States of America.

Published 15 March 2013 by The Open Group. Printed in the United Kingdom by The Open Group.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

POSIX is a registered trademark of IEEE.

The Open Group
Apex Plaza, Forbury Road, Reading, Berkshire RG1 1AX, U.K.

PDF: ISBN 978-0-7381-8265-0 STD698161

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

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher. 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 should be sent by email to austin-group-permissions@opengroup.org.

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. 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 IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. 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 any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, 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 in its standards 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 ten years. When a document is more than ten years old and has not undergone a revision process, 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 its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at <http://standards.ieee.org/develop/wg/>.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to 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 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA website at <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at <http://standards.ieee.org/index.html>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

Participants

IEEE Std 1003.1™-2008/Cor 1-2013 was prepared by the Austin Group, sponsored by the Portable Applications Standards Committee of the IEEE Computer Society, The Open Group, and ISO/IEC JTC 1/SC22.

The Austin Group

At the time this IEEE standard was completed, the Austin Group had the following membership:

Andrew Josey, *Chair*

Donald W. Cragun, *Organizational Representative, IEEE PASC*
Nicholas Stoughton, *Organizational Representative, ISO/IEC JTC 1/SC22*
Mark S. Brown, *Organizational Representative, The Open Group*

Cathy Fox, *Technical Editor*

Austin Group Technical Reviewers

Eric Blake
 Paolo Bonzini
 Mark S. Brown
 Geoff Clare
 Donald W. Cragun
 Ulrich Drepper
 Laszlo Ersek
 Hal Finkel
 Mark Harris
 Tom Honermann

Ross Johnson
 Andrew Josey
 Alexey Khoroshilov
 Antoine Leca
 Vincent Lefevre
 Matt McCutchen
 Christoph Anton Mitterer
 Peter O'Gorman
 Peter Petrov
 James C. Pugsley

Jörg Schilling
 Ed Schouten
 Konrad Schwarz
 Martin Sebor
 Nicolas Sitbon
 Nicholas Stoughton
 Jilles Tjoelker
 Nathan Weeks
 David A. Wheeler

Austin Group Working Group Members

Hans Aberg
 Eitan Adler
 Tanaka Akira
 Matthias Andree
 Matt W. Benjamin
 Guido Berhoerster
 Eric Blake
 Hans Boehm
 Paolo Bonzini
 Davide Brini
 Andries E. Brouwer
 Mark S. Brown
 David Butenhof
 Albert Cahalan
 Stephane Chazelas
 Geoff Clare
 Garrett Cooper
 Donald W. Cragun
 Matthew Dempsky
 Casper Dik
 Niall Douglas
 Ulrich Drepper

Lawrence D.K.B. Dwyer
 Paul Eggert
 Julian Elischer
 Laszlo Ersek
 Bruce Evans
 Sean C. Farley
 Hal Finkel
 Glenn Fowler
 Mike Frysinger
 Pierre Gaston
 Thorsten Glaser
 Philip Guenther
 Bruno Haible
 Mark Harris
 David Holland
 Tom Honermann
 Ross Johnson
 Andrew Josey
 Dan Kegel
 Michael Kerrisk
 Alexey Khoroshilov
 Josh Knight

Jonathan A. Kollasch
 Bruce Korb
 David Korn
 Terry Lambert
 Antoine Leca
 Vincent Lefevre
 Wojtek Lerch
 Scott Lurndal
 Roger Marquis
 Matt McCutchen
 Esmail Mirzaee
 Christoph Anton Mitterer
 Joseph S. Myers
 Alexander Nasonov
 Jonathan Nieder
 Carlos O'Donnell
 Peter O'Gorman
 Peter Petrov
 Wayne Pollock
 James C. Pugsley
 Chet Ramey
 Xavier Roche

Jeroen Ruigrok van der Werven
Bart Schaefer
Jörg Schilling
Nico Schottelius
Ed Schouten
Konrad Schwarz
Jens Schweikhardt
Martin Sebor

Glen Seeds
Thor Lancelot Simon
Keld Simonsen
Nicolas Sitbon
Johannes Sixt
Paul Smith
Nicholas Stoughton
Marcel Telka

Jilles Tjoelker
Fred J. Tydeman
Nathan Weeks
David A. Wheeler
Mats D Wichmann
Garrett Wollman
Jörg Wunsch

The Open Group

When The Open Group approved the Base Specifications, Issue 7, Technical Corrigendum 1 on **25 October 2012**, the membership of The Open Group Base Working Group was as follows:

Andrew Josey, *Chair*
Mark S. Brown, *Austin Group Liaison*

Cathy Fox, *Technical Editor*

Base Working Group Members

Eric Blake
Mark S. Brown
David Butenhof
Geoff Clare

Donald W. Cragun
Lawrence D.K.B. Dwyer
Darrin Johnson
James C. Pugsley

William L. Toth
Kevin Van Vechten

Portable Applications Standards Committee (PASC)

Joseph M. Gwinn, *Chair*
Andrew Josey, *Functional Chair (Interpretations)*
Curtis Royster Jr., *Functional Chair (Logistics)*
Nicholas Stoughton, *Secretary*

IEEE

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Mark S. Brown
Michael Browne
Keith Chow
Donald W. Cragun
Thomas Dineen
Andrew Fieldsend
David Fuschi

Randall Groves
Joseph Gwinn
Werner Hoelzl
Andrew Josey
Piotr Karocki
Fernando Lucas Rodriguez
Greg Luri

Peter Petrov
Bartien Sayogo
Stephen Schwarm
Gil Shultz
Walter Struppler
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 6 February 2013, it had the following membership:

John Kulick, *Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
 Peter Balma
 Farooq Bari
 Ted Burse
 Wael William Diab
 Stephen Dukes
 Jean-Philippe Faure
 Alexander Gelman

Mark Halpin
 Gary Hoffman
 Paul Houzé
 Jim Hughes
 Michael Janezic
 Joseph L. Koepfinger*
 David J. Law
 Oleg Logvinov

Ron Petersen
 Gary Robinson
 Jon Walter Rosdahl
 Adrian Stephens
 Peter Sutherland
 Yatin Trivedi
 Phil Winston
 Yu Yuan

*Member Emeritus

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

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina, *IEEE Standards Program Manager, Document Development*

Michael Kipness, *IEEE Standards Program Manager, Technical Program Development*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

Introduction

This introduction is not part of IEEE Std 1003.1™-2008/Cor1-2013 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®) Base Specifications, Issue 7—Technical Corrigendum 1.

This Technical Corrigendum addresses issues raised in defect reports and interpretation requests submitted up to 17 May 2011 that meet all of the following criteria: (a) They are in the scope of the approved standard. (b) They contain no new APIs (functions/utilities); however, they may add enumeration symbols, non-function #defines, and reserve additional namespaces. (c) They address contradictions between different parts of the standard, or add consistency between the standard and overriding standards, or address security-related problems.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

Contents

1. Changes to Base Definitions.....	2
2. Changes to System Interfaces.....	27
3. Changes to Shell and Utilities	260
4. Changes to Rationale.....	341

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 9945:2009/COR1:2013

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

Base Specifications, Issue 7— Technical Corrigendum 1

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers.

These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents”. They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

NOTE: The editing instructions contained in this corrigendum define how to merge the material contained therein into the existing base standard to form the comprehensive standard.

28 1. Changes to Base Definitions

29 This section contains the set of changes to the text of the Base Definitions.

30 *[Note to reviewers: References to defect reports are provided to aid reviewers.]*

31 **Change Number: Frontmatter/TC1/D5/0001 [246]**

32
33 On Page: ix Line: 0 Section: Typographical Conventions

34

35 In the table of conventions, first column, change from:

36

37 Filename

38

39 to:

40

41 Filename, Pathname

42

43 *Rationale:* Austin Group Defect Report(s) applied: 246.

44 See <http://austingroupbugs.net/view.php?id=246>

45

46

47 **Change Number: Frontmatter/TC1/D5/0002 [341]**

48

49 On Page: ix Line: none Section: Typographical Conventions

50

51 Change:

52

53 '\\'

54

55 in Note 2 to:

56

57 '\\''

58

59 *Rationale:* Austin Group Defect Report(s) applied: 341.

60 See <http://austingroupbugs.net/view.php?id=341>

61

62

63 **Change Number: XBD/TC1/D5/0001 [68]**

64

65

66 On Page: 9 Line: 247 Section: 1.7.1 Codes

67

68 Change from:

69

70 [MX]IEC 60559 Floating-Point

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

73

74 Where applicable, functions are marked with the MX margin legend in
75 the SYNOPSIS section. Where additional semantics apply to a function,
76 the material is identified by use of the MX margin legend.[/MX]

77

78 to:

79

80 [MX]IEC 60559 Floating-Point

81 The functionality described is optional. The functionality described
82 is mandated by the ISO C standard only for implementations that define
83 `__STDC_IEC_559__`.[/MX]

84

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

85 [MXX]IEC 60559 Floating-Point Extension
 86 The functionality described is part of the IEC 60559 Floating-Point
 87 option, but is an extension to the ISO C standard.[/MXX]
 88
 89 *Rationale:* Austin Group Defect Report(s) applied: 68.
 90 See <http://austingroupbugs.net/view.php?id=68>
 91
 92
 93 **Change Number: XBD/TC1/D5/0002 [291]**
 94
 95
 96 On Page: 21 Line: 730 Section: 2.1.5.1 Subprofiling Considerations
 97
 98 Change from:
 99
 100 filename argument
 101
 102 to:
 103
 104 pathname argument
 105
 106 *Rationale:* Austin Group Defect Report(s) applied: 291.
 107 See <http://austingroupbugs.net/view.php?id=291>
 108
 109
 110 **Change Number: XBD/TC1/D5/0003 [82]**
 111
 112
 113 On Page: 24 Line: 830,839 Section: 2.1.5.2 XSI Option Groups
 114
 115 Add to the list of options (in alphabetical order):
 116
 117 `_POSIX_THREAD_ROBUST_PRIO_INHERIT`
 118 `_POSIX_THREAD_ROBUST_PRIO_PROTECT`
 119
 120 *Rationale:* Austin Group Defect Report(s) applied: 82.
 121 See <http://austingroupbugs.net/view.php?id=82>
 122
 123
 124 **Change Number: XBD/TC1/D5/0004 [291]**
 125
 126
 127 On Page: 54 Line: 1647 Section: 3.136 Dot
 128
 129 Change from:
 130
 131 In the context of naming files, the filename consisting of a single dot
 132 character ('.').
 133
 134 to:
 135
 136 In the context of naming files, the filename consisting of a single
 137 <period> character ('.').
 138
 139 *Rationale:* Austin Group Defect Report(s) applied: 291.
 140 See <http://austingroupbugs.net/view.php?id=291>
 141
 142
 143 **Change Number: XBD/TC1/D5/0005 [291]**
 144
 145
 146 On Page: 55 Line: 1650 Section: 3.137 Dot-Dot
 147

148 Change from:

149

150 The filename consisting solely of two dot characters ("..").

151

152 to:

153

154 The filename consisting solely of two <period> characters ("..").

155

156 *Rationale:* Austin Group Defect Report(s) applied: 291.

157 See <http://austingroupbugs.net/view.php?id=291>

158

159

160 **Change Number:** XBD/TC1/D5/0006 [371,425]

161

162

163 On Page: 60 Line: 1763 Section: 3.166 File Descriptor

164

165 Change from:

166

167 The value of a file descriptor is from zero to {OPEN_MAX}. A process
168 can have no more than {OPEN_MAX} file descriptors open simultaneously.

169

170 to:

171

172 The value of a newly-created file descriptor is from zero to
173 {OPEN_MAX}-1. A file descriptor can have a value greater than or equal
174 to {OPEN_MAX} if the value of {OPEN_MAX} has decreased (see `sysconf()`)
175 since the file descriptor was opened.

176

177 *Rationale:* Austin Group Defect Report(s) applied: 371,425.

178 See <http://austingroupbugs.net/view.php?id=371>

179 See <http://austingroupbugs.net/view.php?id=425>

180

181

182 **Change Number:** XBD/TC1/D5/0007 [291]

183

184

185 On Page: 60 Line: 1782 Section: 3.170 Filename

186

187 Change from:

188

189 A name consisting of 1 to {NAME_MAX} bytes used to name a file. The
190 characters composing the name may be selected from the set of all
191 character values excluding the <slash> character and the null byte. The
192 filenames dot and dot-dot have special meaning. A filename is sometimes
193 referred to as a "pathname component".

194

195 to:

196

197 A sequence of bytes consisting of 1 to {NAME_MAX} bytes used to name
198 a file. The bytes composing the name shall not contain the <NUL> or
199 <slash> characters. In the context of a pathname, each filename shall
200 be followed by a <slash> or a <NUL> character; elsewhere, a filename
201 followed by a <NUL> character forms a string (but not necessarily a
202 character string). The filenames dot and dot-dot have special meaning. A
203 filename is sometimes referred to as a "pathname component". See also
204 XBD Section 3.266 Pathname.

205

206 *Rationale:* Austin Group Defect Report(s) applied: 291.

207 See <http://austingroupbugs.net/view.php?id=291>

208

209

210 **Change Number:** XBD/TC1/D5/0008 [291]

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273

On Page: 60 Line: 1786 Section: 3.170

After Section 3.170, add a new section and then renumber existing sections:

3.1xx Filename String

A string consisting of a filename followed by a <NUL> character.

Rationale: Austin Group Defect Report(s) applied: 291.

See <http://austingroupbugs.net/view.php?id=291>

Change Number: XBD/TC1/D5/0009 [291]

On Page: 75 Line: 2145 Section: 3.266 Pathname

Change from:

A character string that is used to identify a file. In the context of POSIX.1-2008, a pathname may be limited to {PATH_MAX} bytes, including the terminating null byte. It has an optional beginning <slash>, followed by zero or more filenames separated by <slash> characters. A pathname may optionally contain one or more trailing <slash> characters. Multiple successive <slash> characters are considered to be the same as one <slash>, except for the case of exactly two leading <slash> characters.

to:

A string that is used to identify a file. In the context of <current version>, a pathname may be limited to {PATH_MAX} bytes, including the terminating null byte. It has optional beginning <slash> characters, followed by zero or more filenames separated by <slash> characters. A pathname can optionally contain one or more trailing <slash> characters. Multiple successive <slash> characters are considered to be the same as one <slash>, except for the case of exactly two leading <slash> characters.

Note: If a pathname consists of only bytes corresponding to characters from the portable filename character set (see XBD Section 3.276), <slash> characters, and a single terminating <NUL> character, the pathname will be usable as a character string in all supported locales; otherwise, the pathname might only be a string (rather than a character string). Additionally, since the single-byte encoding of the <slash> character is required to be the same across all locales and to not occur within a multi-byte character, references to a <slash> character within a pathname are well-defined even when the pathname is not a character string. However, this property does not necessarily hold for the remaining characters within the portable filename character set.

Rationale: Austin Group Defect Report(s) applied: 291.

See <http://austingroupbugs.net/view.php?id=291>

Change Number: XBD/TC1/D5/0010 [291]

On Page: 77 Line: 2193 Section: 3.275

After Section 3.275, add a new section and then renumber existing

274 sections:

275

276 3.2xx Portable Filename

277

278 A filename consisting only of characters from the portable filename
279 character set.

280

281 Note: Applications should avoid using filenames that have the <hyphen>
282 character as the first character since this may cause problems when
283 filenames are passed as command line arguments.

284

285 *Rationale:* Austin Group Defect Report(s) applied: 291.

286 See <http://austingroupbugs.net/view.php?id=291>

287

288

289 **Change Number: XBD/TC1/D5/0011 [291]**

290

291

292 On Page: 77 Line: 2199 Section: 3.276 Portable Filename Character Set

293

294 Add a sentence at the end of the section:

295

296 See also XBD Section 3.266 Pathname.

297

298 *Rationale:* Austin Group Defect Report(s) applied: 291.

299 See <http://austingroupbugs.net/view.php?id=291>

300

301

302 **Change Number: XBD/TC1/D5/0012 [181]**

303

304

305 On Page: 97 Line: 2670 Section: 3.396 Thread

306

307 Change from:

308

309 Each thread has its own thread ID, scheduling priority and policy,
310 errno value, thread-specific key/value bindings, and the required system
311 resources to support a flow of control.

312

313 to:

314

315 Each thread has its own thread ID, scheduling priority and policy, errno
316 value, floating point environment, thread-specific key/value bindings,
317 and the required system resources to support a flow of control.

318

319 *Rationale:* Austin Group Defect Report(s) applied: 181.

320 See <http://austingroupbugs.net/view.php?id=181>

321

322

323 **Change Number: XBD/TC1/D5/0013 [186]**

324

325

326 On Page: 98 Line: 2685 Section: 3.399 Thread-Safe

327

328 Change from:

329

330 A function that may be safely invoked concurrently by multiple threads.

331

332 to:

333

334 A thread-safe function can be safely invoked concurrently with other calls
335 to the same function, or with calls to any other thread-safe functions,
336 by multiple threads.

337

338 *Rationale:* Austin Group Defect Report(s) applied: 186.339 See <http://austingroupbugs.net/view.php?id=186>

340

341

342 **Change Number:** XBD/TC1/D5/0014 [135]

343

344

345 On Page: 108 Line: 2900 Section: 4.4 File Access Permissions

346

347 Delete the last paragraph in the section:

348

349 POSIX.1-2008 does not provide a way to open a directory for searching. It

350 is unspecified whether directory search permission is granted based on

351 the file access modes of the directory's file descriptor or on the mode

352 of the directory at the time the directory is searched.

353

354 *Rationale:* Austin Group Defect Report(s) applied: 135.355 See <http://austingroupbugs.net/view.php?id=135>356 Text is removed that predated the addition of the `O_SEARCH`

357 flag.

358

359

360 **Change Number:** XBD/TC1/D5/0015 [291]

361

362

363 On Page: 109 Line: 2915 Section: 4.7 Filename Portability

364

365 Change from:

366

367 Portable filenames shall not have the <hyphen> character as the first

368 character since this may cause problems when filenames are passed as

369 command line arguments.

370

371 to:

372

373 Note: Applications should avoid using filenames that have the <hyphen>

374 character as the first character since this may cause problems when

375 filenames are passed as command line arguments.

376

377 *Rationale:* Austin Group Defect Report(s) applied: 291.378 See <http://austingroupbugs.net/view.php?id=291>

379

380

381 **Change Number:** XBD/TC1/D5/0016 [83]

382

383

384 On Page: 112 Line: 3040 Section: 4.12 Pathname Resolution

385

386 Change from:

387

388 A pathname that begins with two successive <slash> characters may be

389 interpreted in an implementation-defined manner, although ...

390

391 to:

392

393 If a pathname begins with two successive <slash> characters, the first

394 component following the leading <slash> characters may be interpreted

395 in an implementation-defined manner, although ...

396

397 *Rationale:* Austin Group Defect Report(s) applied: 83.398 See <http://austingroupbugs.net/view.php?id=83>

399

400

401 **Change Number: XBD/TC1/D5/0017 [340]**

402

403

404 On Page: 121 Line: 3303 Section: 5 File Format Notation

405

406 Change from:

407

408 Escape Sequences represent non-graphic characters.

409

410 to:

411

412 Escape Sequences represent non-graphic characters and the escape
413 character (<backslash>).

414

415 *Rationale:* Austin Group Defect Report(s) applied: 340.416 See <http://austingroupbugs.net/view.php?id=340>

417

418

419 **Change Number: XBD/TC1/D5/0018 [302]**

420

421

422 On Page: 127 Line: 3583 Section: 6.1 Portable Character Set

423

424 Change from:

425

426 ... if an application accesses any pair of locales where the character
427 encodings differ, or accesses data from an application running in a
428 locale which has different encodings from the application's current
429 locale, the results are unspecified.

430

431 to:

432

433 ... if an application uses any pair of locales where the character
434 encodings differ, or accesses data from an application using a locale
435 which has different encodings from the locales used by the application,
436 the results are unspecified.

437

438 *Rationale:* Austin Group Defect Report(s) applied: 302.439 See <http://austingroupbugs.net/view.php?id=302>

440

441

442 **Change Number: XBD/TC1/D5/0019 [291]**

443

444

445 On Page: 128 Line: 3589 Section: 6.1 Portable Character Set

446

447 After L3589 insert a new bullet:

448

449 * The encoded values associated with <slash> and <period> shall be
450 invariant across all locales supported by the implementation.

451

452 *Rationale:* Austin Group Defect Report(s) applied: 291.453 See <http://austingroupbugs.net/view.php?id=291>

454

455

456 **Change Number: XBD/TC1/D5/0020 [216]**

457

458

459 On Page: 128 Line: 3596 Section: 6.2 Character Encoding

460

461 Change from:

462

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

463 The POSIX locale contains the characters in Table 6-1 (on page 125), ...
464
465 to:
466
467 The POSIX locale contains the characters in Table 6-1 (on page 125)
468 and Table 6-2 (on page 130), ...
469
470 *Rationale:* Austin Group Defect Report(s) applied: 216.
471 See <http://austingroupbugs.net/view.php?id=216>
472
473
474 **Change Number:** XBD/TC1/D5/0021 [291]
475
476
477 On Page: 128 Line: 3619 Section: 6.2 Character Encoding
478
479 Add a sentence:
480
481 Likewise, the byte values used to encode <period> and <slash> shall not
482 occur as part of any other character in any locale.
483
484 *Rationale:* Austin Group Defect Report(s) applied: 291.
485 See <http://austingroupbugs.net/view.php?id=291>
486
487
488 **Change Number:** XBD/TC1/D5/0022 [216]
489
490
491 On Page: 129 Line: 3661 Section: 6.4 Character Set Description File
492
493 Change from:
494
495 ... the symbolic names and their corresponding encoding values shall be
496 included in the file.
497
498 to:
499
500 ... the symbolic names and their corresponding encoding values shall be
501 included in the file. The encoding values shall each be represented in
502 a single byte.
503
504 *Rationale:* Austin Group Defect Report(s) applied: 216.
505 See <http://austingroupbugs.net/view.php?id=216>
506
507
508 **Change Number:** XBD/TC1/D5/0023 [302]
509
510
511 On Page: 135 Line: 3820-3822 Section: 7.1 General
512
513 Change from:
514
515 The behavior of some of the C-language functions defined in the System
516 Interfaces volume of POSIX.1-2008 shall also be modified based on the
517 current locale, as defined by the last call to `setlocale()`.
518
519 to:
520
521 The behavior of some of the C-language functions defined in the System
522 Interfaces volume of POSIX.1-2008 shall also be modified based on a locale
523 selection. The locale to be used by these functions can be selected in
524 the following ways:
525

526 1. For functions such as `isalnum_l()` that take a locale object as an
527 argument, a locale object can be obtained from `newlocale()` or `duplocale()`
528 and passed to the function.
529

530 2. For functions that do not take a locale object as an argument, the
531 current locale for the thread can be set by calling `uselocale()` or the
532 global locale for the process can be set by calling `setlocale()`. Such
533 functions shall use the current locale of the calling thread if one has
534 been set for that thread; otherwise, they shall use the global locale.
535

536 *Rationale:* Austin Group Defect Report(s) applied: 302.
537 See <http://austingroupbugs.net/view.php?id=302>
538 These changes were overlooked during the revision when
539 per-thread locales were added.
540

541

542

Change Number: XBD/TC1/D5/0024 [302]

543

544

545 On Page: 135 Line: 3839 Section: 7.1 General

546

547 Change from:

548

549 Applications can select the desired locale by invoking the `setlocale()`
550 function (or equivalent) with the appropriate value. If the function
551 is invoked with an empty string, such as:

552

```
553 setlocale(LC_ALL, "");
```

554

555 the value of the corresponding environment variable is used.

556

557 to:

558

559 Applications can select the desired locale by calling the `newlocale()`
560 or `setlocale()` function with the appropriate value. If the function is
561 invoked with an empty string, such as:

562

```
563 newlocale(LC_ALL_MASK, "", (locale_t)0);
```

564

565 or:

566

```
567 setlocale(LC_ALL, "");
```

568

569 the value of the corresponding environment variable is used.
570

571

572 *Rationale:* Austin Group Defect Report(s) applied: 302.

573 See <http://austingroupbugs.net/view.php?id=302>

574 These changes were overlooked during the revision when

575 per-thread locales were added.
576

577

578

Change Number: XBD/TC1/D5/0025 [22,427]

579

580

On Page: 164 Line: 5141-5146 Section: 7.3.6 LC_MESSAGES

581

582 On L5142 in the description of the `yesexpr` keyword, change from:

583

584 ... the acceptable affirmative response ...

585

586 to:

587

588 ... acceptable affirmative responses ...

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

589
590 On L5144-5145 in the description of the noexpr keyword, change from:
591
592 ... the acceptable negative response ...
593
594 to:
595
596 ... acceptable negative responses ...
597
598 *Rationale:* Austin Group Defect Report(s) applied: 22,427.
599 See <http://austingroupbugs.net/view.php?id=22>
600 See <http://austingroupbugs.net/view.php?id=427>
601
602
603 **Change Number:** XBD/TC1/D5/0026 [167]
604
605
606 On Page: 173 Line: 5476 Section: 8.1 Environment Variable Definition
607
608 Change from:
609
610 ... manipulating the environ variable ...
611
612 to:
613
614 ... assigning a new value to the environ variable ...
615
616 *Rationale:* Austin Group Defect Report(s) applied: 167.
617 See <http://austingroupbugs.net/view.php?id=167>
618
619
620 **Change Number:** XBD/TC1/D5/0027 [167]
621
622
623 On Page: 173 Line: 5478 Section: 8.1 Environment Variable Definition
624
625 Add a new paragraph after L5478:
626
627 If the application modifies the pointers to which environ points, the
628 behavior of all interfaces described in the System Interfaces volume of
629 POSIX.1-2008 is undefined.
630
631 *Rationale:* Austin Group Defect Report(s) applied: 167.
632 See <http://austingroupbugs.net/view.php?id=167>
633
634
635 **Change Number:** XBD/TC1/D5/0028 [291]
636
637
638 On Page: 175 Line: 5566 Section: 8.2 Internationalization Variables
639
640 In the description of NLSPATH, change from:
641
642 filename
643
644 to:
645
646 pathname
647
648 *Rationale:* Austin Group Defect Report(s) applied: 291.
649 See <http://austingroupbugs.net/view.php?id=291>
650
651

652 **Change Number: XBD/TC1/D5/0029** [302]

653

654

655 On Page: 177 Line: 5636 Section: 8.2 Internationalization Variables

656

657 Change from:

658

659 At runtime, these values are bound to the locale of a process by calling
660 the `setlocale()` function.

661

662 to:

663

664 These environment variables are used by the `newlocale()` and `setlocale()`
665 functions, and by the standard utilities.

666

667 *Rationale:* Austin Group Defect Report(s) applied: 302.

668 See <http://austingroupbugs.net/view.php?id=302>

669

670

671 **Change Number: XBD/TC1/D5/0030** [139]

672

673

674 On Page: 178 Line: 5674 Section: 8.3 Other Environment Variables

675

676 In the description of the `PATH` variable, change from:

677

678 When a non-zero-length prefix is applied to this filename, a `<slash>`
679 shall be inserted between the prefix and the filename.

680

681 to:

682

683 When a non-zero-length prefix is applied to this filename, a `<slash>`
684 shall be inserted between the prefix and the filename if the prefix did
685 not end in `<slash>`.

686

687 *Rationale:* Austin Group Defect Report(s) applied: 139.

688 See <http://austingroupbugs.net/view.php?id=139>

689 This change is for consistency with the previous change to `CDPATH`
690 made in Interpretation 1003.1-2001 #199

691

692

693 **Change Number: XBD/TC1/D5/0031** [84]

694

695

696 On Page: 178 Line: 5687 Section: 8.3 Other Environment Variables

697

698 At the end of the `PATH` variable, add a new paragraph:

699

700 Since colon is a separator in this context, directory names that might
701 be used in `PATH` should not include a colon character.

702

703 *Rationale:* Austin Group Defect Report(s) applied: 84.

704 See <http://austingroupbugs.net/view.php?id=84>

705

706

707 **Change Number: XBD/TC1/D5/0032** [305,427]

708

709

710 On Page: 186 Line: 5995 Section: 9.3.5 RE Bracket Expression

711

712 At L5995, add a new paragraph:

713

714 8. If a bracket expression contains at least three list elements, where

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

715 the first and last list element are the same single-character element of
 716 <period>, <equals-sign>, or <colon>, then it is unspecified whether the
 717 bracket expression will be treated as a collating symbol, equivalence
 718 class, or character class, respectively; treated as a matching list
 719 expression; or rejected as an error.
 720
 721 *Rationale:* Austin Group Defect Report(s) applied: 305,427.
 722 See <http://austingroupbugs.net/view.php?id=305>
 723 See <http://austingroupbugs.net/view.php?id=427>
 724
 725
 726 **Change Number: XBD/TC1/D5/0033 [134]**
 727
 728
 729 On Page: 197 Line: 6402 Section: 10.1 Directory Structure and Files
 730
 731 Change from:
 732
 733 /dev/null An infinite data source and data sink. ...
 734
 735 to:
 736
 737 /dev/null An empty data source and infinite data sink. ...
 738
 739 *Rationale:* Austin Group Defect Report(s) applied: 134.
 740 See <http://austingroupbugs.net/view.php?id=134>
 741 Improve the description of /dev/null.
 742
 743
 744 **Change Number: XBD/TC1/D5/0034 [79]**
 745
 746
 747 On Page: 201 Line: 6531-6542 Section: 11.1.4 Terminal Access Control
 748
 749 On L6531 change from:
 750
 751 ... if the reading process is ignoring or blocking the SIGTTIN signal, ...
 752
 753 to:
 754
 755 ... if the reading process is ignoring the SIGTTIN signal or the reading
 756 thread is blocking the SIGTTIN signal, ...
 757
 758 On L6539 change from:
 759
 760 ... and the process is ignoring or blocking the SIGTTOU signal, ...
 761
 762 to:
 763
 764 ... and the process is ignoring the SIGTTOU signal or the writing thread
 765 is blocking the SIGTTOU signal, ...
 766
 767 On L6540 change from:
 768
 769 If TOSTOP is set, and the process group of the writing process is
 770 orphaned, and the writing process is not ignoring or blocking the SIGTTOU
 771 signal, ...
 772
 773 to:
 774
 775 If TOSTOP is set, the process group of the writing process is orphaned,
 776 the writing process is not ignoring the SIGTTOU signal, and the writing
 777 thread is not blocking the SIGTTOU signal, ...

778

779 *Rationale:* Austin Group Defect Report(s) applied: 79.780 See <http://austingroupbugs.net/view.php?id=79>

781

782

783 **Change Number:** XBD/TC1/D5/0035 [271]

784

785

786 On Page: 205 Line: 6710 Section: 11.1.11 Closing A Terminal Device File

787

788 Change from:

789

790 The last process to close a terminal device file shall cause any output
791 to be sent to the device and any input to be discarded.

792

793 to:

794

795 The last process to close a terminal device file shall cause any output
796 to be sent to the device and shall cause any input to be discarded.

797

798 *Rationale:* Austin Group Defect Report(s) applied: 271.799 See <http://austingroupbugs.net/view.php?id=271>

800 The text can mean two things:

801 1. "any output to be sent" and "any input" shall be discarded.

802 2. "any output" will be sent and "any input" shall be discarded.

803

804

805 **Change Number:** XBD/TC1/D5/0036 [79]

806

807

808 On Page: 211 Line: 6962 Section: 11.2.5 Local Modes

809

810 Change from:

811

812 Processes that are blocking or ignoring SIGTTOU signals are excepted
813 and allowed to produce output, and the SIGTTOU signal shall not be sent.

814

815 to:

816

817 If the writing process is ignoring the SIGTTOU signal or the writing
818 thread is blocking the SIGTTOU signal, the process is allowed to produce
819 output, and the SIGTTOU signal shall not be sent.

820

821 *Rationale:* Austin Group Defect Report(s) applied: 79.822 See <http://austingroupbugs.net/view.php?id=79>

823

824

825 **Change Number:** XBD/TC1/D5/0037 [171]

826

827

828 On Page: 216 Line: 7120 Section: 12.2 Utility Syntax Guidelines

829

830 In Guideline 5 change from:

831

832 Options without option-arguments should be accepted when grouped behind
833 one '-' delimiter.

834

835 to:

836

837 One or more options without option-arguments, followed by at most one
838 option that takes an option-argument, should be accepted when grouped
839 behind one '-' delimiter.

840

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

841 *Rationale:* Austin Group Defect Report(s) applied: 171.
 842 See <http://austingroupbugs.net/view.php?id=171>
 843
 844
 845 **Change Number: XBD/TC1/D5/0038 [98]**
 846
 847
 848 On Page: 220 Line: 7234 Section: <aio.h>
 849
 850 In the DESCRIPTION section, add FSC|SIO shading to the declaration
 851 of aio_fsync().
 852
 853 *Rationale:* Austin Group Defect Report(s) applied: 98.
 854 See <http://austingroupbugs.net/view.php?id=98>
 855
 856
 857 **Change Number: XBD/TC1/D5/0039 [291]**
 858
 859
 860 On Page: 231 Line: 7575 Section: <dirent.h>
 861
 862 In the DESCRIPTION section, change from:
 863
 864 d_name[] Name of entry.
 865
 866 to:
 867
 868 d_name[] Filename string of entry.
 869
 870 *Rationale:* Austin Group Defect Report(s) applied: 291.
 871 See <http://austingroupbugs.net/view.php?id=291>
 872
 873
 874 **Change Number: XBD/TC1/D5/0040 [291]**
 875
 876
 877 On Page: 231 Line: 7577 Section: <dirent.h>
 878
 879 In the DESCRIPTION section, change from:
 880
 881 The character array d_name is of unspecified size, but the number of
 882 bytes preceding the terminating null byte shall not exceed {NAME_MAX}.
 883
 884 to:
 885
 886 The array d_name is of unspecified size, but shall contain a filename
 887 of at most {NAME_MAX} bytes followed by a terminating null byte.
 888
 889 *Rationale:* Austin Group Defect Report(s) applied: 291.
 890 See <http://austingroupbugs.net/view.php?id=291>
 891
 892
 893 **Change Number: XBD/TC1/D5/0041 [291]**
 894
 895
 896 On Page: 232 Line: 7612 Section: <dirent.h>
 897
 898 In the RATIONALE section, change from:
 899
 900 ... number of characters provided matches (or only slightly exceeds)
 901 the length of the filename.
 902
 903 to:

904
905 ... number of bytes provided matches (or only slightly exceeds) the
906 length of the filename string.
907
908 *Rationale:* Austin Group Defect Report(s) applied: 291.
909 See <http://austingroupbugs.net/view.php?id=291>
910
911
912 **Change Number: XBD/TC1/D5/0042** [206]
913
914
915 On Page: 232 Line: 7629 Section: <dirent.h>
916
917 In the CHANGE HISTORY section, change from:
918
919 ... fopendir() ...
920
921 to:
922 ... fdopendir() ...
923
924
925 *Rationale:* Austin Group Defect Report(s) applied: 206.
926 See <http://austingroupbugs.net/view.php?id=206>
927
928
929 **Change Number: XBD/TC1/D5/0043** [324]
930
931
932 On Page: 236 Line: 7739 Section: <errno.h>
933
934 In the DESCRIPTION section, change from:
935
936 [ENOTDIR] Not a directory.
937
938 to:
939
940 [ENOTDIR] Not a directory or a symbolic link to a directory.
941
942 *Rationale:* Austin Group Defect Report(s) applied: 324.
943 See <http://austingroupbugs.net/view.php?id=324>
944
945
946 **Change Number: XBD/TC1/D5/0044** [274]
947
948
949 On Page: 239 Line: 7848 Section: <fcntl.h>
950
951 In the DESCRIPTION section, after:
952
953 The <fcntl.h> header shall define the following symbolic constants
954 for use as the file access modes for open(), openat(), and fcntl().
955
956 add:
957
958 The values shall be unique, except that O_EXEC and O_SEARCH may have
959 equal values.
960
961 *Rationale:* Austin Group Defect Report(s) applied: 274.
962 See <http://austingroupbugs.net/view.php?id=274>
963
964
965 **Change Number: XBD/TC1/D5/0045** [78,432]
966

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

967
 968 On Page: 240 Line: 7874-7879 Section: <fcntl.h>
 969
 970 In the DESCRIPTION section, delete L7874 to L7875.
 971
 972 The <fcntl.h> header shall define the following symbolic constants as
 973 values for the flag used by open() and openat():
 974
 975 In the DESCRIPTION section, move L7876 to L7879:
 976
 977 O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall
 978 be set to close the file descriptor upon execution of an exec family
 979 function.
 980
 981 O_DIRECTORY Fail if not a directory.
 982 O_NOFOLLOW Do not follow symbolic links.
 983
 984 into the section at L7829 to L7834, so that it reads:
 985
 986 The <fcntl.h> header shall define the following symbolic constants as file
 987 creation flags for use in the oflag value to open() and openat(). The
 988 values shall be bitwise-distinct and shall be suitable for use in #if
 989 preprocessing directives.
 990
 991 O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall
 992 be set to close the file descriptor upon execution of an exec family
 993 function.
 994
 995 O_CREAT Create file if it does not exist.
 996
 997 O_DIRECTORY Fail if not a directory.
 998
 999 O_EXCL Exclusive use flag.
 1000
 1001 O_NOCTTY Do not assign controlling terminal.
 1002
 1003 O_NOFOLLOW Do not follow symbolic links.
 1004
 1005 O_TRUNC Truncate flag.
 1006
 1007 *Rationale:* Austin Group Defect Report(s) applied: 78,432.
 1008 See <http://austingroupbugs.net/view.php?id=78>
 1009 See <http://austingroupbugs.net/view.php?id=432>
 1010
 1011
 1012 **Change Number: XBD/TC1/D5/0046 [346]**
 1013
 1014
 1015 On Page: 249 Line: 8198 Section: <float.h>
 1016
 1017 In the DESCRIPTION section, add a paragraph within the *_MAX_EXP bullet:
 1018
 1019 [CX]Additionally, FLT_MAX_EXP shall be at least as large as FLT_MANT_DIG,
 1020 DBL_MAX_EXP shall be at least as large as DBL_MANT_DIG, and LDBL_MAX_EXP
 1021 shall be at least as large as LDBL_MANT_DIG; which has the effect that
 1022 FLT_MAX, DBL_MAX, and LDBL_MAX are integral. [/CX]
 1023
 1024 *Rationale:* Austin Group Defect Report(s) applied: 346.
 1025 See <http://austingroupbugs.net/view.php?id=346>
 1026
 1027
 1028 **Change Number: XBD/TC1/D5/0047 [346]**
 1029

1030
1031 On Page: 250 Line: 8224 Section: <float.h>
1032
1033 In the RATIONALE section, change from:
1034
1035 None.
1036
1037 to:
1038
1039 All known hardware floating-point formats satisfy the property that
1040 the exponent range is larger than the number of mantissa digits. The
1041 ISO C standard permits a floating-point format where this property is
1042 not true, such that the largest finite value would not be integral;
1043 however, it is unlikely that there will ever be hardware support for
1044 such a floating-point format, and it introduces boundary cases that
1045 portable programs should not have to be concerned with (for example,
1046 a non-integral DBL_MAX means that ceil() would have to worry about
1047 overflow). Therefore, this standard imposes an additional requirement
1048 that the largest representable finite value is integral.
1049
1050 *Rationale:* Austin Group Defect Report(s) applied: 346.
1051 See <http://austingroupbugs.net/view.php?id=346>
1052
1053 **Change Number: XBD/TC1/D5/0048 [403]**
1054
1055
1056 On Page: 254 Line: 8333 Section: <ftw.h>
1057
1058 In the DESCRIPTION section, change from:
1059
1060 FTW_F File.
1061
1062 to:
1063
1064 FTW_F Non-directory file.
1065
1066 *Rationale:* Austin Group Defect Report(s) applied: 403.
1067 See <http://austingroupbugs.net/view.php?id=403>
1068
1069
1070 **Change Number: XBD/TC1/D5/0049 [24]**
1071
1072
1073 On Page: 258 Line: 8441-8442 Section: <grp.h>
1074
1075 In the DESCRIPTION, add XSI shading to endgrent() and getgrent().
1076
1077 *Rationale:* Austin Group Defect Report(s) applied: 24.
1078 See <http://austingroupbugs.net/view.php?id=24>
1079
1080
1081 **Change Number: XBD/TC1/D5/0050 [211]**
1082
1083
1084 On Page: 261 Line: 8510- Section: <inttypes.h>
1085
1086 In the DESCRIPTION section change from:
1087
1088 The <inttypes.h> header shall define at least the following type:
1089
1090 imaxdiv_t Structure type that is the type of the value returned by the
1091 imaxdiv() function.
1092

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

1093
1094 to:
1095
1096 The <inttypes.h> header shall define at least the following types:
1097
1098 imaxdiv_t Structure type that is the type of the value returned by the
1099 imaxdiv() function.
1100
1101 [CX]wchar_t As described in <stddef.h>.[/CX]
1102
1103 *Rationale:* Austin Group Defect Report(s) applied: 211.
1104 See <http://austingroupbugs.net/view.php?id=211>
1105
1106
1107 **Change Number: XBD/TC1/D5/0051 [107]**
1108
1109
1110 On Page: 264 Line: 8623 Section: <langinfo.h>
1111
1112 In the DESCRIPTION section, replace <nl/types.h> with <nl_types.h>.
1113
1114 *Rationale:* Austin Group Defect Report(s) applied: 107.
1115 See <http://austingroupbugs.net/view.php?id=107>
1116 Corrected an editorial error in referencing <nl_types.h>.
1117
1118
1119 **Change Number: XBD/TC1/D5/0052 [108]**
1120
1121
1122 On Page: 268 Line: 8756 Section: <limits.h>
1123
1124 In the DESCRIPTION section, change from:
1125
1126 Many of the symbols listed here are not defined by the ISO/IEC 9899:
1127 1999 standard. Such symbols are not shown as CX shaded.
1128
1129 to:
1130
1131 (Shading block as follows on L8756-8757)
1132
1133 Many of the symbols listed here are not defined by the ISO/IEC 9899:
1134 1999 standard. Such symbols are not shown as CX shaded, except under
1135 the heading "Numerical Limits".
1136
1137 *Rationale:* Austin Group Defect Report(s) applied: 108.
1138 See <http://austingroupbugs.net/view.php?id=108>
1139
1140
1141 **Change Number: XBD/TC1/D5/0053 [291]**
1142
1143
1144 On Page: 272 Line: 8933 Section: <limits.h>
1145
1146 In the DESCRIPTION (<limits.h> {NAME_MAX}) change from:
1147
1148 ... bytes in a filename (not including the terminating null).
1149
1150 to:
1151
1152 ... bytes in a filename (not including the terminating null of a filename
1153 string).
1154
1155 *Rationale:* Austin Group Defect Report(s) applied: 291.

1156 See <http://austingroupbugs.net/view.php?id=291>
1157
1158
1159 **Change Number: XBD/TC1/D5/0054** [182,427]
1160
1161
1162 On Page: 274 Line: 9020 Section: <limits.h>
1163
1164 In the DESCRIPTION section, Minimum Values, change from:
1165
1166 A conforming implementation shall provide values at least this large.
1167
1168 to:
1169
1170 For each of these limits, a conforming implementation shall provide a
1171 value at least this large or shall have no limit.
1172
1173 *Rationale:* Austin Group Defect Report(s) applied: 182,427.
1174 See <http://austingroupbugs.net/view.php?id=182>
1175 See <http://austingroupbugs.net/view.php?id=427>
1176
1177
1178 **Change Number: XBD/TC1/D5/0055** [291]
1179
1180
1181 On Page: 275 Line: 9060 Section: <limits.h>
1182
1183 In the DESCRIPTION ({_POSIX_NAME_MAX}), change from:
1184
1185 ... bytes in a filename (not including the terminating null).
1186
1187 to:
1188
1189 ... bytes in a filename (not including the terminating null of a filename
1190 string).
1191
1192 *Rationale:* Austin Group Defect Report(s) applied: 291.
1193 See <http://austingroupbugs.net/view.php?id=291>
1194
1195
1196 **Change Number: XBD/TC1/D5/0056** [371]
1197
1198
1199 On Page: 275 Line: 9066 Section: <limits.h>
1200
1201 In the DESCRIPTION section, change from:
1202
1203 Maximum number of files that one process can have open at any one time.
1204
1205 to:
1206
1207 A value one greater than the maximum value that the system may assign
1208 to a newly-created file descriptor.
1209
1210 *Rationale:* Austin Group Defect Report(s) applied: 371.
1211 See <http://austingroupbugs.net/view.php?id=371>
1212
1213
1214 **Change Number: XBD/TC1/D5/0057** [291]
1215
1216
1217 On Page: 277 Line: 9182 Section: <limits.h>
1218

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

1219 In the DESCRIPTION ({_XOPEN_NAME_MAX}), change from:
 1220
 1221 ... bytes in a filename (not including the terminating null).
 1222
 1223 to:
 1224
 1225 ... bytes in a filename (not including the terminating null of a filename
 1226 string).
 1227
 1228 *Rationale:* Austin Group Defect Report(s) applied: 291.
 1229 See <http://austingroupbugs.net/view.php?id=291>
 1230
 1231
 1232 **Change Number: XBD/TC1/D5/0058 [108]**
 1233
 1234
 1235 On Page: 278 Line: 9218 Section: <limits.h>
 1236
 1237 In the DESCRIPTION section, add CX shading to:
 1238
 1239 LONG_BIT (P278 L9218-9220)
 1240 WORD_BIT (P279 L9260-9262)
 1241 SSIZE_MAX (P279 L9242-9244)
 1242
 1243 *Rationale:* Austin Group Defect Report(s) applied: 108.
 1244 See <http://austingroupbugs.net/view.php?id=108>
 1245
 1246
 1247 **Change Number: XBD/TC1/D5/0059 [291]**
 1248
 1249
 1250 On Page: 280 Line: 9311 Section: <limits.h>
 1251
 1252 In the RATIONALE section, change "filenames" to "pathnames".
 1253
 1254 *Rationale:* Austin Group Defect Report(s) applied: 291.
 1255 See <http://austingroupbugs.net/view.php?id=291>
 1256
 1257
 1258 **Change Number: XBD/TC1/D5/0060 [301,427]**
 1259
 1260
 1261 On Page: 284 Line: 9450 Section: <locale.h>
 1262
 1263 In the DESCRIPTION section change from:
 1264
 1265 ... a special object descriptor used by the uselocale() function.
 1266
 1267 to:
 1268
 1269 ... a special object descriptor used by the duplocale() and uselocale()
 1270 functions.
 1271
 1272 *Rationale:* Austin Group Defect Report(s) applied: 301,427.
 1273 See <http://austingroupbugs.net/view.php?id=301>
 1274 See <http://austingroupbugs.net/view.php?id=427>
 1275
 1276
 1277 **Change Number: XBD/TC1/D5/0061 [355]**
 1278
 1279
 1280 On Page: 303 Line: 10141 Section: <netinet/in.h>
 1281

1282 In the DESCRIPTION section, change from:
1283
1284 Values of this type shall be cast by applications to struct sockaddr
1285 for use with socket functions.
1286
1287 to:
1288
1289 Pointers to this type shall be cast by applications to struct sockaddr *
1290 for use with socket functions.
1291
1292 *Rationale:* Austin Group Defect Report(s) applied: 355.
1293 See <http://austingroupbugs.net/view.php?id=355>
1294
1295
1296 **Change Number: XBD/TC1/D5/0062 [208]**
1297
1298
1299 On Page: 328 Line: 10963 Section: <signal.h>
1300
1301 In the DESCRIPTION section, change the margin marker notation for the
1302 SIG_HOLD macro from:
1303
1304 OB CX
1305
1306 to:
1307
1308 OB XSI
1309
1310 *Rationale:* Austin Group Defect Report(s) applied: 208.
1311 See <http://austingroupbugs.net/view.php?id=208>
1312 SIG_HOLD shading and margin marker notation is updated from
1313 OB CX to OB XSI as it is used with the sigset function which is an
1314 XSI function.
1315
1316
1317 **Change Number: XBD/TC1/D5/0063 [80]**
1318
1319
1320 On Page: 330 Line: 11032 Section: <signal.h>
1321
1322 In the DESCRIPTION section, in the table of signals, change the shading
1323 on SIGPROF from OB XSR to OB XSI.
1324
1325 *Rationale:* Austin Group Defect Report(s) applied: 80.
1326 See <http://austingroupbugs.net/view.php?id=80>
1327
1328
1329 **Change Number: XBD/TC1/D5/0064 [157]**
1330
1331
1332 On Page: 331 Line: 11055 Section: <signal.h>
1333
1334 In the DESCRIPTION section, change the shading margin marker from XSI
1335 to CX on the following text:
1336
1337 The storage occupied by sa_handler and sa_sigaction may overlap, and a
1338 conforming application shall not use both simultaneously.
1339
1340 *Rationale:* Austin Group Defect Report(s) applied: 157.
1341 See <http://austingroupbugs.net/view.php?id=157>
1342
1343
1344 **Change Number: XBD/TC1/D5/0065 [291,427]**

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

1345

1346

1347 On Page: 351 Line: 11796 Section: <stdio.h>

1348

1349 In the DESCRIPTION section, at P351 L11796 (XBD <stdio.h> FILENAME_MAX),
 1350 change from "filename string" to "pathname".

1351

1352 *Rationale:* Austin Group Defect Report(s) applied: 291,427.1353 See <http://austingroupbugs.net/view.php?id=291>1354 See <http://austingroupbugs.net/view.php?id=427>

1355

1356

1357 **Change Number:** XBD/TC1/D5/0066 [197]

1358

1359

1360 On Page: 356 Line: 11992 Section: <stdlib.h>

1361

1362 In the DESCRIPTION section, add the CX margin marker and shading to the
 1363 getsubopt() function prototype.

1364

1365 *Rationale:* Austin Group Defect Report(s) applied: 197.1366 See <http://austingroupbugs.net/view.php?id=197>

1367 The getsubopt() function is an extension to the ISO C standard.

1368

1369

1370 **Change Number:** XBD/TC1/D5/0067 [355]

1371

1372

1373 On Page: 382 Line: 12813-12817 Section: <sys/socket.h>

1374

1375 In the DESCRIPTION section, change from:

1376

1377 When a sockaddr_storage structure is cast as a sockaddr structure,
 1378 the ss_family field of the sockaddr_storage structure shall map onto
 1379 the sa_family field of the sockaddr structure. When a sockaddr_storage
 1380 structure is cast as a protocol-specific address structure, the ss_family
 1381 field shall map onto a field of that structure that is of type sa_family_t
 1382 and that identifies the protocol's address family.

1383

1384 to:

1385

1386 When a pointer to a sockaddr_storage structure is cast as a pointer to a
 1387 sockaddr structure, the ss_family field of the sockaddr_storage structure
 1388 shall map onto the sa_family field of the sockaddr structure. When
 1389 a pointer to a sockaddr_storage structure is cast as a pointer to a
 1390 protocol-specific address structure, the ss_family field shall map onto
 1391 a field of that structure that is of type sa_family_t and that identifies
 1392 the protocol's address family.

1393

1394 *Rationale:* Austin Group Defect Report(s) applied: 355.1395 See <http://austingroupbugs.net/view.php?id=355>

1396

1397

1398 **Change Number:** XBD/TC1/D5/0068 [207]

1399

1400

1401 On Page: 388 Line: 13042 Section: <sys/stat.h>

1402

1403 In the DESCRIPTION section, add the XSI margin marker, and shade blkcnt_t
 1404 and blksize_t.

1405

1406 *Rationale:* Austin Group Defect Report(s) applied: 207.1407 See <http://austingroupbugs.net/view.php?id=207>

1408 Add notation to show that blkcnt_t and blksize_t are XSI
1409 extensions.
1410
1411
1412 **Change Number: XBD/TC1/D5/0069 [210]**
1413
1414
1415 On Page: 398 Line: 13360f Section: <sys/types.h>
1416
1417 In the DESCRIPTION section, remove the XSI margin marker and shading
1418 from fsblkcnt_t and fsfilcnt_t.
1419
1420 *Rationale:* Austin Group Defect Report(s) applied: 210.
1421 See <http://austingroupbugs.net/view.php?id=210>
1422 The statvfs structure, where fsblkcnt_t and fsfilcnt_t are
1423 used, is now in base.
1424
1425
1426 **Change Number: XBD/TC1/D5/0070 [28]**
1427
1428
1429 On Page: 399 Line: 13389 Section: <sys/types.h>
1430
1431 In the DESCRIPTION section, change from:
1432
1433 <blank name> Also used to identify a trace stream attributes object
1434
1435 to:
1436
1437 trace_attr_t Used to identify a trace stream attributes object
1438
1439 (The existing OB TRC shading should apply to the whole line.)
1440
1441 *Rationale:* Austin Group Defect Report(s) applied: 28.
1442 See <http://austingroupbugs.net/view.php?id=28>
1443
1444
1445 **Change Number: XBD/TC1/D5/0071 [376]**
1446
1447
1448 On Page: 399 Line: 13415 Section: <sys/types.h>
1449
1450 In the DESCRIPTION section, between L13415 and 13416, add a bullet:
1451
1452 * dev_t shall be an integer type.
1453
1454 *Rationale:* Austin Group Defect Report(s) applied: 376.
1455 See <http://austingroupbugs.net/view.php?id=376>
1456
1457
1458 **Change Number: XBD/TC1/D5/0072 [210]**
1459
1460
1461 On Page: 399 Line: 13417 Section: <sys/types.h>
1462
1463 In the DESCRIPTION section, remove the XSI margin marker and shading
1464 from fsblkcnt_t and fsfilcnt_t.
1465
1466 *Rationale:* Austin Group Defect Report(s) applied: 210.
1467 See <http://austingroupbugs.net/view.php?id=210>
1468 The statvfs structure, where fsblkcnt_t and fsfilcnt_t are
1469 used, is now in base.
1470

1471
 1472 **Change Number: XBD/TC1/D5/0073** [327]
 1473
 1474
 1475 On Page: 399 Line: 13420 Section: <sys/types.h>
 1476
 1477 In the DESCRIPTION section, change from:
 1478
 1479 time_t and clock_t shall be integer or real-floating types.
 1480
 1481 to:
 1482
 1483 clock_t shall be an integer or real-floating type. [CX]time_t shall be
 1484 an integer type.[/CX]
 1485
 1486 *Rationale:* Austin Group Defect Report(s) applied: 327.
 1487 See <http://austingroupbugs.net/view.php?id=327>
 1488
 1489
 1490 **Change Number: XBD/TC1/D5/0074** [355]
 1491
 1492
 1493 On Page: 403 Line: 13517 Section: <sys/un.h>
 1494
 1495 In the DESCRIPTION section, change from:
 1496
 1497 Values of this type shall be cast by applications to struct sockaddr
 1498 for use with socket functions.
 1499
 1500 to:
 1501
 1502 Pointers to this type shall be cast by applications to struct sockaddr *
 1503 for use with socket functions.
 1504
 1505 *Rationale:* Austin Group Defect Report(s) applied: 355.
 1506 See <http://austingroupbugs.net/view.php?id=355>
 1507
 1508
 1509 **Change Number: XBD/TC1/D5/0075** [357,427]
 1510
 1511
 1512 On Page: 418 Line: 13996 Section: <tgmath.h>
 1513
 1514 In the DESCRIPTION section, in the table at L13996-13997, scalbln()
 1515 should sort before scalbn().
 1516
 1517 *Rationale:* Austin Group Defect Report(s) applied: 357,427.
 1518 See <http://austingroupbugs.net/view.php?id=357>
 1519 See <http://austingroupbugs.net/view.php?id=427>
 1520
 1521
 1522 **Change Number: XBD/TC1/D5/0076** [212]
 1523
 1524
 1525 On Page: 422 Line: 14153 Section: <time.h>
 1526
 1527 In the DESCRIPTION section, add the CX margin marker notation and shading
 1528 to L14153-14155. (The <time.h> header.... TIMER_ABSTIME ...)
 1529
 1530 *Rationale:* Austin Group Defect Report(s) applied: 212.
 1531 See <http://austingroupbugs.net/view.php?id=212>
 1532
 1533

1534 **Change Number: XBD/TC1/D5/0077 [212]**

1535

1536

1537 On Page: 422 Line: 14159 Section: <time.h>

1538

1539 In the DESCRIPTION section, change from:

1540

1541 ... a progrm defines ...

1542

1543 to:

1544

1545 ... a program defines ...

1546

1547 *Rationale:* Austin Group Defect Report(s) applied: 212.

1548 See <http://austingroupbugs.net/view.php?id=212>

1549 This is an editorial correction.

1550

1551

1552 **Change Number: XBD/TC1/D5/0078 [311]**

1553

1554

1555 On Page: 430 Line: 14446 Section: <unistd.h>

1556

1557 In the DESCRIPTION section, after:

1558

1559 For implementations conforming to POSIX.1-2008, the value shall be
1560 200809L.

1561

1562 add (to the same paragraph):

1563

1564 For profile implementations that define `_POSIX_SUBPROFILE` (see Section
1565 2.1.5.1) in <unistd.h>, `_POSIX2_VERSION` may be left undefined or
1566 be defined with the value -1 to indicate that the Shell and Utilities
1567 volume is not supported. In this case a call to `sysconf(_SC_2_VERSION)`
1568 shall return either 200809L or -1 indicating that the Shell and Utilities
1569 volume is or is not, respectively, supported at runtime.

1570

1571 At L14455 after:

1572

1573 The following symbolic constants, if defined in <unistd.h>, shall have
1574 a value of -1, 0, or greater, unless otherwise specified below.

1575

1576 add:

1577

1578 For profile implementations that define `_POSIX_SUBPROFILE` (see Section
1579 2.1.5.1) in <unistd.h>, constants described below as always having a
1580 value greater than zero need not be defined and, if defined, may have
1581 a value of -1, 0, or greater.

1582

1583 *Rationale:* Austin Group Defect Report(s) applied: 311.

1584 See <http://austingroupbugs.net/view.php?id=311>

1585

1586

1587 **Change Number: XBD/TC1/D5/0079 [209]**

1588

1589

1590 On Page: 439-440 Line: 14870-14875 Section: <unistd.h>

1591

1592 In the DESCRIPTION section, add the XSI margin marker and shading for
1593 L14870 to 14875 (symbolic constants `F_LOCK` ... `F_ULOCK`).

1594

1595 *Rationale:* Austin Group Defect Report(s) applied: 209.

1596 See <http://austingroupbugs.net/view.php?id=209>

1597

1598

1599 **Change Number: XBD/TC1/D5/0080** [360]

1600

1601

1602 On Page: 443 Line: 15048 Section: <unistd.h>

1603

1604 In the DESCRIPTION section, delete L15048 (ctermid() declaration in XBD
1605 <unistd.h>).

1606

1607 At L15124 (<unistd.h>), add a sentence:

1608

1609 Implementations may also include the ctermid() prototype as defined
1610 in <stdio.h>.

1611

1612 Tag the entire paragraph starting on L15124 including the above new
1613 sentence with the OB margin marking.

1614

1615 *Rationale:* Austin Group Defect Report(s) applied: 360.1616 See <http://austingroupbugs.net/view.php?id=360>

1617 Some implementations have had a declaration for ctermid()

1618 and this should still be allowed in Issue 7. Having ctermid() and

1619 pthread_atfork() prototypes in <unistd.h>, however, are namespace

1620 pollution issues that should be corrected in the next revision.

1621

1622

1623 **Change Number: XBD/TC1/D5/0081** [380]

1624

1625

1626 On Page: 455 Line: 15531-15532 Section: <wchar.h>

1627

1628 In the DESCRIPTION section, in the wpcpcy() and wpcncpy() prototype
1629 declarations change from:

1630

1631 restrict*

1632

1633 to:

1634

1635 *restrict

1636

1637 *Rationale:* Austin Group Defect Report(s) applied: 380.1638 See <http://austingroupbugs.net/view.php?id=380>

1639 2. Changes to System Interfaces

1640 This section contains the set of changes to the text of the System Interfaces.

1641 [*Note to reviewers: References to defect reports are provided to aid reviewers.*]1642 **Change Number: XSH/TC1/D5/0001** [316]

1643

1644

1645 On Page: 477 Line: 16210 Section: 2.3 Error Numbers

1646

1647 Change from:

1648

1649 The ERRORS section on each reference page specifies which error conditions

1650 shall be detected by all implementations ("shall fail") and which may

1651 be optionally detected by an implementation ("may fail"). If no error

1652 condition is detected, the action requested shall be successful.

1653

1654 to:

1655

1656 The ERRORS section on each reference page specifies which error conditions
1657 shall be detected by all implementations ("shall fail") and which may
1658 be optionally detected by an implementation ("may fail"). If no error
1659 condition is detected, the action requested shall be successful. If an
1660 error condition is detected, the action requested may have been partially
1661 performed, unless otherwise stated.

1662

1663 *Rationale:* Austin Group Defect Report(s) applied: 316.
1664 See <http://austingroupbugs.net/view.php?id=316>

1665

1666

1667 **Change Number: XSH/TC1/D5/0002 [146]**

1668

1669

1670 On Page: 482 Line: 16419 Section: 2.3 Error Numbers

1671

1672 For the [ENOTDIR] description, change from:

1673

1674 [ENOTDIR] Not a directory. A component of the specified pathname exists,
1675 but it is not a directory, when a directory was expected.

1676

1677 to:

1678

1679 [ENOTDIR] Not a directory. A component of the specified pathname exists,
1680 but it is not a directory, when a directory was expected; or an attempt
1681 was made to create a non-directory file, and the specified pathname
1682 contains at least one non-`<slash>` character and ends with one or more
1683 trailing `<slash>` characters.

1684

1685 *Rationale:* Austin Group Defect Report(s) applied: 146.
1686 See <http://austingroupbugs.net/view.php?id=146>

1687

1688

1689 **Change Number: XSH/TC1/D5/0003 [112]**

1690

1691

1692 On Page: 482 Line: 16430 Section: 2.3 Error Numbers

1693

1694 Change from:

1695

1696 [ENOTSUP] Not supported. The implementation does not support this feature
1697 of the Realtime Option Group.

1698

1699 to:

1700

1701 [ENOTSUP] Not supported. The implementation does not support the requested
1702 feature or value.

1703

1704 *Rationale:* Austin Group Defect Report(s) applied: 112.
1705 See <http://austingroupbugs.net/view.php?id=112>

1706

1707

1708 **Change Number: XSH/TC1/D5/0004 [77]**

1709

1710

1711 On Page: 485 Line: 16549 Section: 2.4.1 Signal Generation and Delivery

1712

1713 Change from:

1714

1715 When any stop signal (SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU) is generated
1716 for a process, any pending SIGCONT signals for that process shall be

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

1717 discarded. Conversely, when SIGCONT is generated for a process, all
 1718 pending stop signals for that process shall be discarded. When SIGCONT is
 1719 generated for a process that is stopped, the process shall be continued,
 1720 even if the SIGCONT signal is blocked or ignored. If SIGCONT is blocked
 1721 and not ignored, it shall remain pending until it is either unblocked
 1722 or a stop signal is generated for the process.

1723
 1724 to:

1725
 1726 When any stop signal (SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU) is generated
 1727 for a process or thread, all pending SIGCONT signals for that process or
 1728 any of the threads within that process shall be discarded. Conversely,
 1729 when SIGCONT is generated for a process or thread, all pending stop
 1730 signals for that process or any of the threads within that process shall
 1731 be discarded. When SIGCONT is generated for a process that is stopped,
 1732 the process shall be continued, even if the SIGCONT signal is ignored by
 1733 the process or is blocked by all threads within the process and there
 1734 are no threads in a call to a sigwait() function selecting SIGCONT.
 1735 If SIGCONT is blocked by all threads within the process, there are no
 1736 threads in a call to a sigwait() function selecting SIGCONT, and SIGCONT
 1737 is not ignored by the process, the SIGCONT signal shall remain pending on
 1738 the process until it is either unblocked by a thread or a thread calls
 1739 a sigwait() function selecting SIGCONT, or a stop signal is generated
 1740 for the process or any of the threads within the process.

1741
 1742 *Rationale:* Austin Group Defect Report(s) applied: 77.
 1743 See <http://austingroupbugs.net/view.php?id=77>

1744

1745

1746 **Change Number: XSH/TC1/D5/0005 [66]**

1747

1748

1749 On Page: 489 Line: 16717 Section: 2.4.3 Signal Actions

1750

1751 In the section "Pointer to a Function", change from:

1752

1753 When signal-catching functions are invoked asynchronously with process
 1754 execution, the behavior of some of the functions defined by this volume
 1755 of POSIX.1-200x is unspecified if they are called from a signal-catching
 1756 function.

1757

1758 to:

1759

1760 If the process is multi-threaded, or if the process is single threaded
 1761 and a signal handler is executed other than as the result of:

1762

1763 * The process calling abort(), raise(), kill(), pthread_kill(), or
 1764 sigqueue() to generate a signal that is not blocked

1765

1766 * A pending signal being unblocked and being delivered before the call
 1767 that unblocked it returns

1768

1769 the behavior is undefined if the signal handler refers to any object
 1770 other than errno with static storage duration other than by assigning a
 1771 value to an object declared as volatile sig_atomic_t, or if the signal
 1772 handler calls any function defined in this standard other than one of
 1773 the functions listed in the following table.

1774

1775 *Rationale:* Austin Group Defect Report(s) applied: 66.

1776 See <http://austingroupbugs.net/view.php?id=66>

1777

1778

1779 **Change Number: XSH/TC1/D5/0006 [102]**

1780

1781

1782 On Page: 489 Line: 16722 Section: 2.4.3 Signal Actions

1783

1784 Add pthread_self() and pthread_kill() to the list of async-signal-safe
1785 functions.

1786

1787 *Rationale:* Austin Group Defect Report(s) applied: 102.1788 See <http://austingroupbugs.net/view.php?id=102>

1789

1790

1791 **Change Number: XSH/TC1/D5/0007 [103]**

1792

1793

1794 On Page: 489 Line: 16722 Section: 2.4.3 Signal Actions

1795

1796 Add pthread_sigmask() to the list of async-signal-safe functions.

1797

1798 *Rationale:* Austin Group Defect Report(s) applied: 103.1799 See <http://austingroupbugs.net/view.php?id=103>

1800

1801

1802 **Change Number: XSH/TC1/D5/0008 [234]**

1803

1804

1805 On Page: 489 Line: 16750 Section: 2.4.3 Signal Actions

1806

1807 In the table of functions that shall be async-signal-safe, add fchdir()
1808 to the list.

1809

1810 *Rationale:* Austin Group Defect Report(s) applied: 234.1811 See <http://austingroupbugs.net/view.php?id=234>

1812

1813

1814 **Change Number: XSH/TC1/D5/0009 [312]**

1815

1816

1817 On Page: 492 Line: 16878 Section: 2.5.1 Interaction of File Descriptors
1818 and Standard I/O Streams

1819

1820 Delete the text:

1821

1822 Otherwise, the result is undefined.

1823

1824 *Rationale:* Austin Group Defect Report(s) applied: 312.1825 See <http://austingroupbugs.net/view.php?id=312>

1826 Since L16842-16843 cover both handles, it is clear that the

1827 statements on L16878 and L16893 are redundant (as well as contradictory).

1828

1829

1830 **Change Number: XSH/TC1/D5/0010 [312]**

1831

1832

1833 On Page: 493 Line: 16893 Section: 2.5.1 Interaction of File Descriptors
1834 and Standard I/O Streams

1835

1836 Delete the text:

1837

1838 If the rules above are not followed, the result is unspecified.

1839

1840 *Rationale:* Austin Group Defect Report(s) applied: 312.1841 See <http://austingroupbugs.net/view.php?id=312>

1842 Since L16842-16843 cover both handles, it is clear that the

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

1843 statements on L16878 and L16893 are redundant (as well as contradictory).
 1844
 1845

1846 **Change Number: XSH/TC1/D5/0011** [345,428]
 1847
 1848

1849 On Page: 497 Line: 17067 Section: 2.7.1 IPC General Description
 1850

1851 Add a new paragraph:
 1852

1853 In addition to the `ipc_perm` structure, each associated data structure
 1854 includes several `time_t` fields for recording timestamps of particular
 1855 operations. When an operation is described as setting a timestamp to
 1856 the current time, that particular timestamp member of the associated
 1857 data structure shall be set to the largest `time_t` value which is not
 1858 greater than the current time.
 1859

1860 *Rationale:* Austin Group Defect Report(s) applied: 345,428.
 1861 See <http://austingroupbugs.net/view.php?id=345>
 1862 See <http://austingroupbugs.net/view.php?id=428>
 1863
 1864

1865 **Change Number: XSH/TC1/D5/0012** [109,429]
 1866
 1867

1868 On Page: 507 Line: 17490 Section: 2.9.1 Thread-Safety
 1869

1870 Add `mblen()` and `mbtowc()` in alphabetic order to the list of functions
 1871 that need not be thread-safe in the table on P507 L17490-17510; remove
 1872 `wcstombs()` from the same table.
 1873

1874 On L17511, change from:
 1875

1876 The `wcrtomb()` and `wcsrtombs()` functions need not be thread-safe if passed
 1877 a NULL `ps` argument.
 1878

1879 to:
 1880

1881 The `mbrlen()`, `mbrtowc()`, `mbsrtowcs()`, `mbsrtowcs()`, `wcrtomb()`,
 1882 `wcsrtombs()`, and `wcsrtombs()` functions need not be thread-safe if passed
 1883 a NULL `ps` argument.
 1884

1885 *Rationale:* Austin Group Defect Report(s) applied: 109,429.
 1886 See <http://austingroupbugs.net/view.php?id=109>
 1887 See <http://austingroupbugs.net/view.php?id=429>
 1888
 1889

1890 **Change Number: XSH/TC1/D5/0013** [89]
 1891
 1892

1893 On Page: 508 Line: 17536-17542 Section: 2.9.3 Thread Mutexes
 1894

1895 On L17536-L17538, change from:
 1896

1897 * It returns successfully from [...] with `m` as the mutex argument.
 1898

1899 to:
 1900

1901 * It calls [...] with `m` as the mutex argument and the call returns zero
 1902 or `EOWNERDEAD`.
 1903

1904 On L17538 (after the 3rd bullet item) add a new bullet item:
 1905

1906 * It calls `pthread_mutex_setprioceiling()` with `m` as the mutex argument
1907 and the call returns `[EOWNERDEAD]`.
1908
1909 On L17539-17542 (last 2 bullet items), change from:
1910
1911 * It returns (successfully or not) from `pthread_cond_wait()` with `m`
1912 as the mutex argument (except as explicitly indicated otherwise for
1913 certain errors).
1914
1915 * It returns (successfully or not) from `pthread_cond_timedwait()` with
1916 `m` as the mutex argument (except as explicitly indicated otherwise for
1917 certain errors).
1918
1919 to:
1920
1921 * It calls `pthread_cond_wait()` with `m` as the mutex argument and the call
1922 returns zero or certain error numbers (see `pthread_cond_timedwait()`).
1923
1924 * It calls `pthread_cond_timedwait()` with `m` as the mutex argument and the
1925 call returns zero or certain error numbers (see `pthread_cond_timedwait()`).
1926
1927 *Rationale:* Austin Group Defect Report(s) applied: 89.
1928 See <http://austingroupbugs.net/view.php?id=89>
1929
1930
1931 **Change Number: XSH/TC1/D5/0014** [410]
1932
1933
1934 On Page: 514 Line: 17760 Section: 2.9.5.2 Cancellation Points
1935
1936 Add `strerror_l()` to the list, between `strerror()` and `strerror_r()`.
1937
1938 Add `strftime_l()` to the list, between `strftime()` and `symlink()`.
1939
1940 *Rationale:* Austin Group Defect Report(s) applied: 410.
1941 See <http://austingroupbugs.net/view.php?id=410>
1942
1943
1944 **Change Number: XSH/TC1/D5/0015** [376]
1945
1946
1947 On Page: 540 Line: 18820 Section: 2.12.1 Defined Types
1948
1949 In the table, change from:
1950
1951 `dev_t` Arithmetic type used for device numbers.
1952
1953 to:
1954
1955 `dev_t` Integer type used for device numbers.
1956
1957 *Rationale:* Austin Group Defect Report(s) applied: 376.
1958 See <http://austingroupbugs.net/view.php?id=376>
1959
1960
1961 **Change Number: XSH/TC1/D5/0016** [327]
1962
1963
1964 On Page: 541 Line: 18867 Section: 2.12.1 Defined Types
1965
1966 Change from:
1967
1968 `time_t` Integer or real-floating type used for time in seconds, as defined

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

1969 in the ISO C standard.
 1970
 1971 to:
 1972
 1973 time_t Integer type used for time in seconds, as defined in the ISO
 1974 C standard.
 1975
 1976 *Rationale:* Austin Group Defect Report(s) applied: 327.
 1977 See <http://austingroupbugs.net/view.php?id=327>
 1978
 1979
 1980 **Change Number: XSH/TC1/D5/0017 [74]**
 1981
 1982
 1983 On Page: 541 Line: 18881-18886 Section: 2.12.3 Pointer Types
 1984
 1985 Delete Section 2.12.3:
 1986
 1987 2.12.3 Pointer Types
 1988
 1989 All function pointer types shall have the same representation as the
 1990 type pointer to void. Conversion of a function pointer to void * shall
 1991 not alter the representation. A void * value resulting from such a
 1992 conversion can be converted back to the original function pointer type,
 1993 using an explicit cast, without loss of information.
 1994
 1995 Note: The ISO C standard does not require this, but it is required for
 1996 POSIX conformance.
 1997
 1998 *Rationale:* Austin Group Defect Report(s) applied: 74.
 1999 See <http://austingroupbugs.net/view.php?id=74>
 2000
 2001
 2002 **Change Number: XSH/TC1/D5/0018 [464]**
 2003
 2004
 2005 On Page: 559 Line: 19380 Section: accept()
 2006
 2007 In the DESCRIPTION section, change from:
 2008
 2009 address_len Points to a socklen_t structure ...
 2010
 2011 to:
 2012
 2013 address_len Either a null pointer, if address is a null pointer,
 2014 or a pointer to a socklen_t object ...
 2015
 2016 *Rationale:* Austin Group Defect Report(s) applied: 464.
 2017 See <http://austingroupbugs.net/view.php?id=464>
 2018
 2019
 2020 **Change Number: XSH/TC1/D5/0019 [461]**
 2021
 2022
 2023 On Page: 561 Line: 19464 Section: access()
 2024
 2025 In the DESCRIPTION section, change from:
 2026
 2027 ... the current working directory is used ...
 2028
 2029 to:
 2030
 2031 ... the current working directory shall be used ...

2032
2033 *Rationale:* Austin Group Defect Report(s) applied: 461.
2034 See <http://austingroupbugs.net/view.php?id=461>
2035
2036
2037 **Change Number: XSH/TC1/D5/0020** [324]
2038
2039
2040 On Page: 561 Line: 19482 Section: `access()`
2041
2042 In the ERRORS section, for the [ENOTDIR] error, change from:
2043
2044 A component of the path prefix is not a directory, ...
2045
2046 to:
2047
2048 A component of the path prefix names an existing file that is neither
2049 a directory nor a symbolic link to a directory, ...
2050
2051 *Rationale:* Austin Group Defect Report(s) applied: 324.
2052 See <http://austingroupbugs.net/view.php?id=324>
2053 This is an editorial issue clarifying the intent of the
2054 standard.
2055
2056
2057 **Change Number: XSH/TC1/D5/0021** [278]
2058
2059
2060 On Page: 562 Line: 19491 `access()`
2061
2062 In the ERRORS section, add (after the [EBADF] error):
2063
2064 [ENOTDIR] The path argument is not an absolute path and fd is a file
2065 descriptor associated with a non-directory file.
2066
2067 *Rationale:* Austin Group Defect Report(s) applied: 278.
2068 See <http://austingroupbugs.net/view.php?id=278>
2069
2070
2071 **Change Number: XSH/TC1/D5/0022** [278]
2072
2073
2074 On Page: 562 Line: 19504 Section: `access()`
2075
2076 In the ERRORS section, delete:
2077
2078 [ENOTDIR] The path argument is not an absolute path and fd is neither
2079 `AT_FDCWD` nor a file descriptor associated with a directory.
2080
2081 *Rationale:* Austin Group Defect Report(s) applied: 278.
2082 See <http://austingroupbugs.net/view.php?id=278>
2083
2084
2085 **Change Number: XSH/TC1/D5/0023** [291]
2086
2087
2088 On Page: 562 Line: 19512-19513 Section: `access()`
2089
2090 In the EXAMPLES section, change both instances of "filename" to
2091 "pathname".
2092
2093 *Rationale:* Austin Group Defect Report(s) applied: 291.
2094 See <http://austingroupbugs.net/view.php?id=291>

2095

2096

2097 **Change Number: XSH/TC1/D5/0024** [320]

2098

2099

2100 On Page: 564 Line: 19589 Section: acos()

2101

2102 Change:

2103

2104 ... and either a NaN (if supported), or an implementation-defined
2105 value shall be returned.

2106

2107 to:

2108

2109 ... and a NaN shall be returned.

2110

2111 *Rationale:* Austin Group Defect Report(s) applied: 320.2112 See <http://austingroupbugs.net/view.php?id=320>

2113

2114

2115 **Change Number: XSH/TC1/D5/0025** [320]

2116

2117

2118 On Page: 566 Line: 19645 Section: acosh()

2119

2120 In the RETURN VALUE section, change from:

2121

2122 ... and either a NaN (if supported), or an implementation-defined
2123 value shall be returned.

2124

2125 to:

2126

2127 ... and a NaN shall be returned.

2128

2129 *Rationale:* Austin Group Defect Report(s) applied: 320.2130 See <http://austingroupbugs.net/view.php?id=320>

2131

2132

2133 **Change Number: XSH/TC1/D5/0026** [98]

2134

2135

2136 On Page: 573 Line: 19785 Section: aio_fsync()

2137

2138 In the SYNOPSIS section, shade the synopsis and add the margin code
2139 FSC|SIO.

2140

2141 On L19788, in the DESCRIPTION section, change from:

2142

2143 The aio_fsync() function shall asynchronously force all I/O operations
2144 associated with the file indicated by the file descriptor aio_fildes
2145 member of the aiocb structure referenced by the aiocbp argument and
2146 queued at the time of the call to aio_fsync() to the synchronized I/O
2147 completion state.

2148

2149 to:

2150

2151 The aio_fsync() function shall asynchronously perform a file
2152 synchronization operation, as specified by the op argument, for I/O
2153 operations associated with the file indicated by the file descriptor
2154 aio_fildes member of the aiocb structure referenced by the aiocbp argument
2155 and queued at the time of the call to aio_fsync().

2156

2157 On L19793, in the DESCRIPTION section, add SIO shading to:

2158
2159 If `op` is `O_DSYNC`, all currently queued I/O operations shall be completed
2160 as if by a call to `fdatasync()`; that is, as defined for synchronized
2161 I/O data integrity completion.
2162
2163 On L19794, in the DESCRIPTION section, add FSC shading to:
2164
2165 If `op` is `O_SYNC`, all currently queued I/O operations shall be completed
2166 as if by a call to `fsync()`;
2167
2168 and add FSC SIO shading to:
2169
2170 ... that is, as defined for synchronized I/O file integrity completion.
2171
2172 On L19797, in the DESCRIPTION section, change from:
2173
2174 ... operation queued by `aio_fsync()` fails, then, as for `fsync()` and
2175 `fdatasync()`, outstanding I/O operations are not guaranteed to have
2176 been completed.
2177
2178 to:
2179
2180 ... operation queued by `aio_fsync()` fails, then outstanding I/O operations
2181 are not guaranteed to have been completed.
2182
2183 On L19825, in the ERRORS section, add SIO shading to:
2184
2185 [EINVAL] This implementation does not support synchronized I/O for
2186 this file.
2187
2188 and add another [EINVAL] error after it, shaded FSC:
2189
2190 [EINVAL] The `aio_fildes` member of the `aiocb` structure refers to a file
2191 on which an `fsync()` operation is not possible.
2192
2193 Rationale: Austin Group Defect Report(s) applied: 98.
2194 See <http://austingroupbugs.net/view.php?id=98>
2195
2196
2197 **Change Number: XSH/TC1/D5/0027 [98]**
2198
2199
2200 On Page: 574 Line: 19826 Section: `aio_fsync()`
2201
2202 In the ERRORS section, change from:
2203
2204 [EINVAL] A value of `op` other than `O_DSYNC` or `O_SYNC` was specified.
2205
2206 to:
2207
2208 [EINVAL] A value of `op` other than `O_DSYNC` or `O_SYNC` was specified, or
2209 `O_DSYNC` was specified and the implementation does not provide runtime
2210 support for the Synchronized Input and Output option, or `O_SYNC` was
2211 specified and the implementation does not provide runtime support for
2212 the File Synchronization option.
2213
2214 On L19829, in the ERRORS section, change from:
2215
2216 The error is returned in the error status for the asynchronous `fsync()`
2217 operation.
2218
2219 to:
2220

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

2221 The error is returned in the error status for the asynchronous operation.

2222

2223 *Rationale:* Austin Group Defect Report(s) applied: 98.

2224 See <http://austingroupbugs.net/view.php?id=98>

2225

2226

2227 **Change Number: XSH/TC1/D5/0028** [317]

2228

2229

2230 On Page: 582 Line: 20076 Section: aio_write()

2231

2232 Change:

2233

2234 If O_APPEND is set for the file descriptor, write operations append to
 2235 the file in the same order as the calls were made.

2236

2237 to:

2238

2239 If O_APPEND is set for the file descriptor, or if aio_fildes is associated
 2240 with a device that is incapable of seeking, write operations append
 2241 to the file in the same order as the calls were made, except under
 2242 circumstances described in Section 2.8.2.

2243

2244 *Rationale:* Austin Group Defect Report(s) applied: 317.

2245 See <http://austingroupbugs.net/view.php?id=317>

2246

2247

2248 **Change Number: XSH/TC1/D5/0029** [324]

2249

2250

2251 On Page: 588 Line: 20264 Section: alphasort()

2252

2253 In the ERRORS section, for the [ENOTDIR] error, change from:

2254

2255 A component of dir is not a directory.

2256

2257 to:

2258

2259 A component of dir names an existing file that is neither a directory
 2260 nor a symbolic link to a directory.

2261

2262 *Rationale:* Austin Group Defect Report(s) applied: 324.

2263 See <http://austingroupbugs.net/view.php?id=324>

2264 This is an editorial issue clarifying the intent of the
 2265 standard.

2266

2267

2268 **Change Number: XSH/TC1/D5/0030** [404]

2269

2270

2271 On Page: 588 Line: 20264 Section: alphasort()

2272

2273 In the ERRORS section, add to the end of the "shall fail" errors for
 2274 scandir():

2275

2276 [E_OVERFLOW] One of the values to be returned or passed to a callback
 2277 function cannot be represented correctly.

2278

2279 *Rationale:* Austin Group Defect Report(s) applied: 404.

2280 See <http://austingroupbugs.net/view.php?id=404>

2281

2282

2283 **Change Number: XSH/TC1/D5/0031** [393]

2284

2285

2286 On Page: 588 Line: 20268 Section: `alphasort()`

2287

2288 In the ERRORS section, [EMFILE] error, change from:

2289

2290 [EMFILE] {OPEN_MAX} file descriptors are currently open in the calling
2291 process.

2292

2293 to:

2294

2295 [EMFILE] All file descriptors available to the process are currently open.

2296

2297 Rationale: Austin Group Defect Report(s) applied: 393.

2298 See <http://austingroupbugs.net/view.php?id=393>

2299

2300

2301 **Change Number: XSH/TC1/D5/0032 [291]**

2302

2303

2304 On Page: 588 Line: 20294 Section: `alphasort()`

2305

2306 In the APPLICATION USAGE section, change from:

2307

2308 If `dir` contains filenames that contain characters outside the domain of
2309 the collating sequence of the current locale, the `alphasort()` function
2310 need not provide a total ordering.

2311

2312 to:

2313

2314 If `dir` contains filenames that do not form character strings, or
2315 which contain characters outside the domain of the collating sequence
2316 of the current locale, the `alphasort()` function need not provide a
2317 total ordering. This condition is not possible if all filenames within
2318 the directory consist only of characters from the portable filename
2319 character set.

2320

2321 Rationale: Austin Group Defect Report(s) applied: 291.

2322 See <http://austingroupbugs.net/view.php?id=291>

2323

2324

2325 **Change Number: XSH/TC1/D5/0033 [86,429]**

2326

2327

2328 On Page: 591 Line: 20382 Section: `asctime()`

2329

2330 In the RATIONALE section, change from:

2331

2332 The standard developers decided to mark the `asctime()` and `asctime_r()`
2333 functions obsolescent even though they are in ...

2334

2335 to:

2336

2337 The standard developers decided to mark the `asctime()` and `asctime_r()`
2338 functions obsolescent even though `asctime()` is in ...

2339

2340 Rationale: Austin Group Defect Report(s) applied: 86,429.

2341 See <http://austingroupbugs.net/view.php?id=86>2342 See <http://austingroupbugs.net/view.php?id=429>

2343

2344

2345 **Change Number: XSH/TC1/D5/0034 [320]**

2346

2347
 2348 On Page: 593 Line: 20436 Section: asin()
 2349
 2350 In the RETURN VALUE section, change from:
 2351
 2352 ... and either a NaN (if supported), or an implementation-defined
 2353 value shall be returned.
 2354
 2355 to:
 2356
 2357 ... and a NaN shall be returned.
 2358
 2359 *Rationale:* Austin Group Defect Report(s) applied: 320.
 2360 See <http://austingroupbugs.net/view.php?id=320>
 2361
 2362
 2363 **Change Number: XSH/TC1/D5/0035 [68]**
 2364
 2365
 2366 On Page: 593 Line: 20438 Section: asin()
 2367
 2368 In the RETURN VALUE section, change from:
 2369
 2370 [MX]If x is subnormal, a range error may occur and x should be
 2371 returned. [/MX]
 2372
 2373 to:
 2374
 2375 [MX]If x is subnormal, a range error may occur [/MX] [MXX]and x should
 2376 be returned. [/MXX]
 2377
 2378 [MX]If x is not returned, asin(), asinf(), and asinl() shall return
 2379 an implementation-defined value no greater in magnitude than DBL_MIN,
 2380 FLT_MIN, and LDBL_MIN, respectively. [/MX]
 2381
 2382 *Rationale:* Austin Group Defect Report(s) applied: 68.
 2383 See <http://austingroupbugs.net/view.php?id=68>
 2384
 2385
 2386 **Change Number: XSH/TC1/D5/0036 [68]**
 2387
 2388
 2389 On Page: 595 Line: 20496 Section: asinh()
 2390
 2391 In the RETURN VALUE section, change from:
 2392
 2393 [MX]If x is subnormal, a range error may occur and x should be
 2394 returned. [/MX]
 2395
 2396 to:
 2397
 2398 [MX]If x is subnormal, a range error may occur [/MX] [MXX]and x should
 2399 be returned. [/MXX]
 2400
 2401 [MX]If x is not returned, asinh(), asinhf(), and asinhl() shall return
 2402 an implementation-defined value no greater in magnitude than DBL_MIN,
 2403 FLT_MIN, and LDBL_MIN, respectively. [/MX]
 2404
 2405 *Rationale:* Austin Group Defect Report(s) applied: 68.
 2406 See <http://austingroupbugs.net/view.php?id=68>
 2407
 2408
 2409 **Change Number: XSH/TC1/D5/0037 [68]**

2410
2411
2412 On Page: 599 Line: 20598 Section: atan()
2413
2414 In the RETURN VALUE section, change from:
2415
2416 [MX]If x is subnormal, a range error may occur and x should be
2417 returned.[/MX]
2418
2419 to:
2420
2421 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
2422 be returned.[/MXX]
2423
2424 [MX]If x is not returned, atan(), atanf(), and atanl() shall return
2425 an implementation-defined value no greater in magnitude than DBL_MIN,
2426 FLT_MIN, and LDBL_MIN, respectively.[/MX]
2427
2428 *Rationale:* Austin Group Defect Report(s) applied: 68.
2429 See <http://austingroupbugs.net/view.php?id=68>
2430
2431
2432 **Change Number: XSH/TC1/D5/0038** [68,428]
2433
2434
2435 On Page: 601 Line: 20655 Section: atan2()
2436
2437 In the RETURN VALUE section, change from:
2438
2439 [MX]If the result underflows, a range error may occur and y/x should
2440 be returned.[/MX]
2441
2442 to:
2443
2444 If the correct value would cause underflow, a range error may occur, and
2445 atan2(), atan2f(), and atan2l() shall return an implementation-defined
2446 value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN,
2447 respectively. [MXX]If IEC 60559 Floating-Point is supported, y/x should
2448 be returned.[/MXX]
2449
2450 *Rationale:* Austin Group Defect Report(s) applied: 68,428.
2451 See <http://austingroupbugs.net/view.php?id=68>
2452 See <http://austingroupbugs.net/view.php?id=428>
2453
2454
2455 **Change Number: XSH/TC1/D5/0039** [320]
2456
2457
2458 On Page: 605 Line: 20745 Section: atanh()
2459
2460 In the RETURN VALUE section, change from:
2461
2462 ... and either a NaN (if supported), or an implementation-defined
2463 value shall be returned.
2464
2465 to:
2466
2467 ... and a NaN shall be returned.
2468
2469 *Rationale:* Austin Group Defect Report(s) applied: 320.
2470 See <http://austingroupbugs.net/view.php?id=320>
2471
2472

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

2473 **Change Number: XSH/TC1/D5/0040** [68]

2474

2475

2476 On Page: 605 Line: 20747 Section: atanh()

2477

2478 In the RETURN VALUE section, change from:

2479

2480 [MX]If x is subnormal, a range error may occur and x should be
 2481 returned.[/MX]

2482

2483 to:

2484

2485 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
 2486 be returned.[/MXX]

2487

2488 [MX]If x is not returned, atanh(), atanhf(), and atanhll() shall return
 2489 an implementation-defined value no greater in magnitude than DBL_MIN,
 2490 FLT_MIN, and LDBL_MIN, respectively.[/MX]

2491

2492 *Rationale:* Austin Group Defect Report(s) applied: 68.

2493

2494 See <http://austingroupbugs.net/view.php?id=68>

2495

2496

2497

2498

2499 On Page: 614 Line: 20979 Section: basename()

2500

2501 In the DESCRIPTION section, change from:

2502

2503 The basename() function may modify the string pointed to by path, and
 2504 may return a pointer to static storage that may then be overwritten by
 2505 a subsequent call to basename().

2506

2507 to:

2508

2509 The basename() function may modify the string pointed to by path, and
 2510 may return a pointer to internal storage. The returned pointer might
 2511 be invalidated or the storage might be overwritten by a subsequent call
 2512 to basename().

2513

2514 *Rationale:* Austin Group Defect Report(s) applied: 75.

2515

2516 See <http://austingroupbugs.net/view.php?id=75>

2517

2518

2519

2520

2521 On Page: 616 Line: 21046 Section: bind()

2522

2523 In the DESCRIPTION section, add a new paragraph after the existing
 2524 third paragraph:

2525

2526 If the address family of the socket is AF_UNIX and the pathname in address
 2527 names a symbolic link, bind() shall fail and set errno to [EADDRINUSE].

2528

2529 *Rationale:* Austin Group Defect Report(s) applied: 146.

2530

2531 See <http://austingroupbugs.net/view.php?id=146>

2532

2533

2534

2535

Change Number: XSH/TC1/D5/0043 [146]

2536 On Page: 617 Line: 21086 Section: bind()
2537
2538 In the ERRORS section, change from:
2539
2540 [ENOENT] A component of the pathname does not name an existing file
2541 or the pathname is an empty string.
2542
2543 to:
2544
2545 [ENOENT] A component of the path prefix of the pathname in address does
2546 not name an existing file or the pathname is an empty string.
2547
2548 [ENOENT] or [ENOTDIR] The pathname in address contains at least one
2549 non-`<slash>` character and ends with one or more trailing `<slash>`
2550 characters. If the pathname names an existing file, an [ENOENT] error
2551 shall not occur.
2552
2553 *Rationale:* Austin Group Defect Report(s) applied: 146.
2554 See <http://austingroupbugs.net/view.php?id=146>
2555
2556
2557 **Change Number: XSH/TC1/D5/0044 [324]**
2558
2559
2560 On Page: 617 Line: 21088 Section: bind()
2561
2562 In the ERRORS section, for the [ENOTDIR] error, change from:
2563
2564 A component of the path prefix of the pathname in address
2565 is not a directory, ...
2566
2567 to:
2568
2569 A component of the path prefix of the pathname in address names an
2570 existing file that is neither a directory nor a symbolic link to a
2571 directory, ...
2572
2573 *Rationale:* Austin Group Defect Report(s) applied: 324.
2574 See <http://austingroupbugs.net/view.php?id=324>
2575 This is an editorial issue clarifying the intent of the
2576 standard.
2577
2578
2579 **Change Number: XSH/TC1/D5/0045 [324]**
2580
2581
2582 On Page: 640 Line: 21719 Section: catopen()
2583
2584 In the ERRORS section, for the [ENOTDIR] error, change from:
2585
2586 A component of the path prefix of the message catalog is not a
2587 directory, ...
2588
2589 to:
2590
2591 A component of the path prefix of the message catalog names an existing
2592 file that is neither a directory nor a symbolic link to a directory, ...
2593
2594 *Rationale:* Austin Group Defect Report(s) applied: 324.
2595 See <http://austingroupbugs.net/view.php?id=324>
2596 This is an editorial issue clarifying the intent of the
2597 standard.
2598

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

2599
 2600 **Change Number: XSH/TC1/D5/0046** [346]
 2601
 2602
 2603 On Page: 645 Line: 21865 Section: ceil()
 2604
 2605 For ceil(), delete P645 L21865-21868 (DESCRIPTION), L21874-21875 (RETURN
 2606 VALUE), and replace L21877-21882 (ERRORS) with:
 2607
 2608 No errors are defined.
 2609
 2610 At L21870 (RETURN VALUE), add:
 2611
 2612 [MX]The result shall have the same sign as x.[/MX]
 2613
 2614 At L21886 (APPLICATION USAGE), change "an int or long" to "an intmax_t"
 2615 and replace L21889-21892 (APPLICATION USAGE) with:
 2616
 2617 These functions may raise the inexact floating-point exception if the
 2618 result differs in value from the argument.
 2619
 2620 *Rationale:* Austin Group Defect Report(s) applied: 346.
 2621 See <http://austingroupbugs.net/view.php?id=346>
 2622
 2623
 2624 **Change Number: XSH/TC1/D5/0047** [324]
 2625
 2626
 2627 On Page: 653 Line: 22116 Section: chdir()
 2628
 2629 In the ERRORS section, for the [ENOTDIR] error, change from:
 2630
 2631 A component of the pathname is not a directory; ...
 2632
 2633 to:
 2634
 2635 A component of the pathname names an existing file that is neither
 2636 a directory nor a symbolic link to a directory; ...
 2637
 2638 *Rationale:* Austin Group Defect Report(s) applied: 324.
 2639 See <http://austingroupbugs.net/view.php?id=324>
 2640 This is an editorial issue clarifying the intent of the
 2641 standard.
 2642
 2643
 2644 **Change Number: XSH/TC1/D5/0048** [300]
 2645
 2646
 2647 On Page: 655 Line: 22172-22173 Section: chmod()
 2648
 2649 Delete the paragraph at L22172-22173 from the DESCRIPTION section
 2650 of chmod().
 2651
 2652 *Rationale:* Austin Group Defect Report(s) applied: 300.
 2653 See <http://austingroupbugs.net/view.php?id=300>
 2654
 2655
 2656 **Change Number: XSH/TC1/D5/0049** [461]
 2657
 2658
 2659 On Page: 655 Line: 22186 Section: chmod()
 2660
 2661 In the DESCRIPTION section, change from:

2662
2663 ... the current working directory is used ...
2664
2665 to:
2666
2667 ... the current working directory shall be used ...
2668
2669 *Rationale:* Austin Group Defect Report(s) applied: 461.
2670 See <http://austingroupbugs.net/view.php?id=461>
2671
2672
2673 **Change Number: XSH/TC1/D5/0050 [324]**
2674
2675
2676 On Page: 656 Line: 22199 Section: chmod()
2677
2678 In the ERRORS section, for the [ENOTDIR] error, change from:
2679
2680 A component of the path prefix is not a directory, ...
2681
2682 to:
2683
2684 A component of the path prefix names an existing file that is neither
2685 a directory nor a symbolic link to a directory, ...
2686
2687 *Rationale:* Austin Group Defect Report(s) applied: 324.
2688 See <http://austingroupbugs.net/view.php?id=324>
2689 This is an editorial issue clarifying the intent of the standard.
2690
2691
2692 **Change Number: XSH/TC1/D5/0051 [278]**
2693
2694
2695 On Page: 656 Line: 22210 chmod()
2696
2697 In the ERRORS section, add (after the [EBADF] error):
2698
2699 [ENOTDIR] The path argument is not an absolute path and fd is a file
2700 descriptor associated with a non-directory file.
2701
2702 *Rationale:* Austin Group Defect Report(s) applied: 278.
2703 See <http://austingroupbugs.net/view.php?id=278>
2704
2705
2706 **Change Number: XSH/TC1/D5/0052 [278]**
2707
2708
2709 On Page: 656 Line: 22222 Section: chmod()
2710
2711 In the ERRORS section, delete:
2712
2713 [ENOTDIR] The path argument is not an absolute path and fd is neither
2714 AT_FDCWD nor a file descriptor associated with a directory.
2715
2716 *Rationale:* Austin Group Defect Report(s) applied: 278.
2717 See <http://austingroupbugs.net/view.php?id=278>
2718
2719
2720 **Change Number: XSH/TC1/D5/0053 [461]**
2721
2722
2723 On Page: 659 Line: 22353 Section: chown()
2724

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

2725 In the DESCRIPTION section, change from:
 2726
 2727 ... the current working directory is used ...
 2728
 2729 to:
 2730
 2731 ... the current working directory shall be used ...
 2732
 2733 *Rationale:* Austin Group Defect Report(s) applied: 461.
 2734 See <http://austingroupbugs.net/view.php?id=461>
 2735
 2736
 2737 **Change Number: XSH/TC1/D5/0054 [324]**
 2738
 2739
 2740 On Page: 660 Line: 22367 Section: `chown()`
 2741
 2742 In the ERRORS section, for the [ENOTDIR] error, change from:
 2743
 2744 A component of the path prefix is not a directory, ...
 2745
 2746 to:
 2747
 2748 A component of the path prefix names an existing file that is neither
 2749 a directory nor a symbolic link to a directory, ...
 2750
 2751 *Rationale:* Austin Group Defect Report(s) applied: 324.
 2752 See <http://austingroupbugs.net/view.php?id=324>
 2753 This is an editorial issue clarifying the intent of the standard.
 2754
 2755
 2756 **Change Number: XSH/TC1/D5/0055 [278]**
 2757
 2758
 2759 On Page: 660 Line: 22379 `chown()`
 2760
 2761 In the ERRORS section, add (after the [EBADF] error):
 2762
 2763 [ENOTDIR] The path argument is not an absolute path and fd is a file
 2764 descriptor associated with a non-directory file.
 2765
 2766 *Rationale:* Austin Group Defect Report(s) applied: 278.
 2767 See <http://austingroupbugs.net/view.php?id=278>
 2768
 2769
 2770 **Change Number: XSH/TC1/D5/0056 [278]**
 2771
 2772
 2773 On Page: 661 Line: 22393 Section: `chown()`
 2774
 2775 In the ERRORS section, delete:
 2776
 2777 [ENOTDIR] The path argument is not an absolute path and fd is neither
 2778 AT_FDCWD nor a file descriptor associated with a directory.
 2779
 2780 *Rationale:* Austin Group Defect Report(s) applied: 278.
 2781 See <http://austingroupbugs.net/view.php?id=278>
 2782
 2783
 2784 **Change Number: XSH/TC1/D5/0057 [401]**
 2785
 2786
 2787 On Page: 664 Line: 22502 Section: `clearerr()`

2788
2789 In the DESCRIPTION section, add to the end of the section:
2790
2791 [CX]The clearerr() function shall not change the setting of errno if
2792 stream is valid.[/CX]
2793
2794 *Rationale:* Austin Group Defect Report(s) applied: 401.
2795 See <http://austingroupbugs.net/view.php?id=401>
2796
2797
2798 **Change Number: XSH/TC1/D5/0058** [106]
2799
2800
2801 On Page: 668 Line: 22669 Section: clock_gettime()
2802
2803 In the ERRORS section, add after the existing [EINVAL] error:
2804
2805 The clock_gettime() function shall fail if:
2806
2807 [Eoverflow] The number of seconds will not fit in an object of type
2808 time_t.
2809
2810 *Rationale:* Austin Group Defect Report(s) applied: 106.
2811 See <http://austingroupbugs.net/view.php?id=106>
2812
2813
2814 **Change Number: XSH/TC1/D5/0059** [419]
2815
2816
2817 On Page: 677 Line: 22921 Section: close()
2818
2819 In the ERRORS section, change from:
2820
2821 [EBADF] The fildes argument is not a valid file descriptor.
2822
2823 to:
2824
2825 [EBADF] The fildes argument is not an open file descriptor.
2826
2827 *Rationale:* Austin Group Defect Report(s) applied: 419.
2828 See <http://austingroupbugs.net/view.php?id=419>
2829
2830
2831 **Change Number: XSH/TC1/D5/0060** [149]
2832
2833
2834 On Page: 678 Line: 22962 Section: close()
2835
2836 In the APPLICATION USAGE section, add a new paragraph at the end of
2837 the section:
2838
2839 Implementations may use file descriptors that must be inherited into child
2840 processes for the child process to remain conforming, such as for message
2841 catalog or tracing purposes. Therefore, an application that calls close()
2842 on an arbitrary integer risks non-conforming behavior, and close() can
2843 only portably be used on file descriptor values that the application has
2844 obtained through explicit actions, as well as the three file descriptors
2845 corresponding to the standard file streams. In multi-threaded parent
2846 applications, the practice of calling close() in a loop after fork()
2847 and before an exec call in order to avoid a race condition of leaking
2848 an unintended file descriptor into a child process, is therefore unsafe,
2849 and the race should instead be combatted by opening all file descriptors
2850 with the FD_CLOEXEC bit set unless the file descriptor is intended to

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

2851 be inherited across exec.
 2852
 2853 *Rationale:* Austin Group Defect Report(s) applied: 149.
 2854 See <http://austingroupbugs.net/view.php?id=149>
 2855
 2856
 2857 **Change Number: XSH/TC1/D5/0061 [149]**
 2858
 2859
 2860 On Page: 678 Line: 22968 Section: close()
 2861
 2862 In the RATIONALE section, add a new paragraph at the end of the section:
 2863
 2864 The standard developers rejected a proposal to add closefrom() to the
 2865 standard. Because the standard permits implementations to use inherited
 2866 file descriptors as a means of providing a conforming environment for
 2867 the child process, it is not possible to standardize an interface that
 2868 closes arbitrary file descriptors above a certain value while still
 2869 guaranteeing a conforming environment.
 2870
 2871 *Rationale:* Austin Group Defect Report(s) applied: 149.
 2872 See <http://austingroupbugs.net/view.php?id=149>
 2873
 2874
 2875 **Change Number: XSH/TC1/D5/0062 [324]**
 2876
 2877
 2878 On Page: 691 Line: 23411 Section: connect()
 2879
 2880 In the ERRORS section, for the [ENOTDIR] error, change from:
 2881
 2882 A component of the path prefix of the pathname in address is not a
 2883 directory, ...
 2884
 2885 to:
 2886
 2887 A component of the path prefix of the pathname in address names an
 2888 existing file that is neither a directory nor a symbolic link to a
 2889 directory, ...
 2890
 2891 *Rationale:* Austin Group Defect Report(s) applied: 324.
 2892 See <http://austingroupbugs.net/view.php?id=324>
 2893 This is an editorial issue clarifying the intent of the standard.
 2894
 2895
 2896 **Change Number: XSH/TC1/D5/0063 [320]**
 2897
 2898
 2899 On Page: 694 Line: 23510 Section: cos()
 2900
 2901 In the RETURN VALUE section, change from:
 2902
 2903 ... and either a NaN (if supported), or an implementation-defined
 2904 value shall be returned.
 2905
 2906 to:
 2907
 2908 ... and a NaN shall be returned.
 2909
 2910 *Rationale:* Austin Group Defect Report(s) applied: 320.
 2911 See <http://austingroupbugs.net/view.php?id=320>
 2912
 2913

2914 **Change Number: XSH/TC1/D5/0064** [291]
2915
2916
2917 On Page: 702 Line: 23739-23741 Section: creat()
2918
2919 In the EXAMPLES section, change both instances of "filename" to
2920 "pathname".
2921
2922 *Rationale:* Austin Group Defect Report(s) applied: 291.
2923 See <http://austingroupbugs.net/view.php?id=291>
2924
2925
2926 **Change Number: XSH/TC1/D5/0065** [75,428]
2927
2928
2929 On Page: 713 Line: 24015 Section: ctermid()
2930
2931 In the RETURN VALUE section, change from:
2932
2933 If *s* is a null pointer, the string shall be generated in an area that may
2934 be static (and therefore may be overwritten by each call), the address
2935 of which shall be returned. Otherwise, ...
2936
2937 to:
2938
2939 If *s* is a null pointer, the string shall be generated in a area that
2940 may be static, the address of which shall be returned. The application
2941 shall not modify the string returned. The returned pointer might be
2942 invalidated or the string content might be overwritten by a subsequent
2943 call to ctermid(). If *s* is not a null pointer, ...
2944
2945 *Rationale:* Austin Group Defect Report(s) applied: 75,428.
2946 See <http://austingroupbugs.net/view.php?id=75>
2947 See <http://austingroupbugs.net/view.php?id=428>
2948
2949
2950 **Change Number: XSH/TC1/D5/0066** [321,428]
2951
2952
2953 On Page: 715 Line: 24085 Section: ctime()
2954
2955 In the DESCRIPTION section, change from:
2956
2957 Unlike ctime(), the thread-safe version of ctime_r() is not required to
2958 set tzname.
2959
2960 to:
2961
2962 Unlike ctime(), the ctime_r() function is not required to set tzname. If
2963 ctime_r() does not set tzname, it shall not set daylight and shall not set
2964 timezone.
2965
2966 *Rationale:* Austin Group Defect Report(s) applied: 321,428.
2967 See <http://austingroupbugs.net/view.php?id=321>
2968 See <http://austingroupbugs.net/view.php?id=428>
2969
2970
2971 **Change Number: XSH/TC1/D5/0067** [422]
2972
2973
2974 On Page: 723 Line: 24328 Section: dirfd()
2975
2976 In the DESCRIPTION section, change from:

2977

2978 If any attempt is made to close the file descriptor, or to modify the
 2979 state of the associated description, other than by means of `closedir()`,
 2980 `readdir()`, `readdir_r()`, or `rewinddir()`, the behavior is undefined.

2981

2982 to:

2983

2984 If any attempt is made to close the file descriptor, or to modify the
 2985 state of the associated description, other than by means of `closedir()`,
 2986 `readdir()`, `readdir_r()`, `rewinddir()`, or `[XSI]seekdir()[/XSI]`,
 2987 the behavior is undefined.

2988

2989 *Rationale:* Austin Group Defect Report(s) applied: 422.2990 See <http://austingroupbugs.net/view.php?id=422>

2991

2992

2993 **Change Number: XSH/TC1/D5/0068 [75]**

2994

2995

2996 On Page: 725 Line: 24380 Section: `dirname()`

2997

2998 In the RETURN VALUE section, change from:

2999

3000 The `dirname()` function may modify the string pointed to by `path`, and
 3001 may return a pointer to static storage that may then be overwritten by
 3002 subsequent calls to `dirname()`.

3003

3004 to:

3005

3006 The `dirname()` function may modify the string pointed to by `path`, and
 3007 may return a pointer to internal storage. The returned pointer might
 3008 be invalidated or the storage might be overwritten by a subsequent call
 3009 to `dirname()`.

3010

3011 *Rationale:* Austin Group Defect Report(s) applied: 75.3012 See <http://austingroupbugs.net/view.php?id=75>

3013

3014

3015 **Change Number: XSH/TC1/D5/0069 [74]**

3016

3017

3018 On Page: 728 Line: 24469 Section: `dlclose()`

3019

3020 Replace the NAME, SYNOPSIS, DESCRIPTION, RETURN VALUE, ERRORS,
 3021 EXAMPLES, and APPLICATION USAGE sections of the `dlclose()` description
 3022 on L24469-24513 with:

3023

3024 NAME

3025

3026 `dlclose` -- close a symbol table handle

3027

3028 SYNOPSIS

3029

3030 `#include <dlfcn.h>`3031 `int dlclos(void *handle);`

3032

3033 DESCRIPTION

3034

3035 The `dlclose()` function shall inform the system that the symbol table
 3036 handle specified by `handle` is no longer needed by the application.

3037

3038 An application writer may use `dlclose()` to make a statement of intent
 3039 on the part of the process, but this statement does not create any

3040 requirement upon the implementation. When the symbol table handle is
3041 closed, the implementation may unload the executable object files that
3042 were loaded by `dlopen()` when the symbol table handle was opened and
3043 those that were loaded by `dlsym()` when using the symbol table handle
3044 identified by handle.

3045
3046 Once a symbol table handle has been closed, an application should assume
3047 that any symbols (function identifiers and data object identifiers)
3048 made visible using handle, are no longer available to the process.
3049

3050 Although a `dlclose()` operation is not required to remove any functions
3051 or data objects from the address space, neither is an implementation
3052 prohibited from doing so. The only restriction on such a removal is that
3053 no function nor data object shall be removed to which references have been
3054 relocated, until or unless all such references are removed. For instance,
3055 an executable object file that had been loaded with a `dlopen()` operation
3056 specifying the `RTLD_GLOBAL` flag might provide a target for dynamic
3057 relocations performed in the processing of other relocatable objects.
3058 In such environments, an application may assume that no relocation,
3059 once made, shall be undone or remade unless the executable object file
3060 containing the relocated object has itself been removed.

3061
3062 RETURN VALUE

3063
3064 If the referenced symbol table handle was successfully closed, `dlclose()`
3065 shall return 0. If handle does not refer to an open symbol table handle
3066 or if the symbol table handle could not be closed, `dlclose()` shall return
3067 a non-zero value. More detailed diagnostic information shall be available
3068 through `dlerror()`.

3069
3070 ERRORS

3071
3072 No errors are defined.

3073
3074 EXAMPLES

3075
3076 The following example illustrates use of `dlopen()` and `dlclose()`:

```
3077  
3078 #include <dlfcn.h>  
3079 int eret;  
3080 void *mylib;  
3081 ...  
3082 /* Open a dynamic library and then close it ... */  
3083 mylib = dlopen("mylib.so", RTLD_LOCAL | RTLD_LAZY);  
3084 ...  
3085 eret = dlclose(mylib);  
3086 ...
```

3087
3088 APPLICATION USAGE

3089
3090 A conforming application should employ a symbol table handle returned from
3091 a `dlopen()` invocation only within a given scope bracketed by a `dlopen()`
3092 operation and the corresponding `dlclose()` operation. Implementations are
3093 free to use reference counting or other techniques such that multiple
3094 calls to `dlopen()` referencing the same executable object file may return
3095 a pointer to the same data object as the symbol table handle.

3096
3097 Implementations are also free to re-use a handle. For these reasons,
3098 the value of a handle must be treated as an opaque data type by the
3099 application, used only in calls to `dlsym()` and `dlclose()`.

3100

3101 *Rationale:* Austin Group Defect Report(s) applied: 74.

3102 See <http://austingroupbugs.net/view.php?id=74>

3103

3104

3105 **Change Number: XSH/TC1/D5/0070 [75]**

3106

3107

3108 On Page: 730 Line: 24542 Section: dlerror()

3109

3110 In the RETURN VALUE section, add a new paragraph at the end of the
3111 section:

3112

3113 The application shall not modify the string returned. The returned
 3114 pointer might be invalidated or the string content might be overwritten
 3115 by a subsequent call to dlerror() in the same thread (if dlerror()
 3116 is thread-safe) or in any thread (if dlerror() is not thread-safe).

3117

3118 On L24555, in the APPLICATION USAGE section, delete the following:

3119

3120 The messages returned by dlerror() may reside in a static buffer that
 3121 is overwritten on each call to dlerror(). Application code should not
 3122 write to this buffer. Programs wishing to preserve an error message
 3123 should make their own copies of that message.

3124

3125 *Rationale:* Austin Group Defect Report(s) applied: 75.3126 See <http://austingroupbugs.net/view.php?id=75>

3127

3128

3129 **Change Number: XSH/TC1/D5/0071 [97]**

3130

3131

3132 On Page: 730 Line: 24539 Section: dlerror()

3133

3134 In the DESCRIPTION section, change from:

3135

3136 The dlerror() function need not be thread-safe.

3137

3138 to:

3139

3140 It is implementation-defined whether or not the dlerror() function is
 3141 thread-safe. A thread-safe implementation shall return only errors that
 3142 occur on the current thread.

3143

3144 *Rationale:* Austin Group Defect Report(s) applied: 97.3145 See <http://austingroupbugs.net/view.php?id=97>

3146

3147

3148 **Change Number: XSH/TC1/D5/0072 [133]**

3149

3150

3151 On Page: 730 Line: 24552 Section: dlerror()

3152

3153 In the EXAMPLES section, indent the printf() statement one tab stop.

3154

3155 *Rationale:* Austin Group Defect Report(s) applied: 133.3156 See <http://austingroupbugs.net/view.php?id=133>

3157 The indentation in the EXAMPLE is improved.

3158

3159

3160 **Change Number: XSH/TC1/D5/0073 [74]**

3161

3162

3163 On Page: 732 Line: 24575 Section: dlopen()

3164

3165 Replace the NAME, SYNOPSIS, DESCRIPTION, and RETURN VALUE sections of

3166 the dlopen() description on P732-733 L24575-24666:
3167
3168 NAME
3169
3170 dlopen -- open a symbol table handle
3171
3172 SYNOPSIS
3173
3174 #include <dlfcn.h>
3175 void *dlopen(const char *file, int mode);
3176
3177 DESCRIPTION
3178
3179 The dlopen() function shall make the symbols (function identifiers and
3180 data object identifiers) in the executable object file specified by file
3181 available to the calling program.
3182
3183 The class of executable object files eligible for this operation and
3184 the manner of their construction are implementation-defined, though
3185 typically such files are shared libraries or programs.
3186
3187 Implementations may permit the construction of embedded dependencies in
3188 executable object files. In such cases, a dlopen() operation shall load
3189 those dependencies in addition to the executable object file specified
3190 by file. Implementations may also impose specific constraints on the
3191 construction of programs that can employ dlopen() and its related
3192 services.
3193
3194 A successful dlopen() shall return a symbol table handle which the caller
3195 may use on subsequent calls to dlsym() and dlclose().
3196
3197 The value of this symbol table handle should not be interpreted in any
3198 way by the caller.
3199
3200 The file argument is used to construct a pathname to the executable
3201 object file. If file contains a <slash> character, the file argument
3202 is used as the pathname for the file. Otherwise, file is used in an
3203 implementation-defined manner to yield a pathname.
3204
3205 If file is a null pointer, dlopen() shall return a global symbol table
3206 handle for the currently running process image. This symbol table handle
3207 shall provide access to the symbols from an ordered set of executable
3208 object files consisting of the original program image file, any executable
3209 object files loaded at program start-up as specified by that process image
3210 file (for example, shared libraries), and the set of executable object
3211 files loaded using dlopen() operations with the RTLD_GLOBAL flag. As
3212 the latter set of executable object files can change during execution,
3213 the set of symbols made available by this symbol table handle can also
3214 change dynamically.
3215
3216 Only a single copy of an executable object file shall be brought into the
3217 address space, even if dlopen() is invoked multiple times in reference
3218 to the executable object file, and even if different pathnames are used
3219 to reference the executable object file.
3220
3221 The mode parameter describes how dlopen() shall operate upon file with
3222 respect to the processing of relocations and the scope of visibility
3223 of the symbols provided within file. When an executable object file is
3224 brought into the address space of a process, it may contain references
3225 to symbols whose addresses are not known until the executable object
3226 file is loaded.
3227
3228 These references shall be relocated before the symbols can be

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

3229 accessed. The mode parameter governs when these relocations take place
3230 and may have the following values:
3231
3232 RTLD_LAZY Relocations shall be performed at an implementation-defined
3233 time, ranging from the time of the dlopen() call until the first reference
3234 to a given symbol occurs. Specifying RTLD_LAZY should improve performance
3235 on implementations supporting dynamic symbol binding since a process
3236 might not reference all of the symbols in an executable object file. And,
3237 for systems supporting dynamic symbol resolution for normal process
3238 execution, this behavior mimics the normal handling of process execution.
3239
3240 RTLD_NOW All necessary relocations shall be performed when the executable
3241 object file is first loaded. This may waste some processing if relocations
3242 are performed for symbols that are never referenced. This behavior may
3243 be useful for applications that need to know that all symbols referenced
3244 during execution will be available before dlopen() returns.
3245
3246 Any executable object file loaded by dlopen() that requires relocations
3247 against global symbols can reference the symbols in the original process
3248 image file, any executable object files loaded at program start-up,
3249 from the initial process image itself, from any other executable object
3250 file included in the same dlopen() invocation, and any executable object
3251 files that were loaded in any dlopen() invocation and which specified
3252 the RTLD_GLOBAL flag. To determine the scope of visibility for the
3253 symbols loaded with a dlopen() invocation, the mode parameter should be
3254 a bitwise-inclusive OR with one of the following values:
3255
3256 RTLD_GLOBAL The executable object file's symbols shall be made available
3257 for relocation processing of any other executable object file. In
3258 addition, symbol lookup using dlopen(NULL, mode) and an associated dlsym()
3259 allows executable object files loaded with this mode to be searched.
3260
3261 RTLD_LOCAL The executable object file's symbols shall not be made
3262 available for relocation processing of any other executable object file.
3263
3264 If neither RTLD_GLOBAL nor RTLD_LOCAL are specified, the default behavior
3265 is unspecified.
3266
3267 If an executable object file is specified in multiple dlopen()
3268 invocations, mode is interpreted at each invocation.
3269
3270 If RTLD_NOW has been specified, all relocations shall have been completed
3271 rendering further RTLD_NOW operations redundant and any further RTLD_LAZY
3272 operations irrelevant.
3273
3274 If RTLD_GLOBAL has been specified, the executable object file shall
3275 maintain the RTLD_GLOBAL status regardless of any previous or future
3276 specification of RTLD_LOCAL, as long as the executable object file
3277 remains in the address space (see dlclose()).
3278
3279 Symbols introduced into the process image through calls to dlopen() may
3280 be used in relocation activities. Symbols so introduced may duplicate
3281 symbols already defined by the program or previous dlopen() operations. To
3282 resolve the ambiguities such a situation might present, the resolution of
3283 a symbol reference to symbol definition is based on a symbol resolution
3284 order. Two such resolution orders are defined: load order and dependency
3285 order. Load order establishes an ordering among symbol definitions,
3286 such that the first definition loaded (including definitions from the
3287 process image file and any dependent executable object files loaded
3288 with it) has priority over executable object files added later (by
3289 dlopen()). Load ordering is used in relocation processing. Dependency
3290 ordering uses a breadth-first order starting with a given executable
3291 object file, then all of its dependencies, then any dependents of those,

3292 iterating until all dependencies are satisfied. With the exception of
3293 the global symbol table handle obtained via a `dlopen()` operation with
3294 a null pointer as the file argument, dependency ordering is used by the
3295 `dlsym()` function. Load ordering is used in `dlsym()` operations upon the
3296 global symbol table handle.

3297
3298 When an executable object file is first made accessible via `dlopen()`,
3299 it and its dependent executable object files are added in dependency
3300 order. Once all the executable object files are added, relocations are
3301 performed using load order. Note that if an executable object file or
3302 its dependencies had been previously loaded, the load and dependency
3303 orders may yield different resolutions.

3304
3305 The symbols introduced by `dlopen()` operations and available through
3306 `dlsym()` are at a minimum those which are exported as identifiers of
3307 global scope by the executable object file. Typically such identifiers
3308 shall be those that were specified in (for example) C source code as
3309 having extern linkage. The precise manner in which an implementation
3310 constructs the set of exported symbols for an executable object file
3311 is implementation-defined.

3312
3313 RETURN VALUE

3314
3315 Upon successful completion, `dlopen()` shall return a symbol table
3316 handle. If file cannot be found, cannot be opened for reading, is not of
3317 an appropriate executable object file format for processing by `dlopen()`,
3318 or if an error occurs during the process of loading file or relocating
3319 its symbolic references, `dlopen()` shall return a null pointer. More
3320 detailed diagnostic information shall be available through `dlerror()`.

3321
3322 *Rationale:* Austin Group Defect Report(s) applied: 74.
3323 See <http://austingroupbugs.net/view.php?id=74>

3324
3325
3326 **Change Number: XSH/TC1/D5/0074 [74]**

3327
3328
3329 On Page: 735 Line: 24689 Section: `dlsym()`

3330
3331 Replace the NAME, SYNOPSIS, DESCRIPTION, RETURN VALUE, ERRORS, EXAMPLES,
3332 APPLICATION USAGE, and RATIONALE sections of the `dlsym()` description on
3333 P735-736 L24689-24748 with:

3334
3335 NAME
3336
3337 `dlsym` -- get the address of a symbol from a symbol table handle

3338
3339 SYNOPSIS
3340
3341 `#include <dlfcn.h>`
3342 `void *dlsym(void *restrict handle, const char *restrict name);`

3343
3344 DESCRIPTION
3345
3346 The `dlsym()` function shall obtain the address of a symbol (a function
3347 identifier or a data object identifier) defined in the symbol table
3348 identified by the handle argument. The handle argument is a symbol table
3349 handle returned from a call to `dlopen()` (and which has not since been
3350 released by a call to `dlclose()`), and name is the symbol's name as a
3351 character string. The return value from `dlsym()`, cast to a pointer to the
3352 type of the named symbol, can be used to call (in the case of a function)
3353 or access the contents of (in the case of a data object) the named symbol.

3354

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

3355 The `dlsym()` function shall search for the named symbol in the symbol table
 3356 referenced by `handle`. If the symbol table was created with lazy loading
 3357 (see `RTLD_LAZY` in `dlopen()`), load ordering shall be used in `dlsym()`
 3358 operations to relocate executable object files needed to resolve the
 3359 symbol. The symbol resolution algorithm used shall be dependency order
 3360 as described in `dlopen()`.

3361
 3362 The `RTLD_DEFAULT` and `RTLD_NEXT` symbolic constants (which may be defined in
 3363 `<dlfcn.h>`) are reserved for future use as special values that applications
 3364 may be allowed to use for `handle`.

3365
 3366 RETURN VALUE

3367
 3368 Upon successful completion, if `name` names a function identifier, `dlsym()`
 3369 shall return the address of the function converted from type pointer to
 3370 function to type pointer to void; otherwise, `dlsym()` shall return the
 3371 address of the data object associated with the data object identifier
 3372 named by `name` converted from a pointer to the type of the data object
 3373 to a pointer to void. If `handle` does not refer to a valid symbol table
 3374 `handle` or if the symbol named by `name` cannot be found in the symbol
 3375 table associated with `handle`, `dlsym()` shall return a null pointer.

3376
 3377 More detailed diagnostic information shall be available through `dlerror()`.

3378
 3379 ERRORS

3380
 3381 No errors are defined.

3382
 3383 EXAMPLES

3384
 3385 The following example shows how `dlopen()` and `dlsym()` can be used to access
 3386 either a function or a data object. For simplicity, error checking has
 3387 been omitted.

```
3388
3389 void *handle;
3390 int (*fptr)(int), *iptr, result;
3391 /* open the needed symbol table */
3392 handle = dlopen("/usr/home/me/libfoo.so", RTLD_LOCAL | RTLD_LAZY);
3393 /* find the address of the function my_function */
3394 fptr = (int (*)(int))dlsym(handle, "my_function");
3395 /* find the address of the data object my_object */
3396 iptr = (int *)dlsym(handle, "my_OBJ");
3397 /* invoke my_function, passing the value of my_OBJ as the parameter */
3398 result = (*fptr)(*iptr);
3399
```

3400 APPLICATION USAGE

3401
 3402 The following special purpose values for `handle` are reserved for future
 3403 use and have the indicated meanings:

3404
 3405 `RTLD_DEFAULT` The identifier lookup happens in the normal global scope;
 3406 that is, a search for a identifier using `handle` would find the same
 3407 definition as a direct use of this identifier in the program code.

3408
 3409 `RTLD_NEXT` Specifies the next executable object file after this one that
 3410 defines `name`. This one refers to the executable object file containing
 3411 the invocation of `dlsym()`. The next executable object file is the one
 3412 found upon the application of a load order symbol resolution algorithm
 3413 (see `dlopen()`). The next symbol is either one of global scope (because
 3414 it was introduced as part of the original process image or because it
 3415 was added with a `dlopen()` operation including the `RTLD_GLOBAL` flag), or
 3416 is in an executable object file that was included in the same `dlopen()`
 3417 operation that loaded this one.

3418

3419 The RTLD_NEXT flag is useful to navigate an intentionally created
3420 hierarchy of multiply-defined symbols created through interposition. For
3421 example, if a program wished to create an implementation of malloc()
3422 that embedded some statistics gathering about memory allocations, such
3423 an implementation could use the real malloc() definition to perform
3424 the memory allocation - and itself only embed the necessary logic to
3425 implement the statistics gathering function.

3426

3427 Note that conversion from a void * pointer to a function pointer as in:

3428

```
3429 fptr = (int (*)(int))dlsym(handle, "my_function");
```

3430

3431 is not defined by the ISO C Standard. This standard requires this
3432 conversion to work correctly on conforming implementations.

3433

3434 RATIONALE

3435

3436 None.

3437

3438 *Rationale:* Austin Group Defect Report(s) applied: 74.

3439 See <http://austingroupbugs.net/view.php?id=74>

3440

3441

3442 **Change Number: XSH/TC1/D5/0075 [149,428]**

3443

3444

3445 On Page: 742 Line: 24917 Section: dup()

3446

3447 In the APPLICATION USAGE section, change from:

3448

3449 None.

3450

3451 to:

3452

3453 Implementations may use file descriptors that must be inherited into child
3454 processes for the child process to remain conforming, such as for message
3455 catalog or tracing purposes. Therefore, an application that calls dup2()
3456 with an arbitrary integer for fildes2 risks non-conforming behavior,
3457 and dup2() can only portably be used to overwrite file descriptor values
3458 that the application has obtained through explicit actions, or for the
3459 three file descriptors corresponding to the standard file streams. In
3460 order to avoid a race condition of leaking an unintended file descriptor
3461 into a child process, an application should consider opening all file
3462 descriptors with the FD_CLOEXEC bit set unless the file descriptor is
3463 intended to be inherited across exec.

3464

3465 *Rationale:* Austin Group Defect Report(s) applied: 149,428.

3466 See <http://austingroupbugs.net/view.php?id=149>

3467 See <http://austingroupbugs.net/view.php?id=428>

3468

3469

3470 **Change Number: XSH/TC1/D5/0076 [149]**

3471

3472

3473 On Page: 742 Line: 24915 Section: dup()

3474

3475 In the RATIONALE section, change from:

3476

3477 The dup() and dup2() functions are redundant. Their services are
3478 also provided by the fcntl() function. They have been included in
3479 this volume of POSIX.1-2008 primarily for historical reasons, since
3480 many existing applications use them.

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

3481

3482 to:

3483

3484 The dup() function is redundant. Its services are also provided by
 3485 the fcntl() function. It has been included in this volume of
 3486 POSIX.1-2008 primarily for historical reasons, since many existing
 3487 applications use it. On the other hand, the dup2() function
 3488 provides unique services, as no other interface is able to atomically
 3489 replace an existing file descriptor.

3490

3491 *Rationale:* Austin Group Defect Report(s) applied: 149.3492 See <http://austingroupbugs.net/view.php?id=149>

3493

3494

3495 **Change Number: XSH/TC1/D5/0077** [283,301]

3496

3497

3498 On Page: 743 Line: 24943 Section: duplocale()

3499

3500 In the DESCRIPTION section, add two new paragraphs to the end of the
 3501 section:

3502

3503 If the locobj argument is LC_GLOBAL_LOCALE, duplocale() shall create a
 3504 new locale object containing a copy of the global locale determined by
 3505 the setlocale() function.

3506

3507 The behavior is undefined if the locobj argument is not a valid locale
 3508 object handle.

3509

3510 *Rationale:* Austin Group Defect Report(s) applied: 283,301.3511 See <http://austingroupbugs.net/view.php?id=283>3512 See <http://austingroupbugs.net/view.php?id=301>

3513

3514

3515 **Change Number: XSH/TC1/D5/0078** [283,301]

3516

3517

3518 On Page: 743 Line: 24951 Section: duplocale()

3519

3520 In the ERRORS section, delete:

3521

3522 The duplocale() function may fail if:

3523

3524 [EINVAL] locobj is not a handle for a locale object.

3525

3526 *Rationale:* Austin Group Defect Report(s) applied: 283.3527 See <http://austingroupbugs.net/view.php?id=283>

3528

3529

3530 **Change Number: XSH/TC1/D5/0079** [301]

3531

3532

3533 On Page: 743 Line: 24978 Section: duplocale()

3534

3535 In the APPLICATION USAGE section, add a new paragraph to the end of
 3536 the section:

3537

3538 The duplocale() function can also be used in conjunction with
 3539 uselocale((locale_t)0). This returns the locale in effect for the
 3540 calling thread, but can have the value LC_GLOBAL_LOCALE. Passing
 3541 LC_GLOBAL_LOCALE to functions such as isalnum_l() results in undefined
 3542 behavior, but applications can convert it into a usable locale object
 3543 by using duplocale().

3544

3545 *Rationale:* Austin Group Defect Report(s) applied: 301.3546 See <http://austingroupbugs.net/view.php?id=301>

3547

3548

3549 **Change Number: XSH/TC1/D5/0080 [75]**

3550

3551

3552 On Page: 747 Line: 25061 Section: `endgrent()`

3553

3554 In the RETURN VALUE section, change from:

3555

3556 The return value may point to a static area which is overwritten
3557 by a subsequent call to `getgrgid()`, `getgrnam()`, or `getgrent()`.

3558

3559 to:

3560

3561 The application shall not modify the structure to which the return
3562 value points, nor any storage areas pointed to by pointers within the
3563 structure. The returned pointer, and pointers within the structure,
3564 might be invalidated or the structure or the storage areas might be
3565 overwritten by a subsequent call to `getgrgid()`, `getgrnam()`, or `getgrent()`.

3566

3567 *Rationale:* Austin Group Defect Report(s) applied: 75.3568 See <http://austingroupbugs.net/view.php?id=75>

3569

3570

3571 **Change Number: XSH/TC1/D5/0081 [75,428]**

3572

3573

3574 On Page: 749 Line: 25121 Section: `endhostent()`

3575

3576 In the RETURN VALUE section, add a new paragraph to the end of the
3577 section:

3578

3579 The application shall not modify the structure to which the return
3580 value points, nor any storage areas pointed to by pointers within the
3581 structure. The returned pointer, and pointers within the structure,
3582 might be invalidated or the structure or the storage areas might be
3583 overwritten by a subsequent call to `gethostent()`.

3584

3585 *Rationale:* Austin Group Defect Report(s) applied: 75,428.3586 See <http://austingroupbugs.net/view.php?id=75>3587 See <http://austingroupbugs.net/view.php?id=428>

3588

3589

3590 **Change Number: XSH/TC1/D5/0082 [75]**

3591

3592

3593 On Page: 749 Line: 25127 Section: `endhostent()`

3594

3595 In the APPLICATION USAGE section, change from:

3596

3597 The `gethostent()` function may return pointers to static data, which may
3598 be overwritten by subsequent calls to any of these functions.

3599

3600 to:

3601

3602 None.

3603

3604 *Rationale:* Austin Group Defect Report(s) applied: 75.3605 See <http://austingroupbugs.net/view.php?id=75>

3606

3607

3608 **Change Number: XSH/TC1/D5/0083 [75]**

3609

3610

3611 On Page: 751 Line: 25175 Section: endnetent()

3612

3613 In the RETURN VALUE section, add a new paragraph to the end of the
3614 section:

3615

3616 The application shall not modify the structure to which the return
 3617 value points, nor any storage areas pointed to by pointers within the
 3618 structure. The returned pointer, and pointers within the structure, might
 3619 be invalidated or the structure or the storage areas might be overwritten
 3620 by a subsequent call to getnetbyaddr(), getnetbyname(), or getnetent().

3621

3622 *Rationale:* Austin Group Defect Report(s) applied: 75.3623 See <http://austingroupbugs.net/view.php?id=75>

3624

3625

3626 **Change Number: XSH/TC1/D5/0084 [75]**

3627

3628

3629 On Page: 752 Line: 25181 Section: endnetent()

3630

3631 In the APPLICATION USAGE section, change from:

3632

3633 The getnetbyaddr(), getnetbyname(), and getnetent() functions may return
 3634 pointers to static data, which may be overwritten by subsequent calls
 3635 to any of these functions.

3636

3637 to:

3638

3639 None.

3640

3641 *Rationale:* Austin Group Defect Report(s) applied: 75.3642 See <http://austingroupbugs.net/view.php?id=75>

3643

3644

3645 **Change Number: XSH/TC1/D5/0085 [75]**

3646

3647

3648 On Page: 753 Line: 25230 Section: endprotoent()

3649

3650 In the RETURN VALUE section, add a new paragraph to the end of the
3651 section:

3652

3653 The application shall not modify the structure to which the return
 3654 value points, nor any storage areas pointed to by pointers within the
 3655 structure. The returned pointer, and pointers within the structure,
 3656 might be invalidated or the structure or the storage areas might be
 3657 overwritten by a subsequent call to getprotobyname(), getprotobynumber(),
 3658 or getprotoent().

3659

3660 *Rationale:* Austin Group Defect Report(s) applied: 75.3661 See <http://austingroupbugs.net/view.php?id=75>

3662

3663

3664 **Change Number: XSH/TC1/D5/0086 [75]**

3665

3666

3667 On Page: 754 Line: 25236 Section: endprotent()

3668

3669 In the APPLICATION USAGE section, change from:

3670

3671 The `getprotobyname()`, `getprotobynumber()`, and `getprotoent()` functions may
3672 return pointers to static data, which may be overwritten by subsequent
3673 calls to any of these functions.

3674

3675 to:

3676

3677 None.

3678

3679 *Rationale:* Austin Group Defect Report(s) applied: 75.3680 See <http://austingroupbugs.net/view.php?id=75>

3681

3682

3683 **Change Number: XSH/TC1/D5/0087 [75]**

3684

3685

3686 On Page: 755 Line: 25271 Section: `endpwent()`

3687

3688 In the RETURN VALUE section, add a new paragraph to the end of the
3689 section:

3690

3691 The application shall not modify the structure to which the return
3692 value points, nor any storage areas pointed to by pointers within the
3693 structure. The returned pointer, and pointers within the structure,
3694 might be invalidated or the structure or the storage areas might be
3695 overwritten by a subsequent call to `getpwuid()`, `getpwnam()`, or `getpwent()`.

3696

3697 On L25278, delete from the ERRORS section:

3698

3699 The return value may point to a static area which is overwritten by a
3700 subsequent call to `getpwuid()`, `getpwnam()`, or `getpwent()`.

3701

3702 *Rationale:* Austin Group Defect Report(s) applied: 75.3703 See <http://austingroupbugs.net/view.php?id=75>

3704

3705

3706 **Change Number: XSH/TC1/D5/0088 [75]**

3707

3708

3709 On Page: 758 Line: 25362 Section: `endservent()`

3710

3711 In the RETURN VALUE section, add a new paragraph to the end of the
3712 section:

3713

3714 The application shall not modify the structure to which the return
3715 value points, nor any storage areas pointed to by pointers within the
3716 structure. The returned pointer, and pointers within the structure, might
3717 be invalidated or the structure or the storage areas might be overwritten
3718 by a subsequent call to `getservbyname()`, `getservbyport()`, or `getservent()`.

3719

3720 *Rationale:* Austin Group Defect Report(s) applied: 75.3721 See <http://austingroupbugs.net/view.php?id=75>

3722

3723

3724 **Change Number: XSH/TC1/D5/0089 [75]**

3725

3726

3727 On Page: 759 Line: 25370 Section: `endservent()`

3728

3729 In the APPLICATION USAGE section, delete the paragraph:

3730

3731 The `getservbyname()`, `getservbyport()`, and `getservent()` functions may
3732 return pointers to static data, which may be overwritten by subsequent

3733 calls to any of these functions.

3734

3735 *Rationale:* Austin Group Defect Report(s) applied: 75.

3736 See <http://austingroupbugs.net/view.php?id=75>

3737

3738

3739 **Change Number: XSH/TC1/D5/0090** [213,428]

3740

3741

3742 On Page: 761 Line: 25428 Section: `endutxent()`

3743

3744 In the DESCRIPTION section, in the table, change from:

3745

3746 LOGIN_PROCESS `ut_id`, `ut_user` (implementation-defined name of the login
 3747 process), `ut_pid`, `ut_tv`

3748

3749 to:

3750

3751 LOGIN_PROCESS `ut_id`, `ut_user` (implementation-defined name of the login
 3752 process), `ut_line`, `ut_pid`, `ut_tv`

3753

3754 *Rationale:* Austin Group Defect Report(s) applied: 213,428.

3755 See <http://austingroupbugs.net/view.php?id=213>

3756 See <http://austingroupbugs.net/view.php?id=428>

3757 `ut_line` is added before `ut_pid` on L25429.

3758

3759

3760 **Change Number: XSH/TC1/D5/0091** [213]

3761

3762

3763 On Page: 761 Line: 25435 Section: `endutxent()`

3764

3765 In the DESCRIPTION section, change from:

3766

3767 If the process has appropriate privileges, the `pututxline()` function
 3768 shall write out the structure into the user accounting database. It
 3769 shall use `getutxid()` to search for a record that satisfies the request.
 3770 If this search succeeds, then the entry shall be replaced. Otherwise,
 3771 a new entry shall be made at the end of the user accounting database.

3772

3773 to:

3774

3775 If the process has appropriate privileges, the `pututxline()` function
 3776 shall write out the structure into the user accounting database. It shall
 3777 search for a record as if by `getutxid()` that satisfies the request. If
 3778 this search succeeds, then the entry shall be replaced. Otherwise,
 3779 a new entry shall be made at the end of the user accounting database.

3780

3781 *Rationale:* Austin Group Defect Report(s) applied: 213.

3782 See <http://austingroupbugs.net/view.php?id=213>

3783

3784

3785 **Change Number: XSH/TC1/D5/0092** [68]

3786

3787

3788 On Page: 765 Line: 25514 Section: `erf()`

3789

3790 In the RETURN VALUE section, change from:

3791

3792 [MX]If `x` is subnormal, a range error may occur, and $2 * x/\text{sqrt}(\pi)$
 3793 should be returned.[/MX]

3794

3795 to:

3796
3797 If the correct value would cause underflow, a range error may occur,
3798 and erf(), erff(), and erfl() shall return an implementation-defined
3799 value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN,
3800 respectively. [MXX]If IEC 60559 Floating-Point is supported, 2 *
3801 x/sqrt(pi) should be returned. [/MXX]
3802
3803 *Rationale:* Austin Group Defect Report(s) applied: 68.
3804 See <http://austingroupbugs.net/view.php?id=68>
3805
3806
3807 **Change Number: XSH/TC1/D5/0093 [68]**
3808
3809
3810 On Page: 767 Line: 25578 Section: erfc()
3811
3812 In the RETURN VALUE section, change from:
3813
3814 If the correct value would cause underflow and is not representable,
3815 a range error may occur and [MX]either 0.0 (if representable), or [MX]
3816 an implementation-defined value shall be returned.
3817
3818 to:
3819
3820 If the correct value would cause underflow, [MXX]and is not
3821 representable[/MXX], a range error may occur, and erf(), erfcf(), and
3822 erfl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is
3823 not supported) an implementation-defined value no greater in magnitude
3824 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.
3825
3826 *Rationale:* Austin Group Defect Report(s) applied: 68.
3827 See <http://austingroupbugs.net/view.php?id=68>
3828
3829
3830 **Change Number: XSH/TC1/D5/0094 [68]**
3831
3832
3833 On Page: 767 Line: 25584 Section: erfc()
3834
3835 Change MX shading to MXX for:
3836
3837 If the correct value would cause underflow and is representable, a range
3838 error may occur and the correct value shall be returned.
3839
3840 *Rationale:* Austin Group Defect Report(s) applied: 68.
3841 See <http://austingroupbugs.net/view.php?id=68>
3842
3843
3844 **Change Number: XSH/TC1/D5/0095 [386]**
3845
3846
3847 On Page: 772 Line: 25708 Section: exec
3848
3849 In the DESCRIPTION section, change from:
3850
3851 In addition, the following variable:
3852
3853 extern char **environ;
3854
3855 to:
3856
3857 In addition, the following variable, which must be declared by the user
3858 if it is to be used directly:

3859
 3860 extern char **environ;
 3861
 3862 *Rationale:* Austin Group Defect Report(s) applied: 386.
 3863 See <http://austingroupbugs.net/view.php?id=386>
 3864
 3865
 3866 **Change Number: XSH/TC1/D5/0096 [167]**
 3867
 3868
 3869 On Page: 772 Line: 25712 Section: exec
 3870
 3871 In the DESCRIPTION section, after L25712, add two new paragraphs:
 3872
 3873 Applications can change the entire environment in a single operation by
 3874 assigning the environ variable to point to an array of character pointers
 3875 to the new environment strings. After assigning a new value to environ,
 3876 applications should not rely on the new environment strings remaining
 3877 part of the environment, as a call to getenv(), [XSI]putenv(), [XSI]
 3878 setenv(), unsetenv() or any function that is dependent on an environment
 3879 variable may, on noticing that environ has changed, copy the environment
 3880 strings to a new array and assign environ to point to it.
 3881
 3882 Any application that directly modifies the pointers to which the environ
 3883 variable points has undefined behavior.
 3884
 3885 *Rationale:* Austin Group Defect Report(s) applied: 167.
 3886 See <http://austingroupbugs.net/view.php?id=167>
 3887
 3888
 3889 **Change Number: XSH/TC1/D5/0097 [291]**
 3890
 3891
 3892 On Page: 773 Line: 25737,25741 Section: exec
 3893
 3894 In the DESCRIPTION section, change both instances of "filename" to
 3895 "filename string".
 3896
 3897 *Rationale:* Austin Group Defect Report(s) applied: 291.
 3898 See <http://austingroupbugs.net/view.php?id=291>
 3899
 3900
 3901 **Change Number: XSH/TC1/D5/0098 [173]**
 3902
 3903
 3904 On Page: 773 Line: 25757 Section: exec
 3905
 3906 In the DESCRIPTION section, change from:
 3907
 3908 If file descriptors 0, 1, and 2 would otherwise be closed after a
 3909 successful call to one of the exec family of functions, and the new
 3910 process image file has the set-user-ID or set-group-ID file mode bits set,
 3911 and the ST_NOSUID bit is not set for the file system containing the new
 3912 process image file, implementations may open an unspecified file for
 3913 each of these file descriptors in the new process image.
 3914
 3915 to:
 3916
 3917 If file descriptor 0, 1, or 2 would otherwise be closed after a successful
 3918 call to one of the exec family of functions, implementations may open an
 3919 unspecified file for the file descriptor in the new process image. If
 3920 a standard utility or a conforming application is executed with file
 3921 descriptor 0 not open for reading or with file descriptor 1 or 2 not

3922 open for writing, the environment in which the utility or application
3923 is executed shall be deemed non-conforming, and consequently the utility
3924 or application might not behave as described in this standard.

3925

3926 *Rationale:* Austin Group Defect Report(s) applied: 173.

3927 See <http://austingroupbugs.net/view.php?id=173>

3928

3929

3930 **Change Number: XSH/TC1/D5/0099** [296]

3931

3932

3933 On Page: 777 Line: 25899 Section: exec

3934

3935 In the ERRORS section, [EACCES] error, change:

3936

3937 Search permission is denied for a directory listed in the new process
3938 image file's path prefix, or the new process image file denies execution
3939 permission, or the ...

3940

3941 to:

3942

3943 The ...

3944

3945 *Rationale:* Austin Group Defect Report(s) applied: 296.

3946 See <http://austingroupbugs.net/view.php?id=296>

3947

3948

3949 **Change Number: XSH/TC1/D5/0100** [324]

3950

3951

3952 On Page: 777 Line: 25912 Section: exec

3953

3954 In the ERRORS section, [ENOTDIR] error, change from:

3955

3956 A component of the new process image file's path prefix is not a
3957 directory, ...

3958

3959 to:

3960

3961 A component of the new process image file's path prefix names an existing
3962 file that is neither a directory nor a symbolic link to a directory, ...

3963

3964 *Rationale:* Austin Group Defect Report(s) applied: 324.

3965 See <http://austingroupbugs.net/view.php?id=324>

3966 This is an editorial issue clarifying the intent of the
3967 standard.

3968

3969

3970 **Change Number: XSH/TC1/D5/0101** [296]

3971

3972

3973 On Page: 777 Line: 25917-25931 Section: exec

3974

3975 In the ERRORS section, at L25917 before:

3976

3977 The exec functions, except for `execlp()` and `execvp()`, shall fail if:

3978

3979 insert:

3980

3981 The exec functions, except for `fexecve()`, shall fail if:

3982

3983 [EACCES] Search permission is denied for a directory listed in the new
3984 process image file's path prefix, or the new process image file denies

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

3985 execution permission.
 3986
 3987 Move the following errors from the indicated line numbers to after this
 3988 inserted [EACCES] error:
 3989
 3990 L25906 [ELOOP]
 3991 L25908 [ENAMETOOLONG]
 3992 L25910 [ENOENT]
 3993 L25912 [ENOTDIR]
 3994
 3995 At L25931 (after [ENOMEM]) add:
 3996
 3997 The exec functions, except for fexecve(), may fail if:
 3998
 3999 Move the following errors from the indicated line numbers to after this
 4000 added line:
 4001
 4002 L25923 [ELOOP]
 4003 L25925 [ENAMETOOLONG]
 4004
 4005 (This leaves just ENOMEM under "The exec functions may fail if:".)
 4006
 4007 *Rationale:* Austin Group Defect Report(s) applied: 296.
 4008 See <http://austingroupbugs.net/view.php?id=296>
 4009
 4010
 4011 **Change Number: XSH/TC1/D5/0102 [302]**
 4012
 4013
 4014 On Page: 779 Line: 25987 Section: exec
 4015
 4016 In the APPLICATION USAGE section, change from:
 4017
 4018 Applications that require other than the default POSIX locale should call
 4019 setlocale() with the appropriate parameters to establish the locale of
 4020 the new process.
 4021
 4022 to:
 4023
 4024 Applications that require other than the default POSIX locale as the
 4025 global locale in the new process image should call setlocale() with the
 4026 appropriate parameters.
 4027
 4028 *Rationale:* Austin Group Defect Report(s) applied: 302.
 4029 See <http://austingroupbugs.net/view.php?id=302>
 4030 These changes were overlooked during the revision when
 4031 per-thread locales were added.
 4032
 4033
 4034 **Change Number: XSH/TC1/D5/0103 [167]**
 4035
 4036
 4037 On Page: 779 Line: 25989 Section: exec
 4038
 4039 In the APPLICATION USAGE section, change from:
 4040
 4041 The environ array should not be accessed directly by the application.
 4042
 4043 The new process might be invoked in a non-conforming environment if the
 4044 envp array does not contain implementation-defined variables required
 4045 by the implementation to provide a conforming environment. See the
 4046 `_CS_V7_ENV` entry in `<unistd.h>` and `confstr()` for details.
 4047

4048 to:

4049

4050 When assigning a new value to the environ variable, applications should
4051 ensure that the environment to which it will point contains at least
4052 the following:

4053

4054 a. Any implementation-defined variables required by the implementation to
4055 provide a conforming environment. See the `_CS_V7_ENV` entry in `<unistd.h>`
4056 and `confstr()` for details.

4057

4058 b. A value for `PATH` which finds conforming versions of all standard
4059 utilities before any other versions.

4060

4061 The same constraint applies to the `envp` array passed to `execle()` or
4062 `execve()`, in order to ensure that the new process image is invoked in
4063 a conforming environment.

4064

4065 *Rationale:* Austin Group Defect Report(s) applied: 167.

4066 See <http://austingroupbugs.net/view.php?id=167>

4067

4068

4069 **Change Number: XSH/TC1/D5/0104 [173]**

4070

4071

4072 On Page: 779 Line: 25993 Section: `exec`

4073

4074 In the `APPLICATION USAGE` section, change from:

4075

4076 Applications should not depend on file descriptors 0, 1, and 2 being
4077 closed after an `exec`. A future version may allow these file descriptors
4078 to be automatically opened for any process.

4079

4080 to:

4081

4082 Applications should not execute programs with file descriptor 0 not open
4083 for reading or with file descriptor 1 or 2 not open for writing, as this
4084 might cause the executed program to misbehave. In order not to pass on
4085 these file descriptors to an executed program, applications should not
4086 just close them but should reopen them on, for example, `/dev/null`. Some
4087 implementations may reopen them automatically, but applications should
4088 not rely on this being done.

4089

4090 *Rationale:* Austin Group Defect Report(s) applied: 173.

4091 See <http://austingroupbugs.net/view.php?id=173>

4092

4093

4094 **Change Number: XSH/TC1/D5/0105 [291,429]**

4095

4096

4097 On Page: 780 Line: 26018 Section: `exec`

4098

4099 In the `RATIONALE` section, change from:

4100

4101 The requirement on a Strictly Conforming POSIX Application also states
4102 that the value passed as the first argument be a filename associated
4103 with the process being started. Although some existing applications pass
4104 a pathname rather than a filename in some circumstances, a filename is
4105 more generally useful, since the common usage of `argv[0]` is in printing
4106 diagnostics. In some cases the filename passed is not the actual filename
4107 of the file; for example, many implementations of the `login` utility
4108 use a convention of prefixing a `<hyphen>` ('-') to the actual filename,
4109 which indicates to the command interpreter being invoked that it is a
4110 "login shell".

4111
4112 to:
4113
4114 The requirement on a Strictly Conforming POSIX Application also states
4115 that the value passed as the first argument be a filename string
4116 associated with the process being started. Although some existing
4117 applications pass a pathname rather than a filename string in some
4118 circumstances, a filename string is more generally useful, since the
4119 common usage of `argv[0]` is in printing diagnostics. In some cases the
4120 filename passed is not the actual filename of the file; for example,
4121 many implementations of the login utility use a convention of prefixing
4122 a <hyphen> ('-') to the actual filename, which indicates to the command
4123 interpreter being invoked that it is a "login shell".
4124
4125 *Rationale:* Austin Group Defect Report(s) applied: 291,429.
4126 See <http://austingroupbugs.net/view.php?id=291>
4127 See <http://austingroupbugs.net/view.php?id=429>
4128
4129
4130 **Change Number: XSH/TC1/D5/0106 [68]**
4131
4132
4133 On Page: 786 Line: 26265 Section: `exp()`
4134
4135 In the RETURN VALUE section, change from:
4136
4137 If the correct value would cause underflow, and is not representable,
4138 a range error may occur, and `[MX]` either 0.0 (if supported), or `[MX]`
4139 an implementation-defined value shall be returned.
4140
4141 to:
4142
4143 If the correct value would cause underflow, `[MXX]` and is not
4144 representable `[MXX]`, a range error may occur, and `exp()`, `expf()`, and
4145 `expl()` shall return `[MXX]0.0`, or `[MXX]` (if IEC 60559 Floating-Point is
4146 not supported) an implementation-defined value no greater in magnitude
4147 than `DBL_MIN`, `FLT_MIN`, and `LDBL_MIN`, respectively.
4148
4149 *Rationale:* Austin Group Defect Report(s) applied: 68.
4150 See <http://austingroupbugs.net/view.php?id=68>
4151
4152
4153 **Change Number: XSH/TC1/D5/0107 [68]**
4154
4155
4156 On Page: 786 Line: 26271 Section: `exp()`
4157
4158 In the RETURN VALUE section, change the MX shading to MXX for:
4159
4160 If the correct value would cause underflow, and is representable,
4161 a range error may occur and the correct value shall be returned.
4162
4163 *Rationale:* Austin Group Defect Report(s) applied: 68.
4164 See <http://austingroupbugs.net/view.php?id=68>
4165
4166
4167 **Change Number: XSH/TC1/D5/0108 [68]**
4168
4169
4170 On Page: 788 Line: 26342 Section: `exp2()`
4171
4172 In the RETURN VALUE section, change from:
4173

4174 If the correct value would cause underflow, and is not representable,
4175 a range error may occur, and [MX]either 0.0 (if supported), or[/MX]
4176 an implementation-defined value shall be returned.
4177
4178 to:
4179
4180 If the correct value would cause underflow, [MXX]and is not
4181 representable[/MXX], a range error may occur, and exp2(), exp2f(), and
4182 exp2l() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is
4183 not supported) an implementation-defined value no greater in magnitude
4184 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.
4185
4186 *Rationale:* Austin Group Defect Report(s) applied: 68.
4187 See <http://austingroupbugs.net/view.php?id=68>
4188
4189
4190 **Change Number: XSH/TC1/D5/0109 [68]**
4191
4192
4193 On Page: 788 Line: 26348 Section: exp2()
4194
4195 In the RETURN VALUE section, change the MX shading to MXX for:
4196
4197 If the correct value would cause underflow, and is representable,
4198 a range error may occur and the correct value shall be returned.
4199
4200 *Rationale:* Austin Group Defect Report(s) applied: 68.
4201 See <http://austingroupbugs.net/view.php?id=68>
4202
4203
4204 **Change Number: XSH/TC1/D5/0110 [68]**
4205
4206
4207 On Page: 790 Line: 26404 Section: expm1()
4208
4209 In the RETURN VALUE section, change from:
4210
4211 [MX]If x is subnormal, a range error may occur and x should be
4212 returned.[/MX]
4213
4214 to:
4215
4216 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
4217 be returned.[/MXX].
4218
4219 [MX]If x is not returned, expm1(), expm1f(), and expm1l() shall return
4220 an implementation-defined value no greater in magnitude than DBL_MIN,
4221 FLT_MIN, and LDBL_MIN, respectively.[/MX]
4222
4223 *Rationale:* Austin Group Defect Report(s) applied: 68.
4224 See <http://austingroupbugs.net/view.php?id=68>
4225
4226
4227 **Change Number: XSH/TC1/D5/0111 [146,324]**
4228
4229
4230 On Page: 795 Line: 26553 Section: fattach()
4231
4232 In the ERRORS section, change from:
4233
4234 [ENOTDIR] A component of the path prefix is not a directory, or the
4235 path argument contains at least one non-`<slash>` character and ends with
4236 one or more trailing `<slash>` characters and the last pathname component

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

4237 names an existing file that is neither a directory nor a symbolic link
 4238 to a directory.
 4239
 4240 to:
 4241
 4242 [ENOTDIR] A component of the path prefix names an existing file that
 4243 is neither a directory nor a symbolic link to a directory, or the path
 4244 argument contains at least one non-`<slash>` character and ends with one
 4245 or more trailing `<slash>` characters.
 4246
 4247 *Rationale:* Austin Group Defect Report(s) applied: 146,324.
 4248 See <http://austingroupbugs.net/view.php?id=146>
 4249 See <http://austingroupbugs.net/view.php?id=324>
 4250 The [ENOTDIR] error is updated, including an editorial
 4251 clarification for the component of the path prefix.
 4252
 4253
 4254 **Change Number: XSH/TC1/D5/0112 [291]**
 4255
 4256
 4257 On Page: 796 Line: 26577-26579 Section: `fattach()`
 4258
 4259 In the EXAMPLES section, change both instances of "filename" to
 4260 "pathname".
 4261
 4262 *Rationale:* Austin Group Defect Report(s) applied: 291.
 4263 See <http://austingroupbugs.net/view.php?id=291>
 4264
 4265
 4266 **Change Number: XSH/TC1/D5/0113 [87]**
 4267
 4268
 4269 On Page: 805 Line: 26801 Section: `fclose()`
 4270
 4271 In the DESCRIPTION section, change from:
 4272
 4273 ... the file offset of the underlying open file description shall be
 4274 adjusted so that the next operation on the open file description deals
 4275 with the byte after the last one read from or written to the stream
 4276 being closed.
 4277
 4278 to:
 4279
 4280 ... the file offset of the underlying open file description shall be
 4281 set to the file position of the stream if the stream is the active
 4282 handle to the underlying file description.
 4283
 4284 *Rationale:* Austin Group Defect Report(s) applied: 87.
 4285 See <http://austingroupbugs.net/view.php?id=87>
 4286
 4287 **Change Number: XSH/TC1/D5/0114 [79]**
 4288
 4289
 4290 On Page: 805 Line: 26824 Section: `fclose()`
 4291
 4292 In the ERRORS section [EIO], change from:
 4293
 4294 ... the process is neither ignoring nor blocking SIGTTOU, and the process
 4295 group of the process is orphaned.
 4296
 4297 to:
 4298
 4299 ... the calling thread is not blocking SIGTTOU, the process is not

4300 ignoring SIGTTOU, and the process group of the process is orphaned.
4301
4302 *Rationale:* Austin Group Defect Report(s) applied: 79.
4303 See <http://austingroupbugs.net/view.php?id=79>
4304
4305 **Change Number: XSH/TC1/D5/0115 [14]**
4306
4307
4308 On Page: 806 Line: 26845 Section: `fclose()`
4309
4310 In the SEE ALSO section, add a reference to XSH Section 2.5.
4311
4312 *Rationale:* Austin Group Defect Report(s) applied: 14.
4313 See <http://austingroupbugs.net/view.php?id=14>
4314 This is an editorial improvement
4315
4316
4317 **Change Number: XSH/TC1/D5/0116 [141]**
4318
4319
4320 On Page: 807 Line: 26910 Section: `fcntl()`
4321
4322 In the DESCRIPTION section (for `F_SETFL`), add to the end of the paragraph
4323 the additional sentence:
4324
4325 If `fildev` does not support non-blocking operations, it is unspecified
4326 whether the `O_NONBLOCK` flag will be ignored.
4327
4328 *Rationale:* Austin Group Defect Report(s) applied: 141.
4329 See <http://austingroupbugs.net/view.php?id=141>
4330
4331
4332 **Change Number: XSH/TC1/D5/0117 [324]**
4333
4334
4335 On Page: 816 Line: 27274 Section: `fdetach()`
4336
4337 In the ERRORS section, for the `[ENOTDIR]` error, change from:
4338
4339 A component of the path prefix is not a directory, ...
4340
4341 to:
4342
4343 A component of the path prefix names an existing file that is neither
4344 a directory nor a symbolic link to a directory, ...
4345
4346 *Rationale:* Austin Group Defect Report(s) applied: 324.
4347 See <http://austingroupbugs.net/view.php?id=324>
4348 This is an editorial issue clarifying the intent of the
4349 standard.
4350
4351
4352 **Change Number: XSH/TC1/D5/0118 [291]**
4353
4354
4355 On Page: 817 Line: L27294-27296 Section: `fdetach()`
4356
4357 In the EXAMPLES section, change both instances of "filename" to
4358 "pathname".
4359
4360 *Rationale:* Austin Group Defect Report(s) applied: 291.
4361 See <http://austingroupbugs.net/view.php?id=291>
4362

4363
 4364 **Change Number: XSH/TC1/D5/0119** [68,428]
 4365
 4366
 4367 On Page: 818 Line: 27340 Section: fdim()
 4368
 4369 In the RETURN VALUE section, change from:
 4370
 4371 If x-y is positive and underflows, a range error may occur, and either
 4372 (x-y) (if representable), [XSI] or 0.0 (if supported), [/XSI] or an
 4373 implementation-defined value shall be returned.
 4374
 4375 to:
 4376
 4377 If the correct value would cause underflow, a range error may occur, and
 4378 fdim(), fdimf(), and fdiml() shall return [MXX]the correct value, or[/MXX]
 4379 (if IEC 60559 Floating-Point is not supported) an implementation-defined
 4380 value no greater in magnitude than DBL_MIN, FLT_MIN, and LDBL_MIN,
 4381 respectively.
 4382
 4383 *Rationale:* Austin Group Defect Report(s) applied: 68,428.
 4384 See <http://austingroupbugs.net/view.php?id=68>
 4385 See <http://austingroupbugs.net/view.php?id=428>
 4386
 4387
 4388 **Change Number: XSH/TC1/D5/0120** [68,428]
 4389
 4390
 4391 On Page: 819 Line: 27359 Section: fdim()
 4392
 4393 In the APPLICATION USAGE section, delete the following text:
 4394
 4395 On implementations supporting IEEE Std 754-1985, x-y cannot underflow, and
 4396 hence the 0.0 return value is shaded as an extension for implementations
 4397 supporting the XSI option rather than an MX extension.
 4398
 4399 *Rationale:* Austin Group Defect Report(s) applied: 68,428.
 4400 See <http://austingroupbugs.net/view.php?id=68>
 4401 See <http://austingroupbugs.net/view.php?id=428>
 4402
 4403
 4404 **Change Number: XSH/TC1/D5/0121** [409]
 4405
 4406
 4407 On Page: 821 Line: 27421 Section: fdopen()
 4408
 4409 In the RATIONALE section, change the text style from:
 4410
 4411 *<italics>a b</italics>*
 4412
 4413 to:
 4414
 4415 a *<italics>b</italics>*
 4416
 4417 *Rationale:* Austin Group Defect Report(s) applied: 409.
 4418 See <http://austingroupbugs.net/view.php?id=409>
 4419
 4420
 4421 **Change Number: XSH/TC1/D5/0122** [422]
 4422
 4423
 4424 On Page: 823 Line: 27468 Section: fdopendir()
 4425

4426 In the DESCRIPTION section, change from:
4427
4428 Upon successful return from fdopendir(), the file descriptor is under
4429 the control of the system, and if any attempt is made to close the file
4430 descriptor, or to modify the state of the associated description, other
4431 than by means of closedir(), readdir(), readdir_r(), or rewinddir(), ...
4432
4433 to:
4434
4435 Upon successful return from fdopendir(), the file descriptor is under
4436 the control of the system, and if any attempt is made to close the file
4437 descriptor, or to modify the state of the associated description, other
4438 than by means of closedir(), readdir(), readdir_r(), rewinddir(),
4439 or [XSI]seekdir()[/XSI], ...
4440
4441 Rationale: Austin Group Defect Report(s) applied: 422.
4442 See <http://austingroupbugs.net/view.php?id=422>
4443
4444
4445 **Change Number: XSH/TC1/D5/0123 [324]**
4446
4447
4448 On Page: 823 Line: 27494 Section: fdopendir()
4449
4450 In the ERRORS section, for the [ENOTDIR] error, change from:
4451
4452 A component of dirname is not a directory.
4453
4454 to:
4455
4456 A component of dirname names an existing file that is neither
4457 a directory nor a symbolic link to a directory.
4458
4459 Rationale: Austin Group Defect Report(s) applied: 324.
4460 See <http://austingroupbugs.net/view.php?id=324>
4461 This is an editorial issue clarifying the intent of the
4462 standard.
4463
4464
4465 **Change Number: XSH/TC1/D5/0124 [401]**
4466
4467
4468 On Page: 833 Line: 27807 Section: feof()
4469
4470 In the DESCRIPTION section, add to the end of the section:
4471
4472 [CX]The feof() function shall not change the setting of errno if
4473 stream is valid.[/CX]
4474
4475 Rationale: Austin Group Defect Report(s) applied: 401.
4476 See <http://austingroupbugs.net/view.php?id=401>
4477
4478
4479 **Change Number: XSH/TC1/D5/0125 [401]**
4480
4481
4482 On Page: 835 Line: 27871 Section: ferror()
4483
4484 In the DESCRIPTION section, add to the end of the section:
4485
4486 [CX]The ferror() function shall not change the setting of errno if
4487 stream is valid.[/CX]
4488

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

4489 *Rationale:* Austin Group Defect Report(s) applied: 401.
4490 See <http://austingroupbugs.net/view.php?id=401>
4491
4492
4493 **Change Number: XSH/TC1/D5/0126 [87]**
4494
4495
4496 On Page: 844 Line: 28024 Section: `fflush()`
4497
4498 In the DESCRIPTION section, change from:
4499
4500 ... the file offset of the underlying open file description shall be
4501 adjusted so that the next operation on the open file description deals
4502 with the byte after the last one read from or written to the stream
4503 being flushed.
4504
4505 to:
4506
4507 ... the file offset of the underlying open file description shall be set
4508 to the file position of the stream, and any characters pushed back onto
4509 the stream by `ungetc()` or `ungetwc()` that have not subsequently been read
4510 from the stream shall be discarded (without further changing the file offset).
4511
4512 *Rationale:* Austin Group Defect Report(s) applied: 87.
4513 See <http://austingroupbugs.net/view.php?id=87>
4514
4515
4516 **Change Number: XSH/TC1/D5/0127 [79]**
4517
4518
4519 On Page: 844 Line: 28042 Section: `fflush()`
4520
4521 In the ERRORS section [EIO], change from:
4522
4523 ... the process is neither ignoring nor blocking SIGTTOU, and the process
4524 group of the process is orphaned.
4525
4526 to:
4527
4528 ... the calling thread is not blocking SIGTTOU, the process is not
4529 ignoring SIGTTOU, and the process group of the process is orphaned.
4530
4531 *Rationale:* Austin Group Defect Report(s) applied: 79.
4532 See <http://austingroupbugs.net/view.php?id=79>
4533
4534
4535 **Change Number: XSH/TC1/D5/0128 [14]**
4536
4537
4538 On Page: 846 Line: 28101 Section: `fflush()`
4539
4540 In the SEE ALSO section, add a reference to XSH Section 2.5.
4541
4542 *Rationale:* Austin Group Defect Report(s) applied: 14.
4543 See <http://austingroupbugs.net/view.php?id=14>
4544 This is an editorial improvement
4545
4546
4547 **Change Number: XSH/TC1/D5/0129 [79]**
4548
4549
4550 On Page: 848 Line: 28184 Section: `fgetc()`
4551

4552 In the ERRORS section [EIO], change from:
4553
4554 ... and either the process is ignoring or blocking the SIGTTIN signal
4555 or the process group is orphaned.
4556
4557 to:
4558
4559 ... and either the calling thread is blocking SIGTTIN or the process is
4560 ignoring SIGTTIN or the process group of the process is orphaned.
4561
4562 *Rationale:* Austin Group Defect Report(s) applied: 79.
4563 See <http://austingroupbugs.net/view.php?id=79>
4564
4565
4566 **Change Number: XSH/TC1/D5/0130 [14]**
4567
4568
4569 On Page: 849 Line: 28206 Section: fgetc()
4570
4571 In the SEE ALSO section, add a reference to XSH Section 2.5.
4572
4573 *Rationale:* Austin Group Defect Report(s) applied: 14.
4574 See <http://austingroupbugs.net/view.php?id=14>
4575 This is an editorial improvement
4576
4577
4578 **Change Number: XSH/TC1/D5/0131 [105]**
4579
4580
4581 On Page: 850 Line: 28241 Section: fgetpos()
4582
4583 In the DESCRIPTION section, add a new paragraph at the end of the
4584 DESCRIPTION:
4585
4586 The fgetpos() function shall not change the setting of errno if
4587 successful.
4588
4589 *Rationale:* Austin Group Defect Report(s) applied: 105.
4590 See <http://austingroupbugs.net/view.php?id=105>
4591
4592
4593 **Change Number: XSH/TC1/D5/0132 [122]**
4594
4595
4596 On Page: 850 Line: 28249 Section: fgetpos()
4597
4598 In the ERRORS section, change from:
4599
4600 The fgetpos() function shall fail if:
4601
4602 CX [EOVERFLOW] The current value of the file position cannot be
4603 represented correctly in an object of type fpos_t.
4604
4605 The fgetpos() function may fail if:
4606
4607 CX [EBADF] The file descriptor underlying stream is not valid.
4608
4609 CX [ESPIPE] The file descriptor underlying stream is associated with a
4610 pipe, FIFO, or socket.
4611
4612 to:
4613
4614 The fgetpos() function shall fail if:

4615
 4616 CX [EBADF] The file descriptor underlying stream is not valid.
 4617
 4618 CX [Eoverflow] The current value of the file position cannot be
 4619 represented correctly in an object of type fpos_t.
 4620
 4621 CX [ESPIPE] The file descriptor underlying stream is associated with a
 4622 pipe, FIFO, or socket.
 4623
 4624 *Rationale:* Austin Group Defect Report(s) applied: 122.
 4625 See <http://austingroupbugs.net/view.php?id=122>
 4626 The [EBADF] and [ESPIPE] errors are made mandatory.
 4627
 4628
 4629 **Change Number: XSH/TC1/D5/0133 [14]**
 4630
 4631
 4632 On Page: 850 Line: 28261 Section: fgetpos()
 4633
 4634 In the SEE ALSO section, add a reference to XSH Section 2.5.
 4635
 4636 *Rationale:* Austin Group Defect Report(s) applied: 14.
 4637 See <http://austingroupbugs.net/view.php?id=14>
 4638 This is an editorial improvement
 4639
 4640
 4641 **Change Number: XSH/TC1/D5/0134 [182]**
 4642
 4643
 4644 On Page: 852 Line: 28298–28308 Section: fgets()
 4645
 4646 In the EXAMPLES section, change from:
 4647
 4648 The following example uses fgets() to read each line of input.
 4649 {LINE_MAX}, which defines the maximum size of the input line, is defined
 4650 in the <limits.h> header.
 4651
 4652 #include <stdio.h>
 4653 ...
 4654 char line[LINE_MAX];
 4655 ...
 4656 while (fgets(line, LINE_MAX, fp) != NULL) {
 4657 ...
 4658 }
 4659 ...
 4660
 4661 to:
 4662
 4663 The following example uses fgets() to read lines of input. It assumes that
 4664 the file it is reading is a text file and that lines in this text file
 4665 are no longer than 16384 (or {LINE_MAX} if it is less than 16384 on the
 4666 implementation where it is running) bytes long. (Note that the standard
 4667 utilities have no line length limit if sysconf(_SC_LINE_MAX) returns -1
 4668 without setting errno. This example assumes that sysconf(_SC_LINE_MAX)
 4669 will not fail.)
 4670
 4671 #include <limits.h>
 4672 #include <stdio.h>
 4673 #include <unistd.h>
 4674 #define MYLIMIT 16384
 4675
 4676 char *line;
 4677 int line_max;

```
4678 if (LINE_MAX >= MYLIMIT) {
4679     // Use maximum line size of MYLIMIT. If LINE_MAX is
4680     // bigger than our limit, sysconf() can't report a
4681     // smaller limit.
4682     line_max = MYLIMIT;
4683 } else {
4684     long limit = sysconf(_SC_LINE_MAX);
4685     line_max = (limit < 0 || limit > MYLIMIT) ? MYLIMIT : (int)limit;
4686 }
4687
4688 // line_max + 1 leaves room for the nul byte added by fgets().
4689 line = malloc(line_max + 1);
4690 if (line == NULL) {
4691     // out of space
4692     ...
4693     return error;
4694 }
4695
4696 while (fgets(line, line_max + 1, fp) != NULL) {
4697     // Verify that a full line has been read...
4698     // If not, report an error or prepare to treat the
4699     // next time through the loop as a read of a
4700     // continuation of the current line.
4701     ...
4702     // Process line...
4703     ...
4704 }
4705 free(line);
4706 ...
4707
4708 Rationale: Austin Group Defect Report(s) applied: 182.
4709 See http://austingroupbugs.net/view.php?id=182
4710
4711 Change Number: XSH/TC1/D5/0135 [14]
4712
4713 On Page: 853 Line: 28316 Section: fgets()
4714
4715 In the SEE ALSO section, add a reference to XSH Section 2.5.
4716
4717 Rationale: Austin Group Defect Report(s) applied: 14.
4718 See http://austingroupbugs.net/view.php?id=14
4719 This is an editorial improvement
4720
4721 Change Number: XSH/TC1/D5/0136 [105]
4722
4723 On Page: 854 Line: 28343 Section: fgetwc()
4724
4725 In the DESCRIPTION section, add a new paragraph at the end of the
4726 DESCRIPTION:
4727 The fgetwc() function shall not change the setting of errno if
4728 successful.
4729
4730 Rationale: Austin Group Defect Report(s) applied: 105.
4731 See http://austingroupbugs.net/view.php?id=105
4732
4733 Change Number: XSH/TC1/D5/0137 [79]
4734
4735
```

4741
 4742 On Page: 854 Line: 28362 Section: fgetwc()
 4743
 4744 In the ERRORS section [EIO], change from:
 4745
 4746 ... and either the process is ignoring or blocking the SIGTTIN signal
 4747 or the process group is orphaned.
 4748
 4749 to:
 4750
 4751 ... and either the calling thread is blocking SIGTTIN or the process is
 4752 ignoring SIGTTIN or the process group of the process is orphaned.
 4753
 4754 *Rationale:* Austin Group Defect Report(s) applied: 79.
 4755 See <http://austingroupbugs.net/view.php?id=79>
 4756
 4757
 4758 **Change Number: XSH/TC1/D5/0138** [14]
 4759
 4760
 4761 On Page: 855 Line: 28381 Section: fgetwc()
 4762
 4763 In the SEE ALSO section, add a reference to XSH Section 2.5.
 4764
 4765 *Rationale:* Austin Group Defect Report(s) applied: 14.
 4766 See <http://austingroupbugs.net/view.php?id=14>
 4767 This is an editorial improvement
 4768
 4769
 4770 **Change Number: XSH/TC1/D5/0139** [14]
 4771
 4772
 4773 On Page: 856 Line: 28437 Section: fgetws()
 4774
 4775 In the SEE ALSO section, add a reference to XSH Section 2.5.
 4776
 4777 *Rationale:* Austin Group Defect Report(s) applied: 14.
 4778 See <http://austingroupbugs.net/view.php?id=14>
 4779 This is an editorial improvement
 4780
 4781
 4782 **Change Number: XSH/TC1/D5/0140** [118]
 4783
 4784
 4785 On Page: 859 Line: 28511 Section: flockfile()
 4786
 4787 In the DESCRIPTION section, change from:
 4788
 4789 All functions that reference (FILE *) objects shall behave ...
 4790
 4791 to:
 4792
 4793 All functions that reference (FILE *) objects, except those with names
 4794 ending in _unlocked, shall behave ...
 4795
 4796 *Rationale:* Austin Group Defect Report(s) applied: 118.
 4797 See <http://austingroupbugs.net/view.php?id=118>
 4798
 4799
 4800 **Change Number: XSH/TC1/D5/0141** [346]
 4801
 4802
 4803 On Page: 861 Line: 28567- Section: floor()

4804
4805 For floor(), delete P861 L28567-28570 (DESCRIPTION), L28577-28579
4806 (RETURN VALUE), and replace L28581-28586 (ERRORS) with:
4807
4808 No errors are defined.
4809
4810 At L28572 (RETURN VALUE), add:
4811
4812 [MX]The result shall have the same sign as x.[/MX]
4813
4814 At L28590 (APPLICATION USAGE), change "an int or long" to "an intmax_t",
4815 and replace L28593-28596 (APPLICATION USAGE) with:
4816
4817 These functions may raise the inexact floating-point exception if the
4818 result differs in value from the argument.
4819
4820 *Rationale:* Austin Group Defect Report(s) applied: 346.
4821 See <http://austingroupbugs.net/view.php?id=346>
4822
4823
4824 **Change Number: XSH/TC1/D5/0142 [461]**
4825
4826
4827 On Page: 866 Line: 28742 Section: fmemopen()
4828
4829 In the DESCRIPTION section, change from:
4830
4831 The mode argument is a character string having one of the following
4832 values:
4833
4834 to:
4835
4836 The mode argument points to a string. If the string is one of the
4837 following, the stream shall be opened in the indicated mode. Otherwise,
4838 the behavior is undefined.
4839
4840 *Rationale:* Austin Group Defect Report(s) applied: 461.
4841 See <http://austingroupbugs.net/view.php?id=461>
4842 This change is to match the wording used in fopen().
4843
4844
4845 **Change Number: XSH/TC1/D5/0143 [396]**
4846
4847
4848 On Page: 866 Line: 28743 Section: fmemopen()
4849
4850 In the DESCRIPTION section, change the first column from:
4851
4852 r or rb
4853 w or wb
4854 a or ab
4855 r+ or rb+ or r+b
4856 w+ or wb+ or w+b
4857 a+ or ab+ or a+b
4858
4859 to:
4860
4861 r
4862 w
4863 a
4864 r+
4865 w+
4866 a+

4867

4868 *Rationale:* Austin Group Defect Report(s) applied: 396.4869 See <http://austingroupbugs.net/view.php?id=396>

4870

4871

4872 **Change Number: XSH/TC1/D5/0144** [396]

4873

4874

4875 On Page: 866 Line: 28751 Section: fmemopen()

4876

4877 In the DESCRIPTION section, change from:

4878

4879 The character 'b' shall have no effect.

4880

4881 to:

4882

4883 Implementations shall accept all mode strings allowed by fopen(),
4884 but the use of the character 'b' shall produce implementation-defined
4885 results, where the resulting FILE * need not behave the same as if 'b'
4886 were omitted.

4887

4888 *Rationale:* Austin Group Defect Report(s) applied: 396.4889 See <http://austingroupbugs.net/view.php?id=396>

4890

4891

4892 **Change Number: XSH/TC1/D5/0145** [461]

4893

4894

4895 On Page: 866 Line: 28757 Section: fmemopen()

4896

4897 In the DESCRIPTION section, change from:

4898

4899 The stream maintains a current position in the buffer. This position is
4900 initially set to either the beginning of the buffer (for r and w modes)
4901 or to the first null byte in the buffer (for a modes). If no null byte
4902 is found in append mode, the initial position is set to one byte after
4903 the end of the buffer.

4904

4905 to:

4906

4907 The stream shall maintain a current position in the buffer. This position
4908 shall be initially set to either the beginning of the buffer (for r and
4909 w modes) or to the first null byte in the buffer (for a modes). If no
4910 null byte is found in append mode, the initial position shall be set to
4911 one byte after the end of the buffer.

4912

4913 *Rationale:* Austin Group Defect Report(s) applied: 461.4914 See <http://austingroupbugs.net/view.php?id=461>

4915

4916

4917 **Change Number: XSH/TC1/D5/0146** [461]

4918

4919

4920 On Page: 866 Line: 28762 Section: fmemopen()

4921

4922 In the DESCRIPTION section, change from:

4923

4924 The stream also maintains the size of the current buffer contents. For
4925 modes r and r+ the size is set to the value given by the size argument.
4926 For modes w and w+ the initial size is zero and for modes a and a+ the
4927 initial size is either the position of the first null byte in the buffer
4928 or the value of the size argument if no null byte is found.

4929

4930 to:

4931

4932 The stream shall also maintain the size of the current buffer contents;
4933 use of `fseek()` or `fseeko()` on the stream with `SEEK_END` shall seek
4934 relative to this size. For modes `r` and `r+` the size shall be set to the
4935 value given by the size argument. For modes `w` and `w+` the initial size
4936 shall be zero and for modes `a` and `a+` the initial size shall be either
4937 the position of the first null byte in the buffer or the value of the
4938 size argument if no null byte is found.

4939

4940 *Rationale:* Austin Group Defect Report(s) applied: 461.

4941 See <http://austingroupbugs.net/view.php?id=461>

4942

4943

4944 **Change Number: XSH/TC1/D5/0147** [461]

4945

4946

4947 On Page: 866 Line: 28766 Section: `fmemopen()`

4948

4949 In the DESCRIPTION section, change from:

4950

4951 A read operation on the stream cannot advance the current buffer
4952 position beyond the current buffer size. Reaching the buffer size in a
4953 read operation counts as "end-of-file". Null bytes in the buffer
4954 have no special meaning for reads. The read operation starts at the
4955 current buffer position of the stream.

4956

4957 to:

4958

4959 A read operation on the stream shall not advance the current buffer
4960 position beyond the current buffer size. Reaching the buffer size in a
4961 read operation shall count as "end-of-file". Null bytes in the buffer
4962 shall have no special meaning for reads. The read operation shall start
4963 at the current buffer position of the stream.

4964

4965 *Rationale:* Austin Group Defect Report(s) applied: 461.

4966 See <http://austingroupbugs.net/view.php?id=461>

4967

4968

4969 **Change Number: XSH/TC1/D5/0148** [461]

4970

4971

4972 On Page: 866 Line: 28770 Section: `fmemopen()`

4973

4974 In the DESCRIPTION section, change from:

4975

4976 A write operation starts either at the current position of the stream
4977 (if mode has not specified 'a' as the first character) or at the current
4978 size of the stream (if mode had 'a' as the first character). If the
4979 current position at the end of the write is larger than the current
4980 buffer size, the current buffer size is set to the current position. A
4981 write operation on the stream cannot advance the current buffer size
4982 beyond the size given in the size argument.

4983

4984 to:

4985

4986 A write operation shall start either at the current position of the stream
4987 (if mode has not specified 'a' as the first character) or at the current
4988 size of the stream (if mode had 'a' as the first character). If the
4989 current position at the end of the write is larger than the current
4990 buffer size, the current buffer size shall be set to the current position.
4991 A write operation on the stream shall not advance the current buffer size
4992 beyond the size given in the size argument.

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

4993

4994 *Rationale:* Austin Group Defect Report(s) applied: 461.4995 See <http://austingroupbugs.net/view.php?id=461>

4996

4997

4998 **Change Number: XSH/TC1/D5/0149** [461]

4999

5000

5001 On Page: 867 Line: 28775 Section: `fmemopen()`

5002

5003 In the DESCRIPTION section, change from:

5004

5005 When a stream open for writing is flushed or closed, a null byte is
 5006 written at the current position or at the end of the buffer, depending
 5007 on the size of the contents. If a stream open for update is flushed or
 5008 closed and the last write has advanced the current buffer size, a null
 5009 byte is written at the end of the buffer if it fits.

5010

5011 to:

5012

5013 When a stream open for writing is flushed or closed, a null byte shall
 5014 be written at the current position or at the end of the buffer, depending
 5015 on the size of the contents. If a stream open for update is flushed or
 5016 closed and the last write has advanced the current buffer size, a null
 5017 byte shall be written at the end of the buffer if it fits.

5018

5019 *Rationale:* Austin Group Defect Report(s) applied: 461.5020 See <http://austingroupbugs.net/view.php?id=461>

5021

5022

5023 **Change Number: XSH/TC1/D5/0150** [396]

5024

5025

5026 On Page: 868 Line: 28824 Section: `fmemopen()`

5027

5028 In the FUTURE DIRECTIONS section, change from:

5029

5030 None.

5031

5032 to:

5033

5034 A future revision of this standard may mandate specific behavior when
 5035 the mode argument includes 'b'.

5036

5037 *Rationale:* Austin Group Defect Report(s) applied: 396.5038 See <http://austingroupbugs.net/view.php?id=396>

5039

5040

5041 **Change Number: XSH/TC1/D5/0151** [68]

5042

5043

5044 On Page: 870 Line: 28885 Section: `fmod()`

5045

5046 In the RETURN VALUE section, change from:

5047

5048 If the correct value would cause underflow, and is not representable,
 5049 a range error may occur, and [MX]either 0.0 (if supported), or [MX]
 5050 an implementation-defined value shall be returned.

5051

5052 to:

5053

5054 If the correct value would cause underflow, [MXX]and is not
 5055 representable [MXX], a range error may occur, and `fmod()`, `modf()`, and

5056 fmodl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is
5057 not supported) an implementation-defined value no greater in magnitude
5058 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

5059

5060 *Rationale:* Austin Group Defect Report(s) applied: 68.

5061 See <http://austingroupbugs.net/view.php?id=68>

5062

5063

5064 **Change Number: XSH/TC1/D5/0152 [320]**

5065

5066

5067 On Page: 870 Line: 28888,28890 Section: fmod()

5068

5069 In the RETURN VALUE section, on lines 28888 and 28890 change from:

5070

5071 ... and either a NaN (if supported), or an implementation-defined
5072 value shall be returned.

5073

5074 to:

5075

5076 ... and a NaN shall be returned.

5077

5078 *Rationale:* Austin Group Defect Report(s) applied: 320.

5079 See <http://austingroupbugs.net/view.php?id=320>

5080

5081

5082 **Change Number: XSH/TC1/D5/0153 [68]**

5083

5084

5085 On Page: 870 Line: 28894 Section: fmod()

5086

5087 In the RETURN VALUE section, change the MX shading to MXX for:

5088

5089 If the correct value would cause underflow, and is representable,
5090 a range error may occur and the correct value shall be returned.

5091

5092 *Rationale:* Austin Group Defect Report(s) applied: 68.

5093 See <http://austingroupbugs.net/view.php?id=68>

5094

5095

5096 **Change Number: XSH/TC1/D5/0154 [291]**

5097

5098

5099 On Page: 875 Line: 29045 Section: fnmatch()

5100

5101 In the NAME section, change from:

5102

5103 filename

5104

5105 to:

5106

5107 filename string

5108

5109 *Rationale:* Austin Group Defect Report(s) applied: 291.

5110 See <http://austingroupbugs.net/view.php?id=291>

5111

5112

5113 **Change Number: XSH/TC1/D5/0155 [291]**

5114

5115

5116 On Page: 875 Line: 29084 Section: fnmatch()

5117

5118 In the APPLICATION USAGE section, change from:

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5119
 5120 match filenames, rather than pathnames
 5121
 5122 to:
 5123
 5124 match filename strings, rather than pathnames
 5125
 5126 *Rationale:* Austin Group Defect Report(s) applied: 291.
 5127 See <http://austingroupbugs.net/view.php?id=291>
 5128
 5129
 5130 **Change Number: XSH/TC1/D5/0156** [291,433]
 5131
 5132
 5133 On Page: 877 Line: 29108,29113,29143,29184 Section: fopen()
 5134
 5135 Change all instances of "filename" to "pathname".
 5136
 5137 *Rationale:* Austin Group Defect Report(s) applied: 291,433.
 5138 See <http://austingroupbugs.net/view.php?id=291>
 5139 See <http://austingroupbugs.net/view.php?id=433>
 5140
 5141
 5142 **Change Number: XSH/TC1/D5/0157** [146,433]
 5143
 5144
 5145 On Page: 878 Line: 29180 Section: fopen()
 5146
 5147 In the ERRORS section, change from:
 5148
 5149 [ENOENT] A component of filename does not name an existing file or
 5150 filename is an empty string.
 5151
 5152 to:
 5153
 5154 [ENOENT] The mode string begins with 'r' and a component of pathname
 5155 does not name an existing file, or mode begins with 'w' or 'a' and a
 5156 component of the path prefix of pathname does not name an existing file,
 5157 or pathname is an empty string.
 5158
 5159 [ENOENT] or [ENOTDIR] The pathname argument contains at least one
 5160 non-`<slash>` character and ends with one or more trailing `<slash>`
 5161 characters. If pathname names an existing file, an [ENOENT] error shall
 5162 not occur.
 5163
 5164 *Rationale:* Austin Group Defect Report(s) applied: 146,433.
 5165 See <http://austingroupbugs.net/view.php?id=146>
 5166 See <http://austingroupbugs.net/view.php?id=433>
 5167
 5168
 5169 **Change Number: XSH/TC1/D5/0158** [324]
 5170
 5171
 5172 On Page: 878 Line: 29184 Section: fopen()
 5173
 5174 In the ERRORS section, for the [ENOTDIR] error, change from:
 5175
 5176 A component of the path prefix is not a directory, ...
 5177
 5178 to:
 5179
 5180 A component of the path prefix names an existing file that is neither
 5181 a directory nor a symbolic link to a directory, ...

5182
5183 *Rationale:* Austin Group Defect Report(s) applied: 324.
5184 See <http://austingroupbugs.net/view.php?id=324>
5185 This is an editorial issue clarifying the intent of the
5186 standard.
5187
5188
5189 **Change Number: XSH/TC1/D5/0159** [14]
5190
5191
5192 On Page: 879 Line: 29227 Section: `fopen()`
5193
5194 In the SEE ALSO section, add a reference to XSH Section 2.5.
5195
5196 *Rationale:* Austin Group Defect Report(s) applied: 14.
5197 See <http://austingroupbugs.net/view.php?id=14>
5198 Editorial improvement.
5199
5200
5201 **Change Number: XSH/TC1/D5/0160** [256,428]
5202
5203
5204 On Page: 886 Line: 29481 Section: `fpathconf()`
5205
5206 In the DESCRIPTION section, change from:
5207
5208 3,4
5209
5210 to:
5211
5212 4,7
5213
5214 *Rationale:* Austin Group Defect Report(s) applied: 256,428.
5215 See <http://austingroupbugs.net/view.php?id=256>
5216 See <http://austingroupbugs.net/view.php?id=428>
5217
5218
5219 **Change Number: XSH/TC1/D5/0161** [256,428]
5220
5221
5222 On Page: 887 Line: 29510 Section: `fpathconf()`
5223
5224 In the DESCRIPTION section, change from:
5225
5226 5. If `path` or `files` refers to a directory, the value returned shall be
5227 the maximum length of a relative pathname when the specified directory
5228 is the working directory.
5229
5230 to:
5231
5232 5. If `path` or `files` refers to a directory, the value returned shall
5233 be the maximum length of a relative pathname that would not cross any
5234 mount points when the specified directory is the working directory.
5235
5236 *Rationale:* Austin Group Defect Report(s) applied: 256,428.
5237 See <http://austingroupbugs.net/view.php?id=256>
5238 See <http://austingroupbugs.net/view.php?id=428>
5239
5240
5241 **Change Number: XSH/TC1/D5/0162** [324]
5242
5243
5244 On Page: 888 Line: 29567 Section: `fpathconf()`

5245
 5246 In the ERRORS section, for the [ENOTDIR] error, change from:
 5247
 5248 A component of the path prefix is not a directory, ...
 5249
 5250 to:
 5251
 5252 A component of the path prefix names an existing file that is neither
 5253 a directory nor a symbolic link to a directory, ...
 5254
 5255 *Rationale:* Austin Group Defect Report(s) applied: 324.
 5256 See <http://austingroupbugs.net/view.php?id=324>
 5257 This is an editorial issue clarifying the intent of the
 5258 standard.
 5259
 5260
 5261 **Change Number: XSH/TC1/D5/0163 [302]**
 5262
 5263
 5264 On Page: 894 Line: 29763 Section: fprintf()
 5265
 5266 In the DESCRIPTION section, change from:
 5267
 5268 ... the process' locale ...
 5269
 5270 to:
 5271
 5272 ... the current locale ...
 5273
 5274 *Rationale:* Austin Group Defect Report(s) applied: 302.
 5275 See <http://austingroupbugs.net/view.php?id=302>
 5276 These changes were overlooked during the revision when
 5277 per-thread locales were added.
 5278
 5279
 5280 **Change Number: XSH/TC1/D5/0164 [316]**
 5281
 5282
 5283 On Page: 899 Line: 30004 Section: fprintf()
 5284
 5285 In the ERRORS section, add after the [EILSEQ] error:
 5286
 5287 [EOVERFLOW] [CX] The value to be returned is greater than {INT_MAX}.[/CX]
 5288
 5289 *Rationale:* Austin Group Defect Report(s) applied: 316.
 5290 See <http://austingroupbugs.net/view.php?id=316>
 5291
 5292
 5293 **Change Number: XSH/TC1/D5/0165 [316]**
 5294
 5295
 5296 On Page: 899 Line: 30012 Section: fprintf()
 5297
 5298 In the ERRORS section, for the snprintf() [EOVERFLOW] error, change from:
 5299
 5300 [EOVERFLOW] [CX]The value of n is greater than {INT_MAX} or the number
 5301 of bytes needed to hold the output excluding the terminating null is
 5302 greater than {INT_MAX}.
 5303
 5304 to:
 5305
 5306 [EOVERFLOW] [CX]The value of n is greater than {INT_MAX}.
 5307

5308 *Rationale:* Austin Group Defect Report(s) applied: 316.
5309 See <http://austingroupbugs.net/view.php?id=316>
5310
5311
5312 **Change Number:** XSH/TC1/D5/0166 [451,291]
5313
5314
5315 On Page: 902 Line: 30140 Section: fprintf()
5316
5317 In the EXAMPLES section, change from:
5318
5319 Creating a Filename
5320
5321 The following example creates a filename using information from a
5322 previous getpwnam() function that returned the HOME directory of the user.
5323
5324 #include <stdio.h>
5325 #include <sys/types.h>
5326 #include <unistd.h>
5327 ...
5328 char filename[PATH_MAX+1];
5329 struct passwd *pw;
5330 ...
5331 sprintf(filename, "%s/%d.out", pw->pw_dir, getpid());
5332 ...
5333
5334 to:
5335
5336 Creating a Pathname
5337
5338 The following example creates a pathname using information from a
5339 previous getpwnam() function that returned the password database
5340 entry of the user.
5341
5342 #include <stdint.h>
5343 #include <stdio.h>
5344 #include <stdlib.h>
5345 #include <string.h>
5346 #include <sys/types.h>
5347 #include <unistd.h>
5348 ...
5349 char *pathname;
5350 struct passwd *pw;
5351 size_t len;
5352 ...
5353 // digits required for pid_t is number of bits times log2(10) = approx 10/33
5354 len = strlen(pw->pw_dir) + 1 + 1+(sizeof(pid_t)*80+32)/33 + sizeof ".out";
5355 pathname = malloc(len);
5356 if (pathname != NULL)
5357 {
5358 sprintf(pathname, len, "%s/%jd.out", pw->pw_dir, (intmax_t)getpid());
5359 ...
5360 }
5361
5362 *Rationale:* Austin Group Defect Report(s) applied: 451,291.
5363 See <http://austingroupbugs.net/view.php?id=451>
5364 See <http://austingroupbugs.net/view.php?id=291>
5365
5366
5367 **Change Number:** XSH/TC1/D5/0167 [14]
5368
5369
5370 On Page: 904 Line: 30229 Section: fprintf()

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5371
 5372 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5373
 5374 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5375 See <http://austingroupbugs.net/view.php?id=14>
 5376 This is an editorial improvement
 5377
 5378
 5379 **Change Number: XSH/TC1/D5/0168 [79]**
 5380
 5381
 5382 On Page: 906 Line: 30300 Section: `fputc()`
 5383
 5384 In the ERRORS section [EIO], change from:
 5385
 5386 ... the process is neither ignoring nor blocking SIGTTOU, and the process
 5387 group of the process is orphaned.
 5388
 5389 to:
 5390
 5391 ... the calling thread is not blocking SIGTTOU, the process is not
 5392 ignoring SIGTTOU, and the process group of the process is orphaned.
 5393
 5394 *Rationale:* Austin Group Defect Report(s) applied: 79.
 5395 See <http://austingroupbugs.net/view.php?id=79>
 5396
 5397
 5398 **Change Number: XSH/TC1/D5/0169 [14]**
 5399
 5400
 5401 On Page: 907 Line: 30319 Section: `fputc()`
 5402
 5403 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5404
 5405 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5406 See <http://austingroupbugs.net/view.php?id=14>
 5407 This is an editorial improvement
 5408
 5409
 5410 **Change Number: XSH/TC1/D5/0170 [174,412]**
 5411
 5412
 5413 On Page: 908 Line: 30373 Section: `fputs()`
 5414
 5415 In the APPLICATION USAGE section, add to the end of the section:
 5416
 5417 This volume of POSIX.1-2008 requires that successful completion simply
 5418 return a non-negative integer. There are at least three known different
 5419 implementation conventions for this requirement:
 5420 * Return a constant value.
 5421 * Return the last character written.
 5422 * Return the number of bytes written. Note that this implementation
 5423 convention cannot be adhered to for strings longer than {INT_MAX}
 5424 bytes as the value would not be representable in the return type of
 5425 the function. For backwards compatibility, implementations can return
 5426 the number of bytes for strings of up to INT_MAX bytes, and return
 5427 INT_MAX for all longer strings.
 5428
 5429 *Rationale:* Austin Group Defect Report(s) applied: 174,412.
 5430 See <http://austingroupbugs.net/view.php?id=174>
 5431 See <http://austingroupbugs.net/view.php?id=412>
 5432
 5433

5434 **Change Number: XSH/TC1/D5/0171** [412]

5435

5436

5437 On Page: 908 Line: 30375 Section: fputs()

5438

5439 In the RATIONALE section, change from:

5440

5441 None

5442

5443 to:

5444

5445 The fputs() function is one whose source code was specified in `_The C`
5446 `Programming Language_` by Brian W. Kernighan and Dennis M. Ritchie. In
5447 the original edition, the function had no defined return value, yet many
5448 practical implementations would, as a side effect, return the value of the
5449 last character written as that was the value remaining in the accumulator
5450 used as a return value. In the second edition of the book, either the
5451 fixed value 0 or EOF would be returned depending upon the return value
5452 of `ferror()`; however, for compatibility with extant implementations,
5453 several implementations would, upon success, return a
5454 positive value representing last byte written.

5455

5456 *Rationale:* Austin Group Defect Report(s) applied: 412.

5457 See <http://austingroupbugs.net/view.php?id=412>

5458

5459

5460 **Change Number: XSH/TC1/D5/0172** [14]

5461

5462

5463 On Page: 909 Line: 30379 Section: fputs()

5464

5465 In the SEE ALSO section, add a reference to XSH Section 2.5.

5466

5467 *Rationale:* Austin Group Defect Report(s) applied: 14.

5468 See <http://austingroupbugs.net/view.php?id=14>

5469

5470 This is an editorial improvement

5471

5472 **Change Number: XSH/TC1/D5/0173** [105]

5473

5474

5475 On Page: 910 Line: 30406 Section: fputwc()

5476

5477 In the DESCRIPTION section, add a new paragraph at the end of the
5478 DESCRIPTION:

5479

5480 The fputwc() function shall not change the setting of `errno` if
5481 successful.

5482

5483 *Rationale:* Austin Group Defect Report(s) applied: 105.

5484 See <http://austingroupbugs.net/view.php?id=105>

5485

5486

5487 **Change Number: XSH/TC1/D5/0174** [79]

5488

5489

5490 On Page: 910 Line: 30426 Section: fputwc()

5491

5492 In the ERRORS section [EIO], change from:

5493

5494 ... the process is neither ignoring nor blocking SIGTTOU, and the process
5495 group of the process is orphaned.

5496

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5497 to:
 5498
 5499 ... the calling thread is not blocking SIGTTOU, the process is not
 5500 ignoring SIGTTOU, and the process group of the process is orphaned.
 5501
 5502 *Rationale:* Austin Group Defect Report(s) applied: 79.
 5503 See <http://austingroupbugs.net/view.php?id=79>
 5504
 5505
 5506 **Change Number: XSH/TC1/D5/0175 [14]**
 5507
 5508
 5509 On Page: 911 Line: 30445 Section: `fputc()`
 5510
 5511 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5512
 5513 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5514 See <http://austingroupbugs.net/view.php?id=14>
 5515 This is an editorial improvement
 5516
 5517
 5518 **Change Number: XSH/TC1/D5/0176 [412]**
 5519
 5520
 5521 On Page: 912 Line: 30489 Section: `fputws()`
 5522
 5523 In the APPLICATION USAGE section, add to the end of the section:
 5524
 5525 This volume of POSIX.1-2008 requires that successful completion simply
 5526 return a non-negative integer. There are at least three known different
 5527 implementation conventions for this requirement:
 5528 * Return a constant value.
 5529 * Return the last character written.
 5530 * Return the number of bytes written. Note that this implementation
 5531 convention cannot be adhered to for strings longer than `{INT_MAX}`
 5532 bytes as the value would not be representable in the return type of
 5533 the function. For backwards compatibility, implementations can return
 5534 the number of bytes for strings of up to `INT_MAX` bytes, and return
 5535 `INT_MAX` for all longer strings.
 5536
 5537 *Rationale:* Austin Group Defect Report(s) applied: 412.
 5538 See <http://austingroupbugs.net/view.php?id=412>
 5539
 5540
 5541 **Change Number: XSH/TC1/D5/0177 [14]**
 5542
 5543
 5544 On Page: 912 Line: 30495 Section: `fputws()`
 5545
 5546 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5547
 5548 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5549 See <http://austingroupbugs.net/view.php?id=14>
 5550 This is an editorial improvement
 5551
 5552
 5553 **Change Number: XSH/TC1/D5/0178 [232]**
 5554
 5555
 5556 On Page: 913 Line: 30537 Section: `fread()`
 5557
 5558 In the EXAMPLES section, change from:
 5559

5560 The following example reads a single element from the fp stream into
5561 the array pointed to by buf.

```
5562 #include <stdio.h>
5563 ...
5564 size_t bytes_read;
5565 char buf[100];
5566 FILE *fp;
5567 ...
5568 bytes_read = fread(buf, sizeof(buf), 1, fp);
5569 ...
5570
5571 to:
```

5572 The following example reads a single element from the fp stream into
5573 the array pointed to by buf.

```
5574 #include <stdio.h>
5575 ...
5576 size_t elements_read;
5577 char buf[100];
5578 FILE *fp;
5579 ...
5580 elements_read = fread(buf, sizeof(buf), 1, fp);
5581 ...
5582
```

5583 If a read error occurs, elements_read will be zero but the number of
5584 bytes read from the stream could be anything from zero to sizeof(buf)-1.

5585 The following example reads multiple single-byte elements from the fp
5586 stream into the array pointed to by buf.

```
5587 #include <stdio.h>
5588 ...
5589 size_t bytes_read;
5590 char buf[100];
5591 FILE *fp;
5592 ...
5593 bytes_read = fread(buf, 1, sizeof(buf), fp);
5594 ...
5595
```

5596 If a read error occurs, bytes_read will contain the number of bytes
5597 read from the stream.

5598 *Rationale:* Austin Group Defect Report(s) applied: 232.
5599 See <http://austingroupbugs.net/view.php?id=232>

5600 **Change Number: XSH/TC1/D5/0179 [14]**

5601 On Page: 914 Line: 30558 Section: fread()
5602

5603 In the SEE ALSO section, add a reference to XSH Section 2.5.
5604

5605 *Rationale:* Austin Group Defect Report(s) applied: 14.
5606 See <http://austingroupbugs.net/view.php?id=14>
5607 This is an editorial improvement

5608 **Change Number: XSH/TC1/D5/0180 [283]**

5609
5610

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5623 On Page: 921 Line: 30834 Section: freelocale()
 5624
 5625 In the DESCRIPTION section, add a new paragraph:
 5626
 5627 The behavior is undefined if the locobj argument is the special locale
 5628 object LC_GLOBAL_LOCALE or is not a valid locale object handle.
 5629
 5630 *Rationale:* Austin Group Defect Report(s) applied: 283.
 5631 See <http://austingroupbugs.net/view.php?id=283>
 5632
 5633
 5634 **Change Number: XSH/TC1/D5/0181** [291,433]
 5635
 5636
 5637 On Page: 923 Line: 30870 Section: freopen()
 5638
 5639 On L30870, L30877, L30881, L30885, L30893, L30914, L30927,
 5640 L30939, L30964, change all instances of "filename" to "pathname".
 5641
 5642 *Rationale:* Austin Group Defect Report(s) applied: 291,433.
 5643 See <http://austingroupbugs.net/view.php?id=291>
 5644 See <http://austingroupbugs.net/view.php?id=433>
 5645
 5646
 5647 **Change Number: XSH/TC1/D5/0182** [146,433]
 5648
 5649
 5650 On Page: 924 Line: 30923 Section: freopen()
 5651
 5652 In the ERRORS section, change from:
 5653
 5654 [ENOENT] A component of filename does not name an existing file or
 5655 filename is an empty string.
 5656
 5657 to:
 5658
 5659 [ENOENT] The mode string begins with 'r' and a component of pathname
 5660 does not name an existing file, or mode begins with 'w' or 'a' and a
 5661 component of the path prefix of pathname does not name an existing file,
 5662 or pathname is an empty string.
 5663
 5664 [ENOENT] or [ENOTDIR]: The pathname argument contains at least one
 5665 non-`<slash>` character and ends with one or more trailing `<slash>`
 5666 characters. If pathname names an existing file, an [ENOENT] error shall
 5667 not occur.
 5668
 5669 *Rationale:* Austin Group Defect Report(s) applied: 146,433.
 5670 See <http://austingroupbugs.net/view.php?id=146>
 5671 See <http://austingroupbugs.net/view.php?id=433>
 5672
 5673
 5674 **Change Number: XSH/TC1/D5/0183** [324]
 5675
 5676
 5677 On Page: 924 Line: 30927 Section: freopen()
 5678
 5679 In the ERRORS section, for the [ENOTDIR] error, change from:
 5680
 5681 A component of the path prefix is not a directory, ...
 5682
 5683 to:
 5684
 5685 A component of the path prefix names an existing file that is neither

5686 a directory nor a symbolic link to a directory, ...
5687
5688 *Rationale:* Austin Group Defect Report(s) applied: 324.
5689 See <http://austingroupbugs.net/view.php?id=324>
5690 This is an editorial issue clarifying the intent of the
5691 standard.
5692
5693
5694 **Change Number: XSH/TC1/D5/0184 [14]**
5695
5696
5697 On Page: 925 Line: 30983 Section: freopen()
5698
5699 In the SEE ALSO section, add a reference to XSH Section 2.5.
5700
5701 *Rationale:* Austin Group Defect Report(s) applied: 14.
5702 See <http://austingroupbugs.net/view.php?id=14>
5703 This is an editorial improvement
5704
5705
5706 **Change Number: XSH/TC1/D5/0185 [302]**
5707
5708
5709 On Page: 929 Line: 31097 Section: fscanf()
5710
5711 change from:
5712
5713 ... the locale of the process ...
5714
5715 to:
5716
5717 ... the current locale ...
5718
5719 *Rationale:* Austin Group Defect Report(s) applied: 302.
5720 See <http://austingroupbugs.net/view.php?id=302>
5721 These changes were overlooked during the revision when
5722 per-thread locales were added.
5723
5724
5725 **Change Number: XSH/TC1/D5/0186 [90]**
5726
5727
5728 On Page: 934 Line: 31304 Section: fscanf()
5729
5730 In the RETURN VALUE section, change from:
5731
5732 If the input ends before the first matching failure or conversion, EOF
5733 shall be returned. If any error occurs, EOF shall be returned, [CX]and
5734 errno shall be set to indicate the error[/CX]. If a read error occurs,
5735 the error indicator for the stream shall be set.
5736
5737 to:
5738
5739 If the input ends before the first conversion (if any) has completed,
5740 and without a matching failure having occurred, EOF shall be returned. If
5741 an error occurs before the first conversion (if any) has completed,
5742 and without a matching failure having occurred, EOF shall be returned
5743 [CX]and errno shall be set to indicate the error[/CX]. If a read error
5744 occurs, the error indicator for the stream shall be set.
5745
5746 Add to the CHANGE HISTORY section:
5747
5748 A change to the second sentence in RETURN VALUE is made to align with

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5749 expected wording changes in the next revision of the C standard.
 5750
 5751 *Rationale:* Austin Group Defect Report(s) applied: 90.
 5752 See <http://austingroupbugs.net/view.php?id=90>
 5753
 5754
 5755 **Change Number: XSH/TC1/D5/0187 [14]**
 5756
 5757
 5758 On Page: 935 Line: 31351 Section: `fscanf()`
 5759
 5760 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5761
 5762 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5763 See <http://austingroupbugs.net/view.php?id=14>
 5764 This is an editorial improvement
 5765
 5766
 5767 **Change Number: XSH/TC1/D5/0188 [79]**
 5768
 5769
 5770 On Page: 938 Line: 31438 Section: `fseek()`
 5771
 5772 In the ERRORS section [EIO], change from:
 5773
 5774 ... the process is neither ignoring nor blocking SIGTTOU, and the process
 5775 group of the process is orphaned.
 5776
 5777 to:
 5778
 5779 ... the calling thread is not blocking SIGTTOU, the process is not
 5780 ignoring SIGTTOU, and the process group of the process is orphaned.
 5781
 5782 *Rationale:* Austin Group Defect Report(s) applied: 79.
 5783 See <http://austingroupbugs.net/view.php?id=79>
 5784
 5785
 5786 **Change Number: XSH/TC1/D5/0189 [122]**
 5787
 5788
 5789 On Page: 938 Line: 31450 Section: `fseek()`
 5790
 5791 In the ERRORS section, change from:
 5792
 5793 CX [ESPIPE] The file descriptor underlying stream is associated with a
 5794 pipe or FIFO.
 5795
 5796 to:
 5797
 5798 CX [ESPIPE] The file descriptor underlying stream is associated with a
 5799 pipe, FIFO, or socket.
 5800
 5801 *Rationale:* Austin Group Defect Report(s) applied: 122.
 5802 See <http://austingroupbugs.net/view.php?id=122>
 5803 The [ESPIPE] error is updated to also apply to a socket.
 5804
 5805
 5806 **Change Number: XSH/TC1/D5/0190 [225]**
 5807
 5808
 5809 On Page: 938 Line: 31450 Section: `fseek()`
 5810
 5811 In the ERRORS section, at the end of the section, add a new paragraph:

5812
5813 The fseek() [CX]and fseeko()[/CX] functions may fail if:
5814
5815 and then move the existing [ENXIO] condition at L31442-31443 out of the
5816 shall fail into this new may fail section, still with CX margin marker
5817 and shading.
5818
5819 [ENXIO] A request was made of a nonexistent device, or the request was
5820 outside the capabilities of the device.
5821
5822 *Rationale:* Austin Group Defect Report(s) applied: 225.
5823 See <http://austingroupbugs.net/view.php?id=225>
5824
5825
5826 **Change Number: XSH/TC1/D5/0191 [14]**
5827
5828
5829 On Page: 938 Line: 31460 Section: fseek()
5830
5831 In the SEE ALSO section, add a reference to XSH Section 2.5.
5832
5833 *Rationale:* Austin Group Defect Report(s) applied: 14.
5834 See <http://austingroupbugs.net/view.php?id=14>
5835 This is an editorial improvement
5836
5837
5838 **Change Number: XSH/TC1/D5/0192 [105]**
5839
5840
5841 On Page: 940 Line: 31503 Section: fsetpos()
5842
5843 In the DESCRIPTION section, add a new paragraph at the end of the
5844 DESCRIPTION:
5845
5846 The fsetpos() function shall not change the setting of errno if
5847 successful.
5848
5849 *Rationale:* Austin Group Defect Report(s) applied: 105.
5850 See <http://austingroupbugs.net/view.php?id=105>
5851
5852
5853 **Change Number: XSH/TC1/D5/0193 [79]**
5854
5855
5856 On Page: 940 Line: 31523 Section: fsetpos()
5857
5858 In the ERRORS section [EIO], change from:
5859
5860 ... the process is neither ignoring nor blocking SIGTTOU, and the process
5861 group of the process is orphaned.
5862
5863 to:
5864
5865 ... the calling thread is not blocking SIGTTOU, the process is not
5866 ignoring SIGTTOU, and the process group of the process is orphaned.
5867
5868 *Rationale:* Austin Group Defect Report(s) applied: 79.
5869 See <http://austingroupbugs.net/view.php?id=79>
5870
5871
5872 **Change Number: XSH/TC1/D5/0194 [225]**
5873
5874

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

5875 On Page: 941 Line: 31531 Section: fsetpos()
 5876
 5877 In the ERRORS section, at the end of the section, add a new paragraph:
 5878
 5879 The fsetpos() function may fail if:
 5880
 5881 and then move the existing [ENXIO] condition at L31527-31528 out of the
 5882 shall fail into the new may fail section, still with CX margin marker
 5883 and shading.
 5884
 5885 *Rationale:* Austin Group Defect Report(s) applied: 225.
 5886 See <http://austingroupbugs.net/view.php?id=225>
 5887
 5888
 5889 **Change Number: XSH/TC1/D5/0195 [450]**
 5890
 5891
 5892 On Page: 941 Line: 31531 Section: fsetpos()
 5893
 5894 In the ERRORS section, change from:
 5895
 5896 [ESPIPE] The file descriptor underlying stream is associated with a pipe
 5897 or FIFO.
 5898
 5899 to:
 5900
 5901 [ESPIPE] The file descriptor underlying stream is associated with a pipe,
 5902 FIFO or socket.
 5903
 5904 *Rationale:* Austin Group Defect Report(s) applied: 450.
 5905 See <http://austingroupbugs.net/view.php?id=450>
 5906
 5907
 5908 **Change Number: XSH/TC1/D5/0196 [450]**
 5909
 5910
 5911 On Page: 941 Line: 31547 Section: fsetpos()
 5912
 5913 In the CHANGE HISTORY section, delete:
 5914
 5915 An additional [ESPIPE] error condition is added for sockets.
 5916
 5917 *Rationale:* Austin Group Defect Report(s) applied: 450.
 5918 See <http://austingroupbugs.net/view.php?id=450>
 5919
 5920
 5921 **Change Number: XSH/TC1/D5/0197 [14]**
 5922
 5923
 5924 On Page: 941 Line: 31541 Section: fsetpos()
 5925
 5926 In the SEE ALSO section, add a reference to XSH Section 2.5.
 5927
 5928 *Rationale:* Austin Group Defect Report(s) applied: 14.
 5929 See <http://austingroupbugs.net/view.php?id=14>
 5930 This is an editorial improvement
 5931
 5932
 5933 **Change Number: XSH/TC1/D5/0198 [461]**
 5934
 5935
 5936 On Page: 946 Line: 31694 Section: fstatat()
 5937

5938 In the DESCRIPTION section, change from:
5939
5940 ... the current working directory is used ...
5941
5942 to:
5943
5944 ... the current working directory shall be used ...
5945
5946 *Rationale:* Austin Group Defect Report(s) applied: 461.
5947 See <http://austingroupbugs.net/view.php?id=461>
5948
5949
5950 **Change Number: XSH/TC1/D5/0199 [324]**
5951
5952
5953 On Page: 946 Line: 31709 Section: `fstatat()`
5954
5955 In the ERRORS section, for the [ENOTDIR] error, change from:
5956
5957 A component of the path prefix is not a directory, ...
5958
5959 to:
5960
5961 A component of the path prefix names an existing file that is neither
5962 a directory nor a symbolic link to a directory, ...
5963
5964 *Rationale:* Austin Group Defect Report(s) applied: 324.
5965 See <http://austingroupbugs.net/view.php?id=324>
5966 This is an editorial issue clarifying the intent of the
5967 standard.
5968
5969
5970 **Change Number: XSH/TC1/D5/0200 [278]**
5971
5972
5973 On Page: 946 Line: 31720 `fstatat()`
5974
5975 In the ERRORS section, add (after the [EBADF] error):
5976
5977 [ENOTDIR] The path argument is not an absolute path and `fd` is a file
5978 descriptor associated with a non-directory file.
5979
5980 *Rationale:* Austin Group Defect Report(s) applied: 278.
5981 See <http://austingroupbugs.net/view.php?id=278>
5982
5983
5984 **Change Number: XSH/TC1/D5/0201 [278]**
5985
5986
5987 On Page: 947 Line: 31731 Section: `fstatat()`
5988
5989 In the ERRORS section, delete:
5990
5991 [ENOTDIR] The path argument is not an absolute path and `fd` is neither
5992 `AT_FDCWD` nor a file descriptor associated with a directory.
5993
5994 *Rationale:* Austin Group Defect Report(s) applied: 278.
5995 See <http://austingroupbugs.net/view.php?id=278>
5996
5997
5998 **Change Number: XSH/TC1/D5/0202 [291]**
5999
6000

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

6001 On Page: 948 Line: 31794 Section: fstatat()
 6002
 6003 In the EXAMPLES section, change from:
 6004
 6005 filename
 6006
 6007 to:
 6008
 6009 pathname
 6010
 6011 *Rationale:* Austin Group Defect Report(s) applied: 291.
 6012 See <http://austingroupbugs.net/view.php?id=291>
 6013
 6014
 6015 **Change Number: XSH/TC1/D5/0203 [324]**
 6016
 6017
 6018 On Page: 951 Line: 31896 Section: fstatvfs()
 6019
 6020 In the ERRORS section, for the [ENOTDIR] error, change from:
 6021
 6022 A component of the path prefix is not a directory, ...
 6023
 6024 to:
 6025
 6026 A component of the path prefix names an existing file that is neither
 6027 a directory nor a symbolic link to a directory, ...
 6028
 6029 *Rationale:* Austin Group Defect Report(s) applied: 324.
 6030 See <http://austingroupbugs.net/view.php?id=324>
 6031 This is an editorial issue clarifying the intent of the
 6032 standard.
 6033
 6034
 6035 **Change Number: XSH/TC1/D5/0204 [1050]**
 6036
 6037
 6038 On Page: 956 Line: 32041 Section: ftell()
 6039
 6040 In the DESCRIPTION, change from:
 6041
 6042 [CX]The ftello() function shall be equivalent to ftell(), except that
 6043 the return value is of type off_t. [/CX]
 6044
 6045 to:
 6046
 6047 The ftell() function shall not change the setting of errno if successful.
 6048
 6049 [CX]The ftello() function shall be equivalent to ftell(), except that
 6050 the return value is of type off_t and the ftello() function may change
 6051 the setting of errno if successful. [/CX]
 6052
 6053 *Rationale:* Austin Group Defect Report(s) applied: 105.
 6054 See <http://austingroupbugs.net/view.php?id=105>
 6055
 6056
 6057 **Change Number: XSH/TC1/D5/0205 [421]**
 6058
 6059
 6060 On Page: 956 Line: 32045 Section: ftell()
 6061
 6062 In the RETURN VALUE section, change from:
 6063

6064 Otherwise, `ftell()` and `ftello()` shall return `-1`, cast to `long`
6065 and `off_t` respectively, ...
6066
6067 to:
6068
6069 Otherwise, `ftell()` and `ftello()` shall return `-1`, ...
6070
6071 *Rationale:* Austin Group Defect Report(s) applied: 421.
6072 See <http://austingroupbugs.net/view.php?id=421>
6073
6074
6075 **Change Number: XSH/TC1/D5/0206 [122]**
6076
6077
6078 On Page: 956 Line: 32054 Section: `ftell()`
6079
6080 In the ERRORS section, change from:
6081
6082 [ESPIPE] The file descriptor underlying stream is associated with a pipe
6083 or FIFO.
6084
6085 to:
6086
6087 [ESPIPE] The file descriptor underlying stream is associated with a pipe,
6088 FIFO, or socket.
6089
6090 *Rationale:* Austin Group Defect Report(s) applied: 122.
6091 See <http://austingroupbugs.net/view.php?id=122>
6092
6093
6094 **Change Number: XSH/TC1/D5/0207 [122]**
6095
6096
6097 On Page: 956 Line: 32055 Section: `ftell()`
6098
6099 In the ERRORS section, delete:
6100
6101 The `ftell()` function may fail if:
6102
6103 CX [ESPIPE] The file descriptor underlying stream is associated with
6104 a socket.
6105
6106 *Rationale:* Austin Group Defect Report(s) applied: 122.
6107 See <http://austingroupbugs.net/view.php?id=122>
6108
6109
6110 **Change Number: XSH/TC1/D5/0208 [14]**
6111
6112
6113 On Page: 956 Line: 32066 Section: `ftell()`
6114
6115 In the SEE ALSO section, add a reference to XSH Section 2.5.
6116
6117 *Rationale:* Austin Group Defect Report(s) applied: 14.
6118 See <http://austingroupbugs.net/view.php?id=14>
6119 This is an editorial improvement
6120
6121
6122 **Change Number: XSH/TC1/D5/0209 [343]**
6123
6124
6125 On Page: 958 Line: 32087 Section: `ftok()`
6126

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

6127 In the DESCRIPTION section, change from:
 6128
 6129 ... existing file that the process is able to stat()
 6130
 6131 to:
 6132
 6133 ... existing file that the process is able to stat(), with the exception
 6134 that if stat() would fail with [Eoverflow] due to file size, ftok()
 6135 shall still succeed.
 6136
 6137 *Rationale:* Austin Group Defect Report(s) applied: 343.
 6138 See <http://austingroupbugs.net/view.php?id=343>
 6139
 6140
 6141 **Change Number: XSH/TC1/D5/0210 [366]**
 6142
 6143
 6144 On Page: 958 Line: 32089 Section: ftok()
 6145
 6146 In the DESCRIPTION section, change from:
 6147
 6148 ... and return different key values when called with different id ...
 6149
 6150 to:
 6151
 6152 ... and should return different key values when called with different id ...
 6153
 6154 *Rationale:* Austin Group Defect Report(s) applied: 366.
 6155 See <http://austingroupbugs.net/view.php?id=366>
 6156
 6157
 6158 **Change Number: XSH/TC1/D5/0211 [343]**
 6159
 6160
 6161 On Page: 958 Line: 32100 Section: ftok()
 6162
 6163 In the ERRORS section, after L32100, add a new paragraph:
 6164
 6165 [EIO] An error occurred while reading from the file system.
 6166
 6167 *Rationale:* Austin Group Defect Report(s) applied: 343.
 6168 See <http://austingroupbugs.net/view.php?id=343>
 6169
 6170
 6171 **Change Number: XSH/TC1/D5/0212 [324]**
 6172
 6173
 6174 On Page: 958 Line: 32106 Section: ftok()
 6175
 6176 In the ERRORS section, for the [ENOTDIR] error, change from:
 6177
 6178 A component of the path prefix is not a directory, ...
 6179
 6180 to:
 6181
 6182 A component of the path prefix names an existing file that is neither
 6183 a directory nor a symbolic link to a directory, ...
 6184
 6185 *Rationale:* Austin Group Defect Report(s) applied: 324.
 6186 See <http://austingroupbugs.net/view.php?id=324>
 6187 This is an editorial issue clarifying the intent of the
 6188 standard.
 6189

6190
6191 **Change Number: XSH/TC1/D5/0213** [366]
6192
6193
6194 On Page: 959 Line: 32119-32128 Section: ftok()
6195
6196 In the EXAMPLES section delete L32119-32128 [EXAMPLES] (the body of the
6197 first example, and the title of the second), leaving just the second
6198 example under the title "Getting an IPC Key".
6199
6200 *Rationale:* Austin Group Defect Report(s) applied: 366.
6201 See <http://austingroupbugs.net/view.php?id=366>
6202
6203
6204 **Change Number: XSH/TC1/D5/0214** [366]
6205
6206
6207 On Page: 959 Line: 32129 Section: ftok()
6208
6209 In the EXAMPLES section, on L32129 delete the word "unique".
6210
6211 *Rationale:* Austin Group Defect Report(s) applied: 366.
6212 See <http://austingroupbugs.net/view.php?id=366>
6213
6214
6215 **Change Number: XSH/TC1/D5/0215** [366]
6216
6217
6218 On Page: 959 Line: 32139 Section: ftok()
6219
6220 In the APPLICATION USAGE section add:
6221
6222 Applications should not assume that the resulting key value is unique.
6223
6224 *Rationale:* Austin Group Defect Report(s) applied: 366.
6225 See <http://austingroupbugs.net/view.php?id=366>
6226
6227
6228 **Change Number: XSH/TC1/D5/0216** [366]
6229
6230
6231 On Page: 959 Line: 32143 Section: ftok()
6232
6233 In the FUTURE DIRECTIONS section, change from:
6234
6235 None.
6236
6237 to:
6238
6239 Future versions of this standard may add new interfaces to provide
6240 unique keys.
6241
6242 *Rationale:* Austin Group Defect Report(s) applied: 366.
6243 See <http://austingroupbugs.net/view.php?id=366>
6244 No current implementation is known to provide unique keys.
6245
6246
6247 **Change Number: XSH/TC1/D5/0217** [403]
6248
6249
6250 On Page: 964 Line: 32261 Section: ftw()
6251
6252 In the DESCRIPTION section, change from:

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

6253
6254 FTW_F For a file.
6255
6256 to:
6257
6258 FTW_F For a non-directory file.
6259
6260 *Rationale:* Austin Group Defect Report(s) applied: 403.
6261 See <http://austingroupbugs.net/view.php?id=403>
6262
6263
6264 **Change Number: XSH/TC1/D5/0218 [324]**
6265
6266
6267 On Page: 965 Line: 32299 Section: `ftw()`
6268
6269 In the ERRORS section, for the [ENOTDIR] error, change from:
6270
6271 A component of path is not a directory.
6272
6273 to:
6274
6275 A component of path names an existing file that is neither
6276 a directory nor a symbolic link to a directory.
6277
6278 *Rationale:* Austin Group Defect Report(s) applied: 324.
6279 See <http://austingroupbugs.net/view.php?id=324>
6280 This is an editorial issue clarifying the intent of the
6281 standard.
6282
6283
6284 **Change Number: XSH/TC1/D5/0219 [361]**
6285
6286
6287 On Page: 966 Line: 32351 Section: `ftw()`
6288
6289 In the CHANGE HISTORY section, change "ia" to "is".
6290
6291 *Rationale:* Austin Group Defect Report(s) applied: 361.
6292 See <http://austingroupbugs.net/view.php?id=361>
6293 Editorial correction.
6294
6295
6296 **Change Number: XSH/TC1/D5/0220 [63,428]**
6297
6298
6299 On Page: 969 Line: 32426-32430 Section: `futimens()`
6300
6301 In the ERRORS section, change from:
6302
6303 [EPERM] The times argument is not a null pointer, does not have both
6304 `tv_nsec` fields set to `UTIME_NOW`, does not have both `tv_nsec` fields set
6305 to `UTIME_OMIT`, the calling process' effective user ID has write access
6306 to the file but does not match the owner of the file, and the calling
6307 process does not have appropriate privileges.
6308
6309 to:
6310
6311 [EPERM] The times argument is not a null pointer, does not have both
6312 `tv_nsec` fields set to `UTIME_NOW`, does not have both `tv_nsec` fields set to
6313 `UTIME_OMIT`, the calling process effective user ID does not match the owner
6314 of the file, and the calling process does not have appropriate privileges.
6315

6316 *Rationale:* Austin Group Defect Report(s) applied: 63,428.
6317 See <http://austingroupbugs.net/view.php?id=63>
6318 See <http://austingroupbugs.net/view.php?id=428>
6319
6320
6321 **Change Number: XSH/TC1/D5/0221 [278]**
6322
6323
6324 On Page: 969 Line: 32438 Section: futimens()
6325
6326 In the ERRORS section, add (after the [EBADF] error):
6327
6328 [ENOTDIR] The path argument is not an absolute path and fd is a file
6329 descriptor associated with a non-directory file.
6330
6331 *Rationale:* Austin Group Defect Report(s) applied: 278.
6332 See <http://austingroupbugs.net/view.php?id=278>
6333
6334
6335 **Change Number: XSH/TC1/D5/0222 [324]**
6336
6337
6338 On Page: 970 Line: 32446 Section: futimens()
6339
6340 In the ERRORS section, for the [ENOTDIR] error, change from:
6341
6342 A component of the path prefix is not a directory, ...
6343
6344 to:
6345
6346 A component of the path prefix names an existing file that is neither
6347 a directory nor a symbolic link to a directory, ...
6348
6349 *Rationale:* Austin Group Defect Report(s) applied: 324.
6350 See <http://austingroupbugs.net/view.php?id=324>
6351 This is an editorial issue clarifying the intent of the
6352 standard.
6353
6354
6355 **Change Number: XSH/TC1/D5/0223 [306]**
6356
6357
6358 On Page: 970 Line: 32457 Section: futimens()
6359
6360 After:
6361
6362 The utimensat() function may fail if:
6363
6364 add:
6365
6366 [EINVAL] The value of the flag argument is not valid.
6367
6368 *Rationale:* Austin Group Defect Report(s) applied: 306.
6369 See <http://austingroupbugs.net/view.php?id=306>
6370
6371
6372 **Change Number: XSH/TC1/D5/0224 [278]**
6373
6374
6375 On Page: 970 Line: 32457 Section: futimens()
6376
6377 In the ERRORS section, delete:
6378

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

6379 [ENOTDIR] The path argument is not an absolute path and fd is neither
 6380 AT_FDCWD nor a file descriptor associated with a directory.

6381
 6382 *Rationale:* Austin Group Defect Report(s) applied: 278.
 6383 See <http://austingroupbugs.net/view.php?id=278>

6384
 6385
 6386 **Change Number: XSH/TC1/D5/0225 [272]**

6387
 6388
 6389 On Page: 972 Line: 32521 Section: fwide()
 6390

6391 Insert before L32521 (CX shaded):

6392
 6393 [CX]The fwide() function shall not change the setting of errno if
 6394 successful.[/CX]

6395
 6396 *Rationale:* Austin Group Defect Report(s) applied: 272.
 6397 See <http://austingroupbugs.net/view.php?id=272>

6398
 6399
 6400 **Change Number: XSH/TC1/D5/0226 [302]**

6401
 6402
 6403 On Page: 973 Line: 32590 Section: fwprintf()
 6404

6405 change from:
 6406
 6407 ... the locale of the process ...

6408 to:
 6409
 6410 ... the current locale ...

6411
 6412
 6413 *Rationale:* Austin Group Defect Report(s) applied: 302.
 6414 See <http://austingroupbugs.net/view.php?id=302>
 6415 These changes were overlooked during the revision when
 6416 per-thread locales were added

6417
 6418
 6419 **Change Number: XSH/TC1/D5/0227 [14]**

6420
 6421
 6422 On Page: 979 Line: 32842 Section: fwprintf()
 6423
 6424 In the SEE ALSO section, add a reference to XSH Section 2.5.

6425
 6426 *Rationale:* Austin Group Defect Report(s) applied: 14.
 6427 See <http://austingroupbugs.net/view.php?id=14>
 6428 This is an editorial improvement

6429
 6430
 6431 **Change Number: XSH/TC1/D5/0228 [14]**

6432
 6433
 6434 On Page: 981 Line: 32908 Section: fwrite()
 6435
 6436 In the SEE ALSO section, add a reference to XSH Section 2.5.

6437
 6438 *Rationale:* Austin Group Defect Report(s) applied: 14.
 6439 See <http://austingroupbugs.net/view.php?id=14>
 6440 This is an editorial improvement

6441

6442
6443 **Change Number: XSH/TC1/D5/0229 [302]**
6444
6445
6446 On Page: 983 Line: 32955 Section: fwscanf()
6447
6448 change from:
6449
6450 ... the locale of the process ...
6451
6452 to:
6453
6454 ... the current locale ...
6455
6456 *Rationale:* Austin Group Defect Report(s) applied: 302.
6457 See <http://austingroupbugs.net/view.php?id=302>
6458 These changes were overlooked during the revision when
6459 per-thread locales were added.
6460
6461
6462 **Change Number: XSH/TC1/D5/0230 [14]**
6463
6464
6465 On Page: 989 Line: 33199 Section: fwscanf()
6466
6467 In the SEE ALSO section, add a reference to XSH Section 2.5.
6468
6469 *Rationale:* Austin Group Defect Report(s) applied: 14.
6470 See <http://austingroupbugs.net/view.php?id=14>
6471 This is an editorial improvement
6472
6473
6474 **Change Number: XSH/TC1/D5/0231 [14]**
6475
6476
6477 On Page: 992 Line: 33301 Section: getc()
6478
6479 In the SEE ALSO section, add a reference to XSH Section 2.5.
6480
6481 *Rationale:* Austin Group Defect Report(s) applied: 14.
6482 See <http://austingroupbugs.net/view.php?id=14>
6483 This is an editorial improvement
6484
6485
6486 **Change Number: XSH/TC1/D5/0232 [395]**
6487
6488
6489 On Page: 993 Line: 33321 Section: getc_unlocked()
6490
6491 In the DESCRIPTION section, add a new paragraph at the end of the section:
6492
6493 If `getc_unlocked` or `putc_unlocked` are implemented as macros they may
6494 evaluate stream more than once, so the stream argument should never be
6495 an expression with side-effects.
6496
6497 *Rationale:* Austin Group Defect Report(s) applied: 395.
6498 See <http://austingroupbugs.net/view.php?id=395>
6499
6500
6501 **Change Number: XSH/TC1/D5/0233 [395]**
6502
6503
6504 On Page: 993 Line: 33331 Section: getc_unlocked()

6505
 6506 In the APPLICATION USAGE section, change from:
 6507
 6508 `putc_unlocked(*f++)`
 6509
 6510 to:
 6511
 6512 `putc_unlocked(c,*f++)`
 6513
 6514 *Rationale:* Austin Group Defect Report(s) applied: 395.
 6515 See <http://austingroupbugs.net/view.php?id=395>
 6516
 6517
 6518 **Change Number: XSH/TC1/D5/0234** [395]
 6519
 6520
 6521 On Page: 994 Line: 33375 Section: `getc_unlocked()`
 6522
 6523 In the SEE ALSO section, add a reference to `flockfile()`.
 6524
 6525 *Rationale:* Austin Group Defect Report(s) applied: 395.
 6526 See <http://austingroupbugs.net/view.php?id=395>
 6527
 6528
 6529 **Change Number: XSH/TC1/D5/0235** [14]
 6530
 6531
 6532 On Page: 994 Line: 33375 Section: `getc_unlocked()`
 6533
 6534 In the SEE ALSO section, add a reference to XSH Section 2.5.
 6535
 6536 *Rationale:* Austin Group Defect Report(s) applied: 14.
 6537 See <http://austingroupbugs.net/view.php?id=14>
 6538 This is an editorial improvement
 6539
 6540
 6541 **Change Number: XSH/TC1/D5/0236** [14]
 6542
 6543
 6544 On Page: 995 Line: 33412 Section: `getchar()`
 6545
 6546 In the SEE ALSO section, add a reference to XSH Section 2.5.
 6547
 6548 *Rationale:* Austin Group Defect Report(s) applied: 14.
 6549 See <http://austingroupbugs.net/view.php?id=14>
 6550 This is an editorial improvement
 6551
 6552
 6553 **Change Number: XSH/TC1/D5/0237** [14]
 6554
 6555
 6556 On Page: 1006 Line: 33803 Section: `getdelim()`
 6557
 6558 In the SEE ALSO section, add a reference to XSH Section 2.5.
 6559
 6560 *Rationale:* Austin Group Defect Report(s) applied: 14.
 6561 See <http://austingroupbugs.net/view.php?id=14>
 6562 This is an editorial improvement
 6563
 6564
 6565 **Change Number: XSH/TC1/D5/0238** [75,428]
 6566
 6567

6568 On Page: 1008 Line: 33853 Section: getenv()
6569

6570 In the DESCRIPTION section, change from:
6571

6572 The string pointed to may be overwritten by a subsequent call to
6573 getenv(), [CX]setenv(), unsetenv(),[/CX] [XSI]or putenv()[/XSI] but
6574 shall not be overwritten by a call to any other function in this volume
6575 of POSIX.1-200x.
6576

6577 to:
6578

6579 [CX]The returned string pointer might be invalidated or[/CX] the
6580 string content might be overwritten by a subsequent call to getenv(),
6581 [CX]setenv(), unsetenv(),[/CX] [XSI]or putenv()[/XSI] but they shall
6582 not be affected by a call to any other function in this volume of
6583 POSIX.1-200x.
6584

6585 *Rationale:* Austin Group Defect Report(s) applied: 75,428.

6586 See <http://austingroupbugs.net/view.php?id=75>

6587 See <http://austingroupbugs.net/view.php?id=428>
6588
6589

6590 **Change Number: XSH/TC1/D5/0239** [167]
6591
6592

6593 On Page: 1008 Line: 33856 Section: getenv()
6594

6595 In the DESCRIPTION section, remove the text:
6596

6597 If the application modifies environ or the pointers to which it points,
6598 the behavior of getenv() is undefined.
6599

6600 *Rationale:* Austin Group Defect Report(s) applied: 167.

6601 See <http://austingroupbugs.net/view.php?id=167>

6602 The text relating to undefined behavior on modification of
6603 environ is removed as this is now covered by general text added in XBD
6604 Section 8.1.
6605
6606

6607 **Change Number: XSH/TC1/D5/0240** [167]
6608
6609

6610 On Page: 1008 Line: 33880 Section: getenv()
6611

6612 In the RATIONALE section, change from:
6613

6614 Conforming applications are required not to modify environ directly,
6615 but to use only the functions described here to manipulate the process
6616 environment as an abstract object. Thus, the implementation of the
6617 environment access functions has complete control over the data structure
6618 used to represent the environment (subject to the requirement that environ
6619 be maintained as a list of strings with embedded <equals-sign> characters
6620 for applications that wish to scan the environment). This constraint
6621 allows the implementation to properly manage the memory it allocates,
6622 either by using allocated storage for all variables (copying them on
6623 the first invocation of setenv() or unsetenv()), or keeping track of
6624 which strings are currently in allocated space and which are not, via a
6625 separate table or some other means. This enables the implementation to
6626 free any allocated space used by strings (and perhaps the pointers to
6627 them) stored in environ when unsetenv() is called. A C runtime start-up
6628 procedure (that which invokes main() and perhaps initializes environ)
6629 can also initialize a flag indicating that none of the environment has
6630 yet been copied to allocated storage, or that the separate table has

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

6631 not yet been initialized.

6632

6633 In fact, for higher performance of `getenv()`, the implementation could
 6634 also maintain a separate copy of the environment in a data structure
 6635 that could be searched much more quickly (such as an indexed hash table,
 6636 or a binary tree), and update both it and the linear list at environ
 6637 when `setenv()` or `unsetenv()` is invoked.

6638

6639 to:

6640

6641 Conforming applications are required not to directly modify the pointers
 6642 to which `environ` points, but to use only the `setenv()`, `unsetenv()`,
 6643 and `putenv()` functions, or assignment to `environ` itself, to manipulate
 6644 the process environment. This constraint allows the implementation to
 6645 properly manage the memory it allocates. This enables the implementation
 6646 to free any space it has allocated to strings (and perhaps the pointers to
 6647 them) stored in `environ` when `unsetenv()` is called. A C runtime start-up
 6648 procedure (that which invokes `main()` and perhaps initializes `environ`)
 6649 can also initialize a flag indicating that none of the environment has
 6650 yet been copied to allocated storage, or that the separate table has
 6651 not yet been initialized. If the application switches to a complete new
 6652 environment by assigning a new value to `environ`, this can be detected
 6653 by `getenv()`, `setenv()`, `unsetenv()`, or `putenv()` and the implementation
 6654 can at that point reinitialize based on the new environment. (This may
 6655 include copying the environment strings into a new array and assigning
 6656 `environ` to point to it.)

6657

6658 In fact, for higher performance of `getenv()`, implementations that do not
 6659 provide `putenv()` could also maintain a separate copy of the environment
 6660 in a data structure that could be searched much more quickly (such
 6661 as an indexed hash table, or a binary tree), and update both it and
 6662 the linear list at `environ` when `setenv()` or `unsetenv()` is invoked. On
 6663 implementations that do provide `putenv()`, such a copy might still be
 6664 worthwhile but would need to allow for the fact that applications can
 6665 directly modify the content of environment strings added with `putenv()`.
 6666 For example, if an environment string found by searching the copy is
 6667 one that was added using `putenv()`, the implementation would need to
 6668 check that the string in `environ` still has the same name (and value,
 6669 if the copy includes values), and whenever searching the copy produces
 6670 no match the implementation would then need to search each environment
 6671 string in `environ` that was added using `putenv()` in case any of them have
 6672 changed their names and now match. Thus each use of `putenv()` to add to
 6673 the environment would reduce the speed advantage of having the copy.

6674

6675 *Rationale:* Austin Group Defect Report(s) applied: 167.

6676 See <http://austingroupbugs.net/view.php?id=167>

6677

6678

6679 **Change Number:** XSH/TC1/D5/0241 [75]

6680

6681

6682 On Page: 1014 Line: 34021 Section: `getgrgid()`

6683

6684 In the RETURN VALUE section, change from:

6685

6686 The return value may point to a static area which is overwritten by a
 6687 subsequent call to `getgrent()`, `getgrgid()`, or `getgrnam()`.

6688

6689 to:

6690

6691 The application shall not modify the structure to which the return
 6692 value points, nor any storage areas pointed to by pointers within the
 6693 structure. The returned pointer, and pointers within the structure,

6694 might be invalidated or the structure or the storage areas might be
6695 overwritten by a subsequent call to `getgrent()`, `getgrgid()`, or `getgrnam()`.

6696

6697 *Rationale:* Austin Group Defect Report(s) applied: 75.

6698 See <http://austingroupbugs.net/view.php?id=75>

6699

6700

6701 **Change Number: XSH/TC1/D5/0242 [75]**

6702

6703

6704 On Page: 1018 Line: 34149 Section: `getgrnam()`

6705

6706 In the RETURN VALUE section, change from:

6707

6708 The return value may point to a static area which is overwritten by a
6709 subsequent call to `getgrent()`, `getgrgid()`, or `getgrnam()`.

6710

6711 to:

6712

6713 The application shall not modify the structure to which the return
6714 value points, nor any storage areas pointed to by pointers within the
6715 structure. The returned pointer, and pointers within the structure,
6716 might be invalidated or the structure or the storage areas might be
6717 overwritten by a subsequent call to `getgrent()`, `getgrgid()`, or `getgrnam()`.

6718

6719 *Rationale:* Austin Group Defect Report(s) applied: 75.

6720 See <http://austingroupbugs.net/view.php?id=75>

6721

6722

6723 **Change Number: XSH/TC1/D5/0243 [172]**

6724

6725

6726 On Page: 1029 Line: 34464 Section: `getlogin()`

6727

6728 In the DESCRIPTION section, add a new paragraph to the end of the section:

6729

6730 The `getlogin()` and `getlogin_r()` functions may make use of file descriptors
6731 0, 1, and 2 to find the controlling terminal of the current process,
6732 examining each in turn until the terminal is found. If in this case none
6733 of these three file descriptors is open to the controlling terminal,
6734 these functions may fail. The method used to find the terminal associated
6735 with a file descriptor may depend on the file descriptor being open to
6736 the actual terminal device, not `/dev/tty`.

6737

6738 *Rationale:* Austin Group Defect Report(s) applied: 172.

6739 See <http://austingroupbugs.net/view.php?id=172>

6740

6741

6742 **Change Number: XSH/TC1/D5/0244 [75]**

6743

6744

6745 On Page: 1029 Line: 34469 Section: `getlogin()`

6746

6747 In the RETURN VALUE section, change from:

6748

6749 The return value from `getlogin()` may point to static data whose content
6750 is overwritten by each call.

6751

6752 to:

6753

6754 The application shall not modify the string returned. The returned
6755 pointer might be invalidated or the string content might be overwritten
6756 by a subsequent call to `getlogin()`.

6757

6758 *Rationale:* Austin Group Defect Report(s) applied: 75.6759 See <http://austingroupbugs.net/view.php?id=75>

6760

6761

6762 **Change Number: XSH/TC1/D5/0245** [172]

6763

6764

6765 On Page: 1029 Line: 34476 Section: `getlogin()`

6766

6767 In the ERRORS section, add after L34476 [ENFILE]:

6768

6769 [ENOTTY] None of the file descriptors 0, 1, or 2 is open to the
6770 controlling terminal of the current process.

6771

6772 *Rationale:* Austin Group Defect Report(s) applied: 172.6773 See <http://austingroupbugs.net/view.php?id=172>

6774

6775

6776 **Change Number: XSH/TC1/D5/0246** [284]

6777

6778

6779 On Page: 1036 Line: 34710 Section: `getnameinfo()`

6780

6781

6782

6783 The `sa` argument points to a socket address structure to be translated.

6784

6785 add:

6786

6787 The `salen` argument contains the length of the address pointed to by `sa`.

6788

6789 *Rationale:* Austin Group Defect Report(s) applied: 284.6790 See <http://austingroupbugs.net/view.php?id=284>

6791

6792

6793 **Change Number: XSH/TC1/D5/0247** [285]

6794

6795

6796 On Page: 1036 Line: 34717 Section: `getnameinfo()`

6797

6798 Change "up to nodelen characters" to "up to nodelen bytes".

6799

6800 *Rationale:* Austin Group Defect Report(s) applied: 285.6801 See <http://austingroupbugs.net/view.php?id=285>

6802

6803

6804 **Change Number: XSH/TC1/D5/0248** [318]

6805

6806

6807 On Page: 1040 Line: 34825 Section: `getopt()`

6808

6809 After "... and `getopt()` shall update it when it finishes with each
6810 element of `argv[]`.", add a new sentence:

6811

6812 If the application sets `optind` to zero before calling `getopt()`, the
6813 behavior is unspecified.

6814

6815 *Rationale:* Austin Group Defect Report(s) applied: 318.6816 See <http://austingroupbugs.net/view.php?id=318>

6817

6818

6819 **Change Number: XSH/TC1/D5/0249** [460]

6820
6821
6822 On Page: 1041 Line: 34860 Section: getopt()
6823
6824 In the ERRORS section, change from:
6825
6826 No errors are defined.
6827
6828 to:
6829
6830 If the application has not set the variable opterr to 0, the first
6831 character of optstring is not a <colon>, and a write error occurs
6832 while getopt() is printing a diagnostic message to stderr, then the
6833 error indicator for stderr shall be set; but getopt() shall still
6834 succeed and the value of errno after getopt() is unspecified.
6835
6836 *Rationale:* Austin Group Defect Report(s) applied: 460.
6837 See <http://austingroupbugs.net/view.php?id=460>
6838
6839
6840 **Change Number: XSH/TC1/D5/0250 [189]**
6841
6842
6843 On Page: 1041 Line: 34866 Section: getopt()
6844
6845 In the EXAMPLES section, at L34866 (the start of the example code),
6846 insert:
6847
6848 #include <stdio.h>
6849 #include <stdlib.h>
6850
6851 Change L34871 from:
6852
6853 int bflg, aflg, errflg;
6854
6855 to:
6856
6857 int bflg = 0, aflg = 0, errflg = 0;
6858
6859 Delete L34874 and 34875:
6860
6861 extern char *optarg;
6862 extern int optind, optopt;
6863
6864 Change L34888-34891 from:
6865
6866 else {
6867 bflg++;
6868 bproc();
6869 }
6870
6871 to:
6872
6873 else
6874 bflg++;
6875
6876 *Rationale:* Austin Group Defect Report(s) applied: 189.
6877 See <http://austingroupbugs.net/view.php?id=189>
6878
6879
6880 **Change Number: XSH/TC1/D5/0251 [189]**
6881
6882

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

6883 On Page: 1042 Line: 34899 Section: getopt()
 6884
 6885 In the EXAMPLES section, reduce the indentation of L34899 by four spaces,
 6886 and of L34900-34903 and L34905-34906 by eight spaces, to be consistent
 6887 with the rest of the case statement.
 6888
 6889 Change L34906 from:
 6890
 6891 "Unrecognized option: -%c\n", optopt);
 6892
 6893 to:
 6894
 6895 "Unrecognized option: '-%c'\n", optopt);
 6896
 6897 Delete L34925-34955 (the entire "Checking Options and Arguments" example).
 6898
 6899 Change L34962 from:
 6900
 6901 char *Options = "hdbtl";
 6902
 6903 to:
 6904
 6905 const char *Options = "hdbtl";
 6906
 6907 Change L34964 from:
 6908
 6909 int dbtype, i;
 6910 char c;
 6911
 6912 to:
 6913
 6914 int dbtype, c;
 6915
 6916 *Rationale:* Austin Group Defect Report(s) applied: 189.
 6917 See <http://austingroupbugs.net/view.php?id=189>
 6918
 6919
 6920 **Change Number: XSH/TC1/D5/0252** [189]
 6921
 6922
 6923 On Page: 1044 Line: 34997 Section: getopt()
 6924
 6925 At L34997, insert a cross-reference to XCU getopt().
 6926
 6927 *Rationale:* Austin Group Defect Report(s) applied: 189.
 6928 See <http://austingroupbugs.net/view.php?id=189>
 6929
 6930
 6931 **Change Number: XSH/TC1/D5/0253** [460]
 6932
 6933
 6934 On Page: 1043 Line: 34980 Section: getopt()
 6935
 6936 In the APPLICATION USAGE section, add a new paragraph to the end of
 6937 the section:
 6938
 6939 While `ferror(stderr)` may be used to detect failures to write a diagnostic
 6940 to `stderr` when `getopt()` returns '?', the value of `errno` is unspecified
 6941 in such a condition. Applications desiring more control over handling
 6942 write failures should set `opterr` to 0 and independently perform output
 6943 to `stderr`, rather than relying on `getopt()` to do the output.
 6944
 6945 *Rationale:* Austin Group Defect Report(s) applied: 460.

6946 See <http://austingroupbugs.net/view.php?id=460>
6947
6948
6949 **Change Number: XSH/TC1/D5/0254 [464]**
6950
6951
6952 On Page: 1045 Line: 35015 Section: `getpeername()`
6953
6954 In the DESCRIPTION section, insert the following sentence at the beginning
6955 of the paragraph:
6956
6957 The `address_len` argument points to a `socklen_t` object which on input
6958 specifies the length of the supplied `sockaddr` structure, and on output
6959 specifies the length of the stored address.
6960
6961 *Rationale:* Austin Group Defect Report(s) applied: 464.
6962 See <http://austingroupbugs.net/view.php?id=464>
6963
6964
6965 **Change Number: XSH/TC1/D5/0255 [75,428]**
6966
6967
6968 On Page: 1057 Line: 35319 Section: `getpwnam()`
6969
6970 In the RETURN VALUE section, change from:
6971
6972 The return value may point to a static area which is overwritten by a
6973 subsequent call to `getpwent()`, `getpwnam()`, or `getpwuid()`.
6974
6975 to:
6976
6977 The application shall not modify the structure to which the return
6978 value points, nor any storage areas pointed to by pointers within the
6979 structure. The returned pointer, and pointers within the structure,
6980 might be invalidated or the structure or the storage areas might be
6981 overwritten by a subsequent call to `getpwent()`, `getpwnam()`, or `getpwuid()`.
6982
6983 *Rationale:* Austin Group Defect Report(s) applied: 75,428.
6984 See <http://austingroupbugs.net/view.php?id=75>
6985 See <http://austingroupbugs.net/view.php?id=428>
6986
6987
6988 **Change Number: XSH/TC1/D5/0256 [75]**
6989
6990
6991 On Page: 1061 Line: 35452 Section: `getpwuid()`
6992
6993 In the RETURN VALUE section, change from:
6994
6995 The return value may point to a static area which is overwritten by a
6996 subsequent call to `getpwent()`, `getpwnam()`, or `getpwuid()`.
6997
6998 to:
6999
7000 The application shall not modify the structure to which the return
7001 value points, nor any storage areas pointed to by pointers within the
7002 structure. The returned pointer, and pointers within the structure,
7003 might be invalidated or the structure or the storage areas might be
7004 overwritten by a subsequent call to `getpwent()`, `getpwnam()`, or `getpwuid()`.
7005
7006 *Rationale:* Austin Group Defect Report(s) applied: 75.
7007 See <http://austingroupbugs.net/view.php?id=75>
7008

7009
 7010 **Change Number: XSH/TC1/D5/0257 [14]**
 7011
 7012
 7013 On Page: 1070 Line: 35775 Section: gets()
 7014
 7015 In the SEE ALSO section, add a reference to XSH Section 2.5.
 7016
 7017 *Rationale:* Austin Group Defect Report(s) applied: 14.
 7018 See <http://austingroupbugs.net/view.php?id=14>
 7019 This is an editorial improvement
 7020
 7021
 7022 **Change Number: XSH/TC1/D5/0258 [421]**
 7023
 7024
 7025 On Page: 1073 Line: 35804 Section: getsid()
 7026
 7027 In the RETURN VALUE section, change from:
 7028
 7029 Otherwise, it shall return (pid_t)-1 and ...
 7030
 7031 to:
 7032
 7033 Otherwise, it shall return -1 and ...
 7034
 7035 *Rationale:* Austin Group Defect Report(s) applied: 421.
 7036 See <http://austingroupbugs.net/view.php?id=421>
 7037
 7038
 7039 **Change Number: XSH/TC1/D5/0259 [464]**
 7040
 7041
 7042 On Page: 1074 Line: 35838 Section: getsockname()
 7043
 7044 In the DESCRIPTION section, insert the following sentence at the beginning
 7045 of the paragraph:
 7046
 7047 The address_len argument points to a socklen_t object which on input
 7048 specifies the length of the supplied sockaddr structure, and on output
 7049 specifies the length of the stored address.
 7050
 7051 *Rationale:* Austin Group Defect Report(s) applied: 464.
 7052 See <http://austingroupbugs.net/view.php?id=464>
 7053
 7054
 7055 **Change Number: XSH/TC1/D5/0260 [196]**
 7056
 7057
 7058 On Page: 1078 Line: 35941 Section: getsubopt()
 7059
 7060 In the DESCRIPTION section, add to the end of the paragraph:
 7061
 7062 The getsubopt() function shall not modify the keylistp vector.
 7063
 7064 *Rationale:* Austin Group Defect Report(s) applied: 196.
 7065 See <http://austingroupbugs.net/view.php?id=196>
 7066
 7067
 7068 **Change Number: XSH/TC1/D5/0261 [196]**
 7069
 7070
 7071 On Page: 1079 Line: 35966 Section: getsubopt()

7072
7073 In the EXAMPLES section, after L35966, insert another line:
7074
7075 #include <unistd.h>
7076
7077 *Rationale:* Austin Group Defect Report(s) applied: 196.
7078 See <http://austingroupbugs.net/view.php?id=196>
7079 This is part of consolidation of the example into one code
7080 fragment.
7081
7082
7083 **Change Number: XSH/TC1/D5/0262** [196]
7084
7085
7086 On Page: 1079 Line: 36003 Section: getsubopt()
7087
7088 In the EXAMPLES section, change from:
7089
7090 while (*subopts != '\0')
7091 switch(getsubopt(&subopts, mount_opts, &value))
7092
7093 to:
7094
7095 while (*subopts != '\0')
7096 {
7097 char *saved = subopts;
7098 switch(getsubopt(&subopts, (char **)mount_opts, &value))
7099
7100 *Rationale:* Austin Group Defect Report(s) applied: 196.
7101 See <http://austingroupbugs.net/view.php?id=196>
7102 This is part of consolidation of the example into one code
7103 fragment.
7104
7105
7106 **Change Number: XSH/TC1/D5/0263** [196]
7107
7108
7109 On Page: 1079 Line: 36004-36026 Section: getsubopt()
7110
7111 In the EXAMPLES section, adjust the indentation of L36004-36026 to
7112 reflect the new code block.
7113
7114 *Rationale:* Austin Group Defect Report(s) applied: 196.
7115 See <http://austingroupbugs.net/view.php?id=196>
7116 This is part of consolidation of the example into one code
7117 fragment.
7118
7119
7120 **Change Number: XSH/TC1/D5/0264** [196]
7121
7122
7123 On Page: 1080 Line: 36024 Section: getsubopt()
7124
7125 In the EXAMPLES section, change from:
7126
7127 printf("Unknown suboption '%s'\n", value);
7128 break;
7129
7130 to:
7131
7132 printf("Unknown suboption '%s'\n", saved);
7133 abort();
7134

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

7135 *Rationale:* Austin Group Defect Report(s) applied: 196.
 7136 See <http://austingroupbugs.net/view.php?id=196>
 7137 This is part of consolidation of the example into one code
 7138 fragment.
 7139
 7140
 7141 **Change Number: XSH/TC1/D5/0265 [196]**
 7142
 7143
 7144 On Page: 1080 Line: 36026 Section: `getsubopt()`
 7145
 7146 In the EXAMPLES section, after L36026, insert a new line:
 7147
 7148 }
 7149
 7150 *Rationale:* Austin Group Defect Report(s) applied: 196.
 7151 See <http://austingroupbugs.net/view.php?id=196>
 7152 This is part of consolidation of the example into one code
 7153 fragment.
 7154
 7155
 7156 **Change Number: XSH/TC1/D5/0266 [196]**
 7157
 7158
 7159 On Page: 1080 Line: 36033 Section: `getsubopt()`
 7160
 7161 In the EXAMPLES section, after L36033, insert a new paragraph describing
 7162 the previous example:
 7163
 7164 If the above example is invoked with:
 7165
 7166 `program -o ro,rsiz=512`
 7167
 7168 then after option parsing, the variable `do_all` will be 0, `type` will be a
 7169 null pointer, `read_size` will be 512, `write_size` will be 0, and `read_only`
 7170 will be 1. If it is invoked with:
 7171
 7172 `program -o oops`
 7173
 7174 it will print "Unknown suboption `oops'" before aborting.
 7175
 7176 *Rationale:* Austin Group Defect Report(s) applied: 196.
 7177 See <http://austingroupbugs.net/view.php?id=196>
 7178 This is part of consolidation of the example into one code
 7179 fragment.
 7180
 7181
 7182 **Change Number: XSH/TC1/D5/0267 [196]**
 7183
 7184
 7185 On Page: 1080 Line: 36055 Section: `getsubopt()`
 7186
 7187 In the APPLICATION USAGE section, change from:
 7188
 7189 None.
 7190
 7191 to:
 7192
 7193 The value of `*valuep` when `getsubopt()` returns -1 is unspecified.
 7194 Historical implementations provide various incompatible extensions to
 7195 allow an application to access the suboption text that was not found in
 7196 the `keylistp` array.
 7197

7198 *Rationale:* Austin Group Defect Report(s) applied: 196.
7199 See <http://austingroupbugs.net/view.php?id=196>
7200
7201
7202 **Change Number: XSH/TC1/D5/0268** [196]
7203
7204
7205 On Page: 1080 Line: 36034- Section: `getsubopt()`
7206
7207 In the EXAMPLES section, move L36034-36036 (the subheading Parsing
7208 Suboptions and following paragraph) prior to the code example at P1079
7209 L35965; then delete L36037-36053 (the second code fragment).
7210
7211 *Rationale:* Austin Group Defect Report(s) applied: 196.
7212 See <http://austingroupbugs.net/view.php?id=196>
7213 This is part of consolidation of the example into one code
7214 fragment.
7215
7216
7217 **Change Number: XSH/TC1/D5/0269** [196]
7218
7219
7220 On Page: 1081 Line: 36057 Section: `getsubopt()`
7221
7222 In the RATIONALE section, change from:
7223
7224 None.
7225
7226 to:
7227
7228 The `keylistp` argument of `getsubopt()` is typed as `char * const *` to match
7229 historical practice. However, the standard is clear that implementations
7230 will not modify either the array or the strings contained in the array,
7231 as if the argument had been typed `const char * const *`.
7232
7233 *Rationale:* Austin Group Defect Report(s) applied: 196.
7234 See <http://austingroupbugs.net/view.php?id=196>
7235
7236
7237 **Change Number: XSH/TC1/D5/0270** [14]
7238
7239
7240 On Page: 1085 Line: 36186 Section: `getwc()`
7241
7242 In the SEE ALSO section, add a reference to XSH Section 2.5.
7243
7244 *Rationale:* Austin Group Defect Report(s) applied: 14.
7245 See <http://austingroupbugs.net/view.php?id=14>
7246 This is an editorial improvement
7247
7248
7249 **Change Number: XSH/TC1/D5/0271** [14]
7250
7251
7252 On Page: 1086 Line: 36218 Section: `getwchar()`
7253
7254 In the SEE ALSO section, add a reference to XSH Section 2.5.
7255
7256 *Rationale:* Austin Group Defect Report(s) applied: 14.
7257 See <http://austingroupbugs.net/view.php?id=14>
7258 This is an editorial improvement
7259
7260

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

7261 **Change Number: XSH/TC1/D5/0272** [96]
 7262
 7263
 7264 On Page: 1093 Line: 36478 Section: grantpt()
 7265
 7266 In the RATIONALE section, change from:
 7267
 7268 None.
 7269
 7270 to:
 7271
 7272 See RATIONALE for posix_openpt().
 7273
 7274 On L36482, in the SEE ALSO section, add posix_openpt().
 7275
 7276 *Rationale:* Austin Group Defect Report(s) applied: 96.
 7277 See <http://austingroupbugs.net/view.php?id=96>
 7278 Add reference to posix_openpt() to the RATIONALE and SEE
 7279 ALSO sections.
 7280
 7281
 7282 **Change Number: XSH/TC1/D5/0273** [68]
 7283
 7284
 7285 On Page: 1098 Line: 36646 Section: hypot()
 7286
 7287 In the RETURN VALUE section, change the MX shading to MXX for:
 7288
 7289 If both arguments are subnormal and the correct result is subnormal,
 7290 a range error may occur and the correct result is returned.
 7291
 7292 and change from "is returned" to "shall be returned".
 7293
 7294 *Rationale:* Austin Group Defect Report(s) applied: 68.
 7295 See <http://austingroupbugs.net/view.php?id=68>
 7296
 7297
 7298 **Change Number: XSH/TC1/D5/0274** [302]
 7299
 7300
 7301 On Page: 1135 Line: 37918 Section: isalnum()
 7302
 7303 In the DESCRIPTION section, change from:
 7304
 7305 ... current locale of the process ...
 7306
 7307 to:
 7308
 7309 ... current locale ...
 7310
 7311 *Rationale:* Austin Group Defect Report(s) applied: 302.
 7312 See <http://austingroupbugs.net/view.php?id=302>
 7313 These changes were overlooked during the revision when
 7314 per-thread locales were added.
 7315
 7316
 7317 **Change Number: XSH/TC1/D5/0275** [283]
 7318
 7319
 7320 On Page: 1135 Line: 37922 Section: isalnum()
 7321
 7322 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7323

7324 [CX]The behavior is undefined if the locale argument to `isalnum_l()`
7325 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
7326 object handle. [/CX]
7327
7328 *Rationale:* Austin Group Defect Report(s) applied: 283.
7329 See <http://austingroupbugs.net/view.php?id=283>
7330
7331
7332 **Change Number: XSH/TC1/D5/0276 [283]**
7333
7334
7335 On Page: 1135 Line: 37927 Section: `isalnum()`
7336
7337 In the ERRORS section, change from:
7338
7339 The `isalnum_l()` function may fail if:
7340
7341 [EINVAL] locale is not a valid locale object handle.
7342
7343 to:
7344
7345 No errors are defined.
7346
7347 *Rationale:* Austin Group Defect Report(s) applied: 283.
7348 See <http://austingroupbugs.net/view.php?id=283>
7349
7350
7351 **Change Number: XSH/TC1/D5/0277 [302]**
7352
7353
7354 On Page: 1137 Line: 37961 Section: `isalpha()`
7355
7356 In the DESCRIPTION section, change from:
7357
7358 ... current locale of the process .
7359
7360 to:
7361
7362 ... current locale ...
7363
7364 *Rationale:* Austin Group Defect Report(s) applied: 302.
7365 See <http://austingroupbugs.net/view.php?id=302>
7366 These changes were overlooked during the revision when
7367 per-thread locales were added.
7368
7369
7370 **Change Number: XSH/TC1/D5/0278 [283]**
7371
7372
7373 On Page: 1137 Line: 37965 Section: `isalpha()`
7374
7375 In the DESCRIPTION section, add a new paragraph to the end of the section:
7376
7377 [CX]The behavior is undefined if the locale argument to `isalpha_l()`
7378 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
7379 object handle. [/CX]
7380
7381 *Rationale:* Austin Group Defect Report(s) applied: 283.
7382 See <http://austingroupbugs.net/view.php?id=283>
7383
7384
7385 **Change Number: XSH/TC1/D5/0279 [283]**
7386

7387
7388 On Page: 1137 Line: 37970 Section: isalpha()
7389
7390 In the ERRORS section, change from:
7391
7392 The isalpha_l() function may fail if:
7393
7394 [EINVAL] locale is not a valid locale object handle.
7395
7396 to:
7397
7398 No errors are defined.
7399
7400 *Rationale:* Austin Group Defect Report(s) applied: 283.
7401 See <http://austingroupbugs.net/view.php?id=283>
7402
7403
7404 **Change Number: XSH/TC1/D5/0280** [302]
7405
7406
7407 On Page: 1142 Line: 38097 Section: isblank()
7408
7409 In the DESCRIPTION section, change from:
7410
7411 ... current locale of the process ...
7412
7413 to:
7414
7415 ... current locale ...
7416
7417 *Rationale:* Austin Group Defect Report(s) applied: 302.
7418 See <http://austingroupbugs.net/view.php?id=302>
7419 These changes were overlooked during the revision when
7420 per-thread locales were added.
7421
7422
7423 **Change Number: XSH/TC1/D5/0281** [283]
7424
7425
7426 On Page: 1142 Line: 38101 Section: isblank()
7427
7428 In the DESCRIPTION section, add a new paragraph to the end of the section:
7429
7430 [CX]The behavior is undefined if the locale argument to isblank_l()
7431 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
7432 object handle.[/CX]
7433
7434 *Rationale:* Austin Group Defect Report(s) applied: 283.
7435 See <http://austingroupbugs.net/view.php?id=283>
7436
7437
7438 **Change Number: XSH/TC1/D5/0282** [283]
7439
7440
7441 On Page: 1142 Line: 38106 Section: isblank()
7442
7443 In the ERRORS section, change from:
7444
7445 The isblank_l() function may fail if:
7446
7447 [EINVAL] locale is not a valid locale object handle.
7448
7449 to:

7450

7451 No errors are defined.

7452

7453 *Rationale:* Austin Group Defect Report(s) applied: 283.7454 See <http://austingroupbugs.net/view.php?id=283>

7455

7456

7457 **Change Number: XSH/TC1/D5/0283** [302]

7458

7459

7460 On Page: 1143 Line: 38138 Section: `isctrl()`

7461

7462 In the DESCRIPTION section, change from:

7463

7464 ... current locale of the process ...

7465

7466 to:

7467

7468 ... current locale ...

7469

7470 *Rationale:* Austin Group Defect Report(s) applied: 302.7471 See <http://austingroupbugs.net/view.php?id=302>

7472 These changes were overlooked during the revision when

7473 per-thread locales were added.

7474

7475

7476 **Change Number: XSH/TC1/D5/0284** [283]

7477

7478

7479 On Page: 1143 Line: 38142 Section: `isctrl()`

7480

7481 In the DESCRIPTION section, add a new paragraph to the end of the section:

7482

7483 [CX]The behavior is undefined if the locale argument to `isctrl_l()`7484 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale

7485 object handle.[/CX]

7486

7487 *Rationale:* Austin Group Defect Report(s) applied: 283.7488 See <http://austingroupbugs.net/view.php?id=283>

7489

7490

7491 **Change Number: XSH/TC1/D5/0285** [283]

7492

7493

7494 On Page: 1143 Line: 38147 Section: `isctrl()`

7495

7496 In the ERRORS section, change from:

7497

7498 The `isctrl_l()` function may fail if:

7499

7500 [EINVAL] locale is not a valid locale object handle.

7501

7502 to:

7503

7504 No errors are defined.

7505

7506 *Rationale:* Austin Group Defect Report(s) applied: 283.7507 See <http://austingroupbugs.net/view.php?id=283>

7508

7509

7510 **Change Number: XSH/TC1/D5/0286** [302]

7511

7512

7513 On Page: 1145 Line: 38181 Section: isdigit()
 7514
 7515 In the DESCRIPTION section, change from:
 7516
 7517 ... current locale of the process ...
 7518
 7519 to:
 7520
 7521 ... current locale ...
 7522
 7523 *Rationale:* Austin Group Defect Report(s) applied: 302.
 7524 See <http://austingroupbugs.net/view.php?id=302>
 7525 These changes were overlooked during the revision when
 7526 per-thread locales were added.
 7527
 7528
 7529 **Change Number: XSH/TC1/D5/0287** [283]
 7530
 7531
 7532 On Page: 1145 Line: 38185 Section: isdigit()
 7533
 7534 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7535
 7536 [CX]The behavior is undefined if the locale argument to `isdigit_l()`
 7537 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
 7538 object handle. [/CX]
 7539
 7540 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7541 See <http://austingroupbugs.net/view.php?id=283>
 7542
 7543
 7544 **Change Number: XSH/TC1/D5/0288** [283]
 7545
 7546
 7547 On Page: 1145 Line: 38190 Section: isdigit()
 7548
 7549 In the ERRORS section, change from:
 7550
 7551 The `isdigit_l()` function may fail if:
 7552
 7553 [EINVAL] locale is not a valid locale object handle.
 7554
 7555 to:
 7556
 7557 No errors are defined.
 7558
 7559 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7560 See <http://austingroupbugs.net/view.php?id=283>
 7561
 7562
 7563 **Change Number: XSH/TC1/D5/0289** [302]
 7564
 7565
 7566 On Page: 1148 Line: 38254 Section: isgraph()
 7567
 7568 In the DESCRIPTION section, change from:
 7569
 7570 ... current locale of the process ...
 7571
 7572 to:
 7573
 7574 ... current locale ...
 7575

7576 *Rationale:* Austin Group Defect Report(s) applied: 302.
7577 See <http://austingroupbugs.net/view.php?id=302>
7578 These changes were overlooked during the revision when
7579 per-thread locales were added.

7580

7581

7582 **Change Number: XSH/TC1/D5/0290** [283]

7583

7584

7585 On Page: 1148 Line: 38258 Section: isgraph()

7586

7587 In the DESCRIPTION section, add a new paragraph to the end of the section:

7588

7589 [CX]The behavior is undefined if the locale argument to isgraph_l()
7590 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
7591 object handle.[/CX]

7592

7593 *Rationale:* Austin Group Defect Report(s) applied: 283.
7594 See <http://austingroupbugs.net/view.php?id=283>

7595

7596

7597 **Change Number: XSH/TC1/D5/0291** [283]

7598

7599

7600 On Page: 1148 Line: 38263 Section: isgraph()

7601

7602 In the ERRORS section, change from:

7603

7604 The isgraph_l() function may fail if:

7605

7606 [EINVAL] locale is not a valid locale object handle.

7607

7608 to:

7609

7610 No errors are defined.

7611

7612 *Rationale:* Austin Group Defect Report(s) applied: 283.
7613 See <http://austingroupbugs.net/view.php?id=283>

7614

7615

7616 **Change Number: XSH/TC1/D5/0292** [302]

7617

7618

7619 On Page: 1156 Line: 38515 Section: islower()

7620

7621 In the DESCRIPTION section, change from:

7622

7623 ... current locale of the process ...

7624

7625 to:

7626

7627 ... current locale ...

7628

7629 *Rationale:* Austin Group Defect Report(s) applied: 302.
7630 See <http://austingroupbugs.net/view.php?id=302>

7631

7632 These changes were overlooked during the revision when
7633 per-thread locales were added.

7633

7634

7635 **Change Number: XSH/TC1/D5/0293** [283]

7636

7637

7638 On Page: 1156 Line: 38519 Section: islower()

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

7639
 7640 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7641
 7642 [CX]The behavior is undefined if the locale argument to islower_l()
 7643 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 7644 object handle.[/CX]
 7645
 7646 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7647 See <http://austingroupbugs.net/view.php?id=283>
 7648
 7649
 7650 **Change Number: XSH/TC1/D5/0294 [283]**
 7651
 7652
 7653 On Page: 1156 Line: 38524 Section: islower()
 7654
 7655 In the ERRORS section, change from:
 7656
 7657 The islower_l() function may fail if:
 7658
 7659 [EINVAL] locale is not a valid locale object handle.
 7660
 7661 to:
 7662
 7663 No errors are defined.
 7664
 7665 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7666 See <http://austingroupbugs.net/view.php?id=283>
 7667
 7668
 7669 **Change Number: XSH/TC1/D5/0295 [302]**
 7670
 7671
 7672 On Page: 1156 Line: 38530 Section islower()
 7673
 7674 In the EXAMPLES section, change from:
 7675
 7676 ... the locale of the user .
 7677
 7678 to:
 7679
 7680 ... the current locale..
 7681
 7682 *Rationale:* Austin Group Defect Report(s) applied: 302.
 7683 See <http://austingroupbugs.net/view.php?id=302>
 7684 These changes were overlooked during the revision when
 7685 per-thread locales were added.
 7686
 7687
 7688 **Change Number: XSH/TC1/D5/0296 [304]**
 7689
 7690
 7691 On Page: 1156 Line: 38539,38545,38558,38564 Section: islower()
 7692
 7693 In the EXAMPLES section, on L38539 and L38558 change from:
 7694
 7695 char c;
 7696
 7697 to:
 7698
 7699 unsigned char c;
 7700
 7701 In the EXAMPLES section, on L38545 and L38564 change from:

7702
7703 (char)
7704
7705 to
7706
7707 (unsigned char)
7708
7709 *Rationale:* Austin Group Defect Report(s) applied: 304.
7710 See <http://austingroupbugs.net/view.php?id=304>
7711
7712
7713 **Change Number: XSH/TC1/D5/0297** [302]
7714
7715
7716 On Page: 1160 Line: 38666 Section: `isprint()`
7717
7718 In the DESCRIPTION section, change from:
7719
7720 ... current locale of the process ...
7721
7722 to:
7723
7724 ... current locale ...
7725
7726 *Rationale:* Austin Group Defect Report(s) applied: 302.
7727 See <http://austingroupbugs.net/view.php?id=302>
7728 These changes were overlooked during the revision when
7729 per-thread locales were added.
7730
7731
7732 **Change Number: XSH/TC1/D5/0298** [283]
7733
7734
7735 On Page: 1160 Line: 38670 Section: `isprint()`
7736
7737 In the DESCRIPTION section, add a new paragraph to the end of the section:
7738
7739 [CX]The behavior is undefined if the locale argument to `isprint_l()`
7740 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
7741 object handle. [/CX]
7742
7743 *Rationale:* Austin Group Defect Report(s) applied: 283.
7744 See <http://austingroupbugs.net/view.php?id=283>
7745
7746
7747 **Change Number: XSH/TC1/D5/0299** [283]
7748
7749
7750 On Page: 1160 Line: 38675 Section: `isprint()`
7751
7752 In the ERRORS section, change from:
7753
7754 The `isprint_l()` function may fail if:
7755
7756 [EINVAL] locale is not a valid locale object handle.
7757
7758 to:
7759
7760 No errors are defined.
7761
7762 *Rationale:* Austin Group Defect Report(s) applied: 283.
7763 See <http://austingroupbugs.net/view.php?id=283>
7764

7765
 7766 **Change Number: XSH/TC1/D5/0300 [302]**
 7767
 7768
 7769 On Page: 1162 Line: 38709 Section: ispunct()
 7770
 7771 In the DESCRIPTION section, change from:
 7772
 7773 ... current locale of the process ...
 7774
 7775 to:
 7776
 7777 ... current locale ...
 7778
 7779 *Rationale:* Austin Group Defect Report(s) applied: 302.
 7780 See <http://austingroupbugs.net/view.php?id=302>
 7781 These changes were overlooked during the revision when
 7782 per-thread locales were added.
 7783
 7784
 7785 **Change Number: XSH/TC1/D5/0301 [283]**
 7786
 7787
 7788 On Page: 1162 Line: 38713 Section: ispunct()
 7789
 7790 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7791
 7792 [CX]The behavior is undefined if the locale argument to `ispunct_l()`
 7793 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
 7794 object handle. [/CX]
 7795
 7796 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7797 See <http://austingroupbugs.net/view.php?id=283>
 7798
 7799
 7800 **Change Number: XSH/TC1/D5/0302 [283]**
 7801
 7802
 7803 On Page: 1162 Line: 38718 Section: ispunct()
 7804
 7805 In the ERRORS section, change from:
 7806
 7807 The `ispunct_l()` function may fail if:
 7808
 7809 [EINVAL] locale is not a valid locale object handle.
 7810
 7811 to:
 7812
 7813 No errors are defined.
 7814
 7815 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7816 See <http://austingroupbugs.net/view.php?id=283>
 7817
 7818
 7819 **Change Number: XSH/TC1/D5/0303 [302]**
 7820
 7821
 7822 On Page: 1164 Line: 38752 Section: isspace()
 7823
 7824 In the DESCRIPTION section, change from:
 7825
 7826 ... current locale of the process ...
 7827

7828 to:
7829
7830 ... current locale ...
7831
7832 *Rationale:* Austin Group Defect Report(s) applied: 302.
7833 See <http://austingroupbugs.net/view.php?id=302>
7834 These changes were overlooked during the revision when
7835 per-thread locales were added.
7836
7837
7838 **Change Number: XSH/TC1/D5/0304** [283]
7839
7840
7841 On Page: 1164 Line: 38756 Section: isspace()
7842
7843 In the DESCRIPTION section, add a new paragraph to the end of the section:
7844
7845 [CX]The behavior is undefined if the locale argument to isspace_l()
7846 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
7847 object handle.[/CX]
7848
7849 *Rationale:* Austin Group Defect Report(s) applied: 283.
7850 See <http://austingroupbugs.net/view.php?id=283>
7851
7852
7853 **Change Number: XSH/TC1/D5/0305** [283]
7854
7855
7856 On Page: 1164 Line: 38761 Section: isspace()
7857
7858 In the ERRORS section, change from:
7859
7860 The isspace_l() function may fail if:
7861
7862 [EINVAL] locale is not a valid locale object handle.
7863
7864 to:
7865
7866 No errors are defined.
7867
7868 *Rationale:* Austin Group Defect Report(s) applied: 283.
7869 See <http://austingroupbugs.net/view.php?id=283>
7870
7871
7872 **Change Number: XSH/TC1/D5/0306** [302]
7873
7874
7875 On Page: 1167 Line: 38833 Section: isupper()
7876
7877 In the DESCRIPTION section, change from:
7878
7879 ... current locale of the process ...
7880
7881 to:
7882
7883 ... current locale ...
7884
7885 *Rationale:* Austin Group Defect Report(s) applied: 302.
7886 See <http://austingroupbugs.net/view.php?id=302>
7887 These changes were overlooked during the revision when
7888 per-thread locales were added.
7889
7890

7891 **Change Number: XSH/TC1/D5/0307** [283]
 7892
 7893
 7894 On Page: 1167 Line: 38837 Section: isupper()
 7895
 7896 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7897
 7898 [CX]The behavior is undefined if the locale argument to isupper_l()
 7899 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 7900 object handle. [/CX]
 7901
 7902 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7903 See <http://austingroupbugs.net/view.php?id=283>
 7904
 7905
 7906 **Change Number: XSH/TC1/D5/0308** [283]
 7907
 7908
 7909 On Page: 1167 Line: 38842 Section: isupper()
 7910
 7911 In the ERRORS section, change from:
 7912
 7913 The isupper_l() function may fail if:
 7914
 7915 [EINVAL] locale is not a valid locale object handle.
 7916
 7917 to:
 7918
 7919 No errors are defined.
 7920
 7921 *Rationale:* Austin Group Defect Report(s) applied: 283.
 7922 See <http://austingroupbugs.net/view.php?id=283>
 7923
 7924
 7925 **Change Number: XSH/TC1/D5/0309** [302]
 7926
 7927
 7928 On Page: 1169 Line: 38876 Section: iswalnum()
 7929
 7930 In the DESCRIPTION section, change from:
 7931
 7932 ... current locale of the process ...
 7933
 7934 to:
 7935
 7936 ... current locale ...
 7937
 7938 *Rationale:* Austin Group Defect Report(s) applied: 302.
 7939 See <http://austingroupbugs.net/view.php?id=302>
 7940 These changes were overlooked during the revision when
 7941 per-thread locales were added.
 7942
 7943
 7944 **Change Number: XSH/TC1/D5/0310** [283]
 7945
 7946
 7947 On Page: 1169 Line: 38880 Section: iswalnum()
 7948
 7949 In the DESCRIPTION section, add a new paragraph to the end of the section:
 7950
 7951 [CX]The behavior is undefined if the locale argument to iswalnum_l()
 7952 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 7953 object handle. [/CX]

7954

7955 *Rationale:* Austin Group Defect Report(s) applied: 283.7956 See <http://austingroupbugs.net/view.php?id=283>

7957

7958

7959 **Change Number: XSH/TC1/D5/0311 [283]**

7960

7961

7962 On Page: 1169 Line: 38885 Section: `iswalnum()`

7963

7964 In the ERRORS section, change from:

7965

7966 The `iswalnum_l()` function may fail if:

7967

7968 `[EINVAL]` locale is not a valid locale object handle.

7969

7970 to:

7971

7972 No errors are defined.

7973

7974 *Rationale:* Austin Group Defect Report(s) applied: 283.7975 See <http://austingroupbugs.net/view.php?id=283>

7976

7977

7978 **Change Number: XSH/TC1/D5/0312 [302]**

7979

7980

7981 On Page: 1171 Line: 38924 Section: `iswalpha()`

7982

7983 In the DESCRIPTION section, change from:

7984

7985 ... current locale of the process ...

7986

7987 to:

7988

7989 ... current locale ...

7990

7991 *Rationale:* Austin Group Defect Report(s) applied: 302.7992 See <http://austingroupbugs.net/view.php?id=302>

7993 These changes were overlooked during the revision when

7994 per-thread locales were added.

7995

7996

7997 **Change Number: XSH/TC1/D5/0313 [283]**

7998

7999

8000 On Page: 1171 Line: 38928 Section: `iswalpha()`

8001

8002 In the DESCRIPTION section, add a new paragraph to the end of the section:

8003

8004 `[CX]`The behavior is undefined if the locale argument to `iswalpha_l()`8005 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale8006 object handle.`[/CX]`

8007

8008 *Rationale:* Austin Group Defect Report(s) applied: 283.8009 See <http://austingroupbugs.net/view.php?id=283>

8010

8011

8012 **Change Number: XSH/TC1/D5/0314 [283]**

8013

8014

8015 On Page: 1171 Line: 38933 Section: `iswalpha()`

8016

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

8017 In the ERRORS section, change from:
8018
8019 The iswalpha_l() function may fail if:
8020
8021 [EINVAL] locale is not a valid locale object handle.
8022
8023 to:
8024
8025 No errors are defined.
8026
8027 *Rationale:* Austin Group Defect Report(s) applied: 283.
8028 See <http://austingroupbugs.net/view.php?id=283>
8029
8030
8031 **Change Number: XSH/TC1/D5/0315 [302]**
8032
8033
8034 On Page: 1173 Line: 38972 Section: iswblank()
8035
8036 In the DESCRIPTION section, change from:
8037
8038 ... current locale of the process ...
8039
8040 to:
8041
8042 ... current locale ...
8043
8044 *Rationale:* Austin Group Defect Report(s) applied: 302.
8045 See <http://austingroupbugs.net/view.php?id=302>
8046 These changes were overlooked during the revision when
8047 per-thread locales were added.
8048
8049
8050 **Change Number: XSH/TC1/D5/0316 [283]**
8051
8052
8053 On Page: 1173 Line: 38976 Section: iswblank()
8054
8055 In the DESCRIPTION section, add a new paragraph to the end of the section:
8056
8057 [CX]The behavior is undefined if the locale argument to iswblank_l()
8058 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
8059 object handle.[/CX]
8060
8061 *Rationale:* Austin Group Defect Report(s) applied: 283.
8062 See <http://austingroupbugs.net/view.php?id=283>
8063
8064
8065 **Change Number: XSH/TC1/D5/0317 [283]**
8066
8067
8068 On Page: 1173 Line: 38981 Section: iswblank()
8069
8070 In the ERRORS section, change from:
8071
8072 The iswblank_l() function may fail if:
8073
8074 [EINVAL] locale is not a valid locale object handle.
8075
8076 to:
8077
8078 No errors are defined.
8079

8080 *Rationale:* Austin Group Defect Report(s) applied: 283.
8081 See <http://austingroupbugs.net/view.php?id=283>
8082
8083
8084 **Change Number: XSH/TC1/D5/0318** [302]
8085
8086
8087 On Page: 1174 Line: 39013 Section: `iswcntrl()`
8088
8089 In the DESCRIPTION section, change from:
8090
8091 ... current locale of the process ...
8092
8093 to:
8094
8095 ... current locale ...
8096
8097 *Rationale:* Austin Group Defect Report(s) applied: 302.
8098 See <http://austingroupbugs.net/view.php?id=302>
8099 These changes were overlooked during the revision when
8100 per-thread locales were added.
8101
8102
8103 **Change Number: XSH/TC1/D5/0319** [283]
8104
8105
8106 On Page: 1174 Line: 39017 Section: `iswcntrl()`
8107
8108 In the DESCRIPTION section, add a new paragraph to the end of the section:
8109
8110 [CX]The behavior is undefined if the locale argument to `iswcntrl_l()`
8111 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
8112 object handle. [/CX]
8113
8114 *Rationale:* Austin Group Defect Report(s) applied: 283.
8115 See <http://austingroupbugs.net/view.php?id=283>
8116
8117
8118 **Change Number: XSH/TC1/D5/0320** [283]
8119
8120
8121 On Page: 1174 Line: 39022 Section: `iswcntrl()`
8122
8123 In the ERRORS section, change from:
8124
8125 The `iswcntrl_l()` function may fail if:
8126
8127 [EINVAL] locale is not a valid locale object handle.
8128
8129 to:
8130
8131 No errors are defined.
8132
8133 *Rationale:* Austin Group Defect Report(s) applied: 283.
8134 See <http://austingroupbugs.net/view.php?id=283>
8135
8136
8137 **Change Number: XSH/TC1/D5/0321** [283]
8138
8139
8140 On Page: 1176 Line: 39067 Section: `iswctype()`
8141
8142 In the DESCRIPTION section, add a new paragraph to the end of the section:

8143
 8144 [CX]The behavior is undefined if the locale argument to `iswctype_l()`
 8145 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
 8146 object handle. [/CX]
 8147
 8148 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8149 See <http://austingroupbugs.net/view.php?id=283>
 8150
 8151
 8152 **Change Number: XSH/TC1/D5/0322 [283]**
 8153
 8154
 8155 On Page: 1176 Line: 39072 Section: `iswctype()`
 8156
 8157 In the ERRORS section, change from:
 8158
 8159 The `iswctype_l()` function may fail if:
 8160
 8161 [EINVAL] locale is not a valid locale object handle.
 8162
 8163 to:
 8164
 8165 No errors are defined.
 8166
 8167 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8168 See <http://austingroupbugs.net/view.php?id=283>
 8169
 8170
 8171 **Change Number: XSH/TC1/D5/0323 [302]**
 8172
 8173
 8174 On Page: 1178 Line: 39147 Section: `iswdigit()`
 8175
 8176 In the DESCRIPTION section, change from:
 8177
 8178 ... current locale of the process ...
 8179
 8180 to:
 8181
 8182 ... current locale ...
 8183
 8184 *Rationale:* Austin Group Defect Report(s) applied: 302.
 8185 See <http://austingroupbugs.net/view.php?id=302>
 8186 These changes were overlooked during the revision when
 8187 per-thread locales were added.
 8188
 8189
 8190 **Change Number: XSH/TC1/D5/0324 [283]**
 8191
 8192
 8193 On Page: 1178 Line: 39151 Section: `iswdigit()`
 8194
 8195 In the DESCRIPTION section, add a new paragraph to the end of the section:
 8196
 8197 [CX]The behavior is undefined if the locale argument to `iswdigit_l()`
 8198 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
 8199 object handle. [/CX]
 8200
 8201 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8202 See <http://austingroupbugs.net/view.php?id=283>
 8203
 8204
 8205 **Change Number: XSH/TC1/D5/0325 [283]**

8206
8207
8208 On Page: 1178 Line: 39156 Section: iswdigit()
8209
8210 In the ERRORS section, change from:
8211
8212 The iswdigit_l() function may fail if:
8213
8214 [EINVAL] locale is not a valid locale object handle.
8215
8216 to:
8217
8218 No errors are defined.
8219
8220 *Rationale:* Austin Group Defect Report(s) applied: 283.
8221 See <http://austingroupbugs.net/view.php?id=283>
8222
8223
8224 **Change Number: XSH/TC1/D5/0326** [302]
8225
8226
8227 On Page: 1180 Line: 39195 Section: iswgraph()
8228
8229 In the DESCRIPTION section, change from:
8230
8231 ... current locale of the process ...
8232
8233 to:
8234
8235 ... current locale ...
8236
8237 *Rationale:* Austin Group Defect Report(s) applied: 302.
8238 See <http://austingroupbugs.net/view.php?id=302>
8239 These changes were overlooked during the revision when
8240 per-thread locales were added.
8241
8242
8243 **Change Number: XSH/TC1/D5/0327** [283]
8244
8245
8246 On Page: 1180 Line: 39199 Section: iswgraph()
8247
8248 In the DESCRIPTION section, add a new paragraph to the end of the section:
8249
8250 [CX]The behavior is undefined if the locale argument to iswgraph_l()
8251 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
8252 object handle. [/CX]
8253
8254 *Rationale:* Austin Group Defect Report(s) applied: 283.
8255 See <http://austingroupbugs.net/view.php?id=283>
8256
8257
8258 **Change Number: XSH/TC1/D5/0328** [283]
8259
8260
8261 On Page: 1180 Line: 39204 Section: iswgraph()
8262
8263 In the ERRORS section, change from:
8264
8265 The iswgraph_l() function may fail if:
8266
8267 [EINVAL] locale is not a valid locale object handle.
8268

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

8269 to:
 8270
 8271 No errors are defined.
 8272
 8273 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8274 See <http://austingroupbugs.net/view.php?id=283>
 8275
 8276
 8277 **Change Number: XSH/TC1/D5/0329 [302]**
 8278
 8279
 8280 On Page: 1182 Line: 39243 Section: `iswlower()`
 8281
 8282 In the DESCRIPTION section, change from:
 8283
 8284 ... current locale of the process ...
 8285
 8286 to:
 8287
 8288 ... current locale ...
 8289
 8290 *Rationale:* Austin Group Defect Report(s) applied: 302.
 8291 See <http://austingroupbugs.net/view.php?id=302>
 8292 These changes were overlooked during the revision when
 8293 per-thread locales were added.
 8294
 8295
 8296 **Change Number: XSH/TC1/D5/0330 [283]**
 8297
 8298
 8299 On Page: 1182 Line: 39247 Section: `iswlower()`
 8300
 8301 In the DESCRIPTION section, add a new paragraph to the end of the section:
 8302
 8303 [CX]The behavior is undefined if the locale argument to `iswlower_l()`
 8304 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
 8305 object handle.[/CX]
 8306
 8307 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8308 See <http://austingroupbugs.net/view.php?id=283>
 8309
 8310
 8311 **Change Number: XSH/TC1/D5/0331 [283]**
 8312
 8313
 8314 On Page: 1182 Line: 39252 Section: `iswlower()`
 8315
 8316 In the ERRORS section, change from:
 8317
 8318 The `iswlower_l()` function may fail if:
 8319
 8320 [EINVAL] locale is not a valid locale object handle.
 8321
 8322 to:
 8323
 8324 No errors are defined.
 8325
 8326 *Rationale:* Austin Group Defect Report(s) applied: 283.
 8327 See <http://austingroupbugs.net/view.php?id=283>
 8328
 8329
 8330 **Change Number: XSH/TC1/D5/0332 [302]**
 8331

8332
8333 On Page: 1184 Line: 39291 Section: `iswprint()`
8334
8335 In the DESCRIPTION section, change from:
8336
8337 ... current locale of the process ...
8338
8339 to:
8340
8341 ... current locale ...
8342
8343 *Rationale:* Austin Group Defect Report(s) applied: 302.
8344 See <http://austingroupbugs.net/view.php?id=302>
8345 These changes were overlooked during the revision when
8346 per-thread locales were added.
8347
8348
8349 **Change Number: XSH/TC1/D5/0333 [283]**
8350
8351
8352 On Page: 1184 Line: 39295 Section: `iswprint()`
8353
8354 In the DESCRIPTION section, add a new paragraph to the end of the section:
8355
8356 [CX]The behavior is undefined if the locale argument to `iswprint_l()`
8357 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
8358 object handle. [/CX]
8359
8360 *Rationale:* Austin Group Defect Report(s) applied: 283.
8361 See <http://austingroupbugs.net/view.php?id=283>
8362
8363
8364 **Change Number: XSH/TC1/D5/0334 [283]**
8365
8366
8367 On Page: 1184 Line: 39300 Section: `iswprint()`
8368
8369 In the ERRORS section, change from:
8370
8371 The `iswprint_l()` function may fail if:
8372
8373 [EINVAL] locale is not a valid locale object handle.
8374
8375 to:
8376
8377 No errors are defined.
8378
8379 *Rationale:* Austin Group Defect Report(s) applied: 283.
8380 See <http://austingroupbugs.net/view.php?id=283>
8381
8382
8383 **Change Number: XSH/TC1/D5/0335 [302]**
8384
8385
8386 On Page: 1186 Line: 39339 Section: `iswpunct()`
8387
8388 In the DESCRIPTION section, change from:
8389
8390 ... current locale of the process ...
8391
8392 to:
8393
8394 ... current locale ...

8395
8396 *Rationale:* Austin Group Defect Report(s) applied: 302.
8397 See <http://austingroupbugs.net/view.php?id=302>
8398 These changes were overlooked during the revision when
8399 per-thread locales were added.
8400
8401
8402 **Change Number: XSH/TC1/D5/0336** [283]
8403
8404
8405 On Page: 1186 Line: 39343 Section: `iswpunct()`
8406
8407 In the DESCRIPTION section, add a new paragraph to the end of the section:
8408
8409 [CX]The behavior is undefined if the locale argument to `iswpunct_l()`
8410 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
8411 object handle.[/CX]
8412
8413 *Rationale:* Austin Group Defect Report(s) applied: 283.
8414 See <http://austingroupbugs.net/view.php?id=283>
8415
8416
8417 **Change Number: XSH/TC1/D5/0337** [283]
8418
8419
8420 On Page: 1186 Line: 39348 Section: `iswpunct()`
8421
8422 In the ERRORS section, change from:
8423
8424 The `iswpunct_l()` function may fail if:
8425
8426 [EINVAL] locale is not a valid locale object handle.
8427
8428 to:
8429
8430 No errors are defined.
8431
8432 *Rationale:* Austin Group Defect Report(s) applied: 283.
8433 See <http://austingroupbugs.net/view.php?id=283>
8434
8435
8436 **Change Number: XSH/TC1/D5/0338** [302]
8437
8438
8439 On Page: 1188 Line: 39387 Section: `iswspace()`
8440
8441 In the DESCRIPTION section, change from:
8442
8443 ... current locale of the process ...
8444
8445 to:
8446
8447 ... current locale ...
8448
8449 *Rationale:* Austin Group Defect Report(s) applied: 302.
8450 See <http://austingroupbugs.net/view.php?id=302>
8451 These changes were overlooked during the revision when
8452 per-thread locales were added.
8453
8454
8455 **Change Number: XSH/TC1/D5/0339** [283]
8456
8457

8458 On Page: 1188 Line: 39391 Section: iswspace()
8459
8460 In the DESCRIPTION section, add a new paragraph to the end of the section:
8461
8462 [CX]The behavior is undefined if the locale argument to iswspace_l()
8463 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
8464 object handle. [/CX]
8465
8466 *Rationale:* Austin Group Defect Report(s) applied: 283.
8467 See <http://austingroupbugs.net/view.php?id=283>
8468
8469
8470 **Change Number: XSH/TC1/D5/0340** [283]
8471
8472
8473 On Page: 1188 Line: 39396 Section: iswspace()
8474
8475 In the ERRORS section, change from:
8476
8477 The iswspace_l() function may fail if:
8478
8479 [EINVAL] locale is not a valid locale object handle.
8480
8481 to:
8482
8483 No errors are defined.
8484
8485 *Rationale:* Austin Group Defect Report(s) applied: 283.
8486 See <http://austingroupbugs.net/view.php?id=283>
8487
8488
8489 **Change Number: XSH/TC1/D5/0341** [302]
8490
8491
8492 On Page: 1190 Line: 39435 Section: iswupper()
8493
8494 In the DESCRIPTION section, change from:
8495
8496 ... current locale of the process ...
8497
8498 to:
8499
8500 ... current locale ...
8501
8502 *Rationale:* Austin Group Defect Report(s) applied: 302.
8503 See <http://austingroupbugs.net/view.php?id=302>
8504 These changes were overlooked during the revision when
8505 per-thread locales were added.
8506
8507
8508 **Change Number: XSH/TC1/D5/0342** [283]
8509
8510
8511 On Page: 1190 Line: 39439 Section: iswupper()
8512
8513 In the DESCRIPTION section, add a new paragraph to the end of the section:
8514
8515 [CX]The behavior is undefined if the locale argument to iswupper_l()
8516 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
8517 object handle. [/CX]
8518
8519 *Rationale:* Austin Group Defect Report(s) applied: 283.
8520 See <http://austingroupbugs.net/view.php?id=283>

8521

8522

8523 **Change Number: XSH/TC1/D5/0343** [283]

8524

8525

8526 On Page: 1190 Line: 39444 Section: iswupper()

8527

8528 In the ERRORS section, change from:

8529

8530 The iswupper_l() function may fail if:

8531

8532 [EINVAL] locale is not a valid locale object handle.

8533

8534 to:

8535

8536 No errors are defined.

8537

8538 *Rationale:* Austin Group Defect Report(s) applied: 283.8539 See <http://austingroupbugs.net/view.php?id=283>

8540

8541

8542 **Change Number: XSH/TC1/D5/0344** [302]

8543

8544

8545 On Page: 1192 Line: 39483 Section: iswxdigit()

8546

8547 In the DESCRIPTION section, change from:

8548

8549 ... current locale of the process ...

8550

8551 to:

8552

8553 ... current locale ...

8554

8555 *Rationale:* Austin Group Defect Report(s) applied: 302.8556 See <http://austingroupbugs.net/view.php?id=302>

8557 These changes were overlooked during the revision when

8558 per-thread locales were added

8559

8560

8561 **Change Number: XSH/TC1/D5/0345** [283]

8562

8563

8564 On Page: 1192 Line: 39487 Section: iswxdigit()

8565

8566 In the DESCRIPTION section, add a new paragraph to the end of the section:

8567

8568 [CX]The behavior is undefined if the locale argument to iswxdigit_l()

8569 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale

8570 object handle.[/CX]

8571

8572 *Rationale:* Austin Group Defect Report(s) applied: 283.8573 See <http://austingroupbugs.net/view.php?id=283>

8574

8575

8576 **Change Number: XSH/TC1/D5/0346** [283]

8577

8578

8579 On Page: 1192 Line: 39492 Section: iswxdigit()

8580

8581 In the ERRORS section, change from:

8582

8583 The iswxdigit_l() function may fail if:

8584
8585 [EINVAL] locale is not a valid locale object handle.
8586
8587 to:
8588
8589 No errors are defined.
8590
8591 *Rationale:* Austin Group Defect Report(s) applied: 283.
8592 See <http://austingroupbugs.net/view.php?id=283>
8593
8594
8595 **Change Number: XSH/TC1/D5/0347 [302]**
8596
8597
8598 On Page: 1194 Line: 39531 Section: isxdigit()
8599
8600 In the DESCRIPTION section, change from:
8601
8602 ... current locale of the process ...
8603
8604 to:
8605
8606 ... current locale ...
8607
8608 *Rationale:* Austin Group Defect Report(s) applied: 302.
8609 See <http://austingroupbugs.net/view.php?id=302>
8610 These changes were overlooked during the revision when
8611 per-thread locales were added.
8612
8613
8614 **Change Number: XSH/TC1/D5/0348 [283]**
8615
8616
8617 On Page: 1194 Line: 39535 Section: isxdigit()
8618
8619 In the DESCRIPTION section, add a new paragraph to the end of the section:
8620
8621 [CX]The behavior is undefined if the locale argument to isxdigit_l()
8622 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
8623 object handle. [/CX]
8624
8625 *Rationale:* Austin Group Defect Report(s) applied: 283.
8626 See <http://austingroupbugs.net/view.php?id=283>
8627
8628
8629 **Change Number: XSH/TC1/D5/0349 [283]**
8630
8631
8632 On Page: 1194 Line: 39540 Section: isxdigit()
8633
8634 In the ERRORS section, change from:
8635
8636 The isxdigit_l() function may fail if:
8637
8638 [EINVAL] locale is not a valid locale object handle.
8639
8640 to:
8641
8642 No errors are defined.
8643
8644 *Rationale:* Austin Group Defect Report(s) applied: 283.
8645 See <http://austingroupbugs.net/view.php?id=283>
8646

8647

8648 **Change Number: XSH/TC1/D5/0350 [68]**

8649

8650

8651 On Page: 1196 Line: 39582 Section: j0()

8652

8653 In the RETURN VALUE section, add MXX shading to:

8654

8655 If x is NaN, a NaN shall be returned.

8656

8657 *Rationale:* Austin Group Defect Report(s) applied: 68.8658 See <http://austingroupbugs.net/view.php?id=68>

8659

8660

8661 **Change Number: XSH/TC1/D5/0351 [324]**

8662

8663

8664 On Page: 1206 Line: 39847 Section: lchown()

8665

8666 In the ERRORS section, for the [ENOTDIR] error, change from:

8667

8668 A component of the path prefix is not a directory, ...

8669

8670 to:

8671

8672 A component of the path prefix names an existing file that is neither

8673 a directory nor a symbolic link to a directory, ...

8674

8675 *Rationale:* Austin Group Defect Report(s) applied: 324.8676 See <http://austingroupbugs.net/view.php?id=324>

8677 This is an editorial issue clarifying the intent of the

8678 standard.

8679

8680

8681 **Change Number: XSH/TC1/D5/0352 [68]**

8682

8683

8684 On Page: 1210 Line: 39934 Section: ldexp()

8685

8686 In the RETURN VALUE section, change from:

8687

8688 If the correct value would cause underflow, and is not representable,

8689 a range error may occur, and [MX]either 0.0 (if supported), or [MX]

8690 an implementation-defined value shall be returned.

8691

8692 to:

8693

8694 If the correct value would cause underflow, [MXX]and is not

8695 representable[MXX], a range error may occur, and ldexp(), ldexpf(), and

8696 ldexpl() shall return [MXX]0.0, or [MXX] (if IEC 60559 Floating-Point is

8697 not supported) an implementation-defined value no greater in magnitude

8698 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.

8699

8700 *Rationale:* Austin Group Defect Report(s) applied: 68.8701 See <http://austingroupbugs.net/view.php?id=68>

8702

8703

8704 **Change Number: XSH/TC1/D5/0353 [68]**

8705

8706

8707 On Page: 1210 Line: 39939 Section: ldexp()

8708

8709 In the RETURN VALUE section, change the MX shading to MXX for:

8710
8711 If the correct value would cause underflow, and is representable,
8712 a range error may occur and the correct value shall be returned.
8713
8714 *Rationale:* Austin Group Defect Report(s) applied: 68.
8715 See <http://austingroupbugs.net/view.php?id=68>
8716
8717
8718 **Change Number: XSH/TC1/D5/0354 [326]**
8719
8720
8721 On Page: 1216 Line: 40123 Section: link()
8722
8723 Change:
8724
8725 The linkat() function shall be equivalent to the link() function except
8726 in the case where either path1 or path2 or both are relative paths.
8727
8728 to:
8729
8730 The linkat() function shall be equivalent to the link() function except
8731 that symbolic links shall be handled as specified by the value of flag
8732 (see below) and except in the case where either path1 or path2 or both
8733 are relative paths.
8734
8735 *Rationale:* Austin Group Defect Report(s) applied: 326.
8736 See <http://austingroupbugs.net/view.php?id=326>
8737
8738
8739 **Change Number: XSH/TC1/D5/0355 [461]**
8740
8741
8742 On Page: 1216 Line: 40134 Section: link()
8743
8744 In the DESCRIPTION section, change from:
8745
8746 ... the current working directory is used ...
8747
8748 to:
8749
8750 ... the current working directory shall be used ...
8751
8752 *Rationale:* Austin Group Defect Report(s) applied: 461.
8753 See <http://austingroupbugs.net/view.php?id=461>
8754
8755
8756 **Change Number: XSH/TC1/D5/0356 [326]**
8757
8758
8759 On Page: 1216 Line: 40136 Section link()
8760
8761 Change:
8762
8763 ... the behavior shall be identical to a call to link().
8764
8765 to:
8766
8767 ... the behavior shall be identical to a call to link(),
8768 except that symbolic links shall be handled as specified
8769 by the value of flag.
8770
8771 *Rationale:* Austin Group Defect Report(s) applied: 326.
8772 See <http://austingroupbugs.net/view.php?id=326>

8773
 8774
 8775 **Change Number: XSH/TC1/D5/0357** [324]
 8776
 8777
 8778 On Page: 1217 Line: 40158 Section: link()
 8779
 8780 In the ERRORS section, for the [ENOTDIR] error, change from:
 8781
 8782 A component of either path prefix is not a directory, ...
 8783
 8784 to:
 8785
 8786 A component of either path prefix names an existing file that is neither
 8787 a directory nor a symbolic link to a directory, ...
 8788
 8789 *Rationale:* Austin Group Defect Report(s) applied: 324.
 8790 See <http://austingroupbugs.net/view.php?id=324>
 8791 This is an editorial issue clarifying the intent of the
 8792 standard.
 8793
 8794
 8795 **Change Number: XSH/TC1/D5/0358** [147,429]
 8796
 8797
 8798 On Page: 1217 Line: 40161 Section: link()
 8799
 8800 In the ERRORS section, add to the end of the [ENOTDIR] error (replacing
 8801 the full stop):
 8802
 8803 ..., or the path1 argument names an existing non-directory file and
 8804 the path2 argument names a nonexistent file, contains at least one
 8805 non-`<slash>` character, and ends with one or more trailing `<slash>`
 8806 characters.
 8807
 8808 *Rationale:* Austin Group Defect Report(s) applied: 147,429.
 8809 See <http://austingroupbugs.net/view.php?id=147>
 8810 See <http://austingroupbugs.net/view.php?id=429>
 8811
 8812
 8813 **Change Number: XSH/TC1/D5/0359** [277]
 8814
 8815
 8816 On Page: 1217 Line: 40172 Section: link()
 8817
 8818 In the ERRORS section, change from:
 8819
 8820 ... nor a valid file descriptor open for reading.
 8821
 8822 to:
 8823
 8824 ... nor a valid file descriptor open for reading or searching.
 8825
 8826 *Rationale:* Austin Group Defect Report(s) applied: 277.
 8827 See <http://austingroupbugs.net/view.php?id=277>
 8828
 8829
 8830 **Change Number: XSH/TC1/D5/0360** [278]
 8831
 8832
 8833 On Page: 1217 Line: 40172 Section: link()
 8834
 8835 In the ERRORS section, add (after the [EBADF] error):

8836

8837 [ENOTDIR] The path1 or path2 argument is not an absolute path and fd1 or
8838 fd2, respectively, is a file descriptor associated with a non-directory
8839 file.

8840

8841 *Rationale:* Austin Group Defect Report(s) applied: 278.

8842 See <http://austingroupbugs.net/view.php?id=278>

8843

8844

8845 **Change Number: XSH/TC1/D5/0361 [278]**

8846

8847

8848 On Page: 1218 Line: 40182 Section: link()

8849

8850 In the ERRORS section, delete:

8851

8852 [ENOTDIR] The path1 or path2 argument is not an absolute path and fd1 or
8853 fd2, respectively, is neither AT_FDCWD nor a file descriptor associated
8854 with a directory.

8855

8856 *Rationale:* Austin Group Defect Report(s) applied: 278.

8857 See <http://austingroupbugs.net/view.php?id=278>

8858

8859

8860 **Change Number: XSH/TC1/D5/0362 [75]**

8861

8862

8863 On Page: 1235 Line: 40719 Section: localeconv()

8864

8865 In the RETURN VALUE section, change from:

8866

8867 The application shall not modify the structure pointed to by the return
8868 value which may be overwritten by a subsequent call to localeconv(). In
8869 addition, calls to setlocale() with the categories LC_ALL, LC_MONETARY,
8870 or LC_NUMERIC or calls to uselocale() which change the categories
8871 LC_MONETARY or LC_NUMERIC may overwrite the contents of the structure.

8872

8873 to:

8874

8875 The application shall not modify the structure to which the return
8876 value points, [CX]nor any storage areas pointed to by pointers within
8877 the structure. The returned pointer, and pointers within the structure,
8878 might be invalidated or [CX] the structure [CX]or the storage areas [CX]
8879 might be overwritten by a subsequent call to localeconv(). In addition,
8880 [CX]the returned pointer, and pointers within the structure, might be
8881 invalidated or [CX] the structure [CX]or the storage areas [CX] might
8882 be overwritten by subsequent calls to setlocale() with the categories
8883 LC_ALL, LC_MONETARY, or LC_NUMERIC, [CX]or by calls to uselocale()
8884 which change the categories LC_MONETARY or LC_NUMERIC [CX].

8885

8886 *Rationale:* Austin Group Defect Report(s) applied: 75.

8887 See <http://austingroupbugs.net/view.php?id=75>

8888

8889

8890 **Change Number: XSH/TC1/D5/0363 [291]**

8891

8892

8893 On Page: 1239 Line: 40849-40860 Section: localtime()

8894

8895 In the EXAMPLES section, change all instances of "filename" to "pathname".

8896

8897 *Rationale:* Austin Group Defect Report(s) applied: 291.

8898 See <http://austingroupbugs.net/view.php?id=291>

8899

8900

8901 **Change Number: XSH/TC1/D5/0364 [68]**

8902

8903

8904 On Page: 1249 Line: 41185 Section: `loglp()`

8905

8906 In the RETURN VALUE section, change from:

8907

8908 [MX]If `x` is subnormal, a range error may occur and `x` should be
8909 returned. [/MX]

8910

8911 to:

8912

8913 [MX]If `x` is subnormal, a range error may occur [/MX] [MXX] and `x` should
8914 be returned. [/MXX]

8915

8916 [MX]If `x` is not returned, `loglp()`, `loglpf()`, and `loglpl()` shall return
8917 an implementation-defined value no greater in magnitude than `DBL_MIN`,
8918 `FLT_MIN`, and `LDBL_MIN`, respectively. [/MX]

8919

8920 *Rationale:* Austin Group Defect Report(s) applied: 68.8921 See <http://austingroupbugs.net/view.php?id=68>

8922

8923

8924 **Change Number: XSH/TC1/D5/0365 [394]**

8925

8926

8927 On Page: 1256 Line: 41363 Section: `longjmp()`

8928

8929 In the DESCRIPTION section, change from:

8930

8931 The `longjmp()` function shall restore the environment saved by the most
8932 recent invocation of `setjmp()` in the same thread, with the corresponding
8933 `jmp_buf` argument. If there is no such invocation, or if the function
8934 containing the invocation of `setjmp()` has terminated execution in the
8935 interim, or if the invocation of `setjmp()` was within the scope of an
8936 identifier with variably modified type and execution has left that scope
8937 in the interim, the behavior is undefined

8938

8939 to:

8940

8941 The `longjmp()` function shall restore the environment saved by the most
8942 recent invocation of `setjmp()` in the same process, with the corresponding
8943 `jmp_buf` argument. If the most recent invocation of `setjmp()` with the
8944 corresponding `jmp_buf` occurred in another thread, or if there is no such
8945 invocation, or if the function containing the invocation of `setjmp()` has
8946 terminated execution in the interim, or if the invocation of `setjmp()`
8947 was within the scope of an identifier with variably modified type and
8948 execution has left that scope in the interim, the behavior is undefined

8949

8950 *Rationale:* Austin Group Defect Report(s) applied: 394.8951 See <http://austingroupbugs.net/view.php?id=394>

8952

8953

8954 **Change Number: XSH/TC1/D5/0366 [421]**

8955

8956

8957 On Page: 1265 Line: 41639 Section: `lseek()`

8958

8959 In the RETURN VALUE section, change from:

8960

8961 Otherwise, `(off_t)-1` shall be returned, ...

8962

8963 to:

8964

8965 Otherwise, -1 shall be returned, ...

8966

8967 *Rationale:* Austin Group Defect Report(s) applied: 421.8968 See <http://austingroupbugs.net/view.php?id=421>

8969

8970

8971 **Change Number: XSH/TC1/D5/0367** [109]

8972

8973

8974 On Page: 1270 Line: 41765 Section: mblen()

8975

8976 In the DESCRIPTION section, add a new paragraph to the end of the section

8977 with CX shading:

8978

8979 The mblen() function need not be thread-safe.

8980

8981 *Rationale:* Austin Group Defect Report(s) applied: 109.8982 See <http://austingroupbugs.net/view.php?id=109>

8983

8984

8985 **Change Number: XSH/TC1/D5/0368** [109,105]

8986

8987

8988 On Page: 1272 Line: 41808 Section: mbrlen()

8989

8990 In the DESCRIPTION section, add two new paragraphs to the end of the

8991 section:

8992

8993 [CX]The mbrlen() function need not be thread-safe if called with a NULL

8994 ps argument.[/CX]

8995

8996 The mbrlen() function shall not change the setting of errno if successful.

8997

8998 *Rationale:* Austin Group Defect Report(s) applied: 109,105.8999 See <http://austingroupbugs.net/view.php?id=109>9000 See <http://austingroupbugs.net/view.php?id=105>

9001

9002

9003 **Change Number: XSH/TC1/D5/0369** [109,105]

9004

9005

9006 On Page: 1274 Line: 41871 Section: mbrtowc()

9007

9008 In the DESCRIPTION section, add two new paragraphs to the end of the

9009 section:

9010

9011 [CX]The mbrtowc() function need not be thread-safe if called with a NULL

9012 ps argument.[/CX]

9013

9014 The mbrtowc() function shall not change the setting of errno if

9015 successful.

9016

9017 *Rationale:* Austin Group Defect Report(s) applied: 109,105.9018 See <http://austingroupbugs.net/view.php?id=109>9019 See <http://austingroupbugs.net/view.php?id=105>

9020

9021

9022 **Change Number: XSH/TC1/D5/0370** [109,105]

9023

9024

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

9025 On Page: 1277 Line: 41989 Section: mbsnrtowcs()
 9026
 9027 In the DESCRIPTION section, add two new paragraphs to the end of the
 9028 section:
 9029
 9030 [CX]The mbsnrtowcs() and mbsrtowcs() functions need not be thread-safe if
 9031 called with a NULL ps argument.[/CX]
 9032
 9033 The mbsrtowcs() function shall not change the setting of errno if
 9034 successful.
 9035
 9036 *Rationale:* Austin Group Defect Report(s) applied: 109,105.
 9037 See <http://austingroupbugs.net/view.php?id=109>
 9038 See <http://austingroupbugs.net/view.php?id=105>
 9039
 9040
 9041 **Change Number: XSH/TC1/D5/0371 [195]**
 9042
 9043
 9044 On Page: 1279 Line: 42044 Section: mbstowcs()
 9045
 9046 In the RETURN VALUE section, change from:
 9047
 9048 ... and may set errno ...
 9049
 9050 to:
 9051
 9052 ... and shall set errno ...
 9053
 9054 *Rationale:* Austin Group Defect Report(s) applied: 195.
 9055 See <http://austingroupbugs.net/view.php?id=195>
 9056
 9057
 9058 **Change Number: XSH/TC1/D5/0372 [109]**
 9059
 9060
 9061 On Page: 1281 Line: 42094 Section: mbtowc()
 9062
 9063 In the DESCRIPTION section, add a new paragraph to the end of the section,
 9064 with CX shading:
 9065
 9066 The mbtowc() function need not be thread-safe.
 9067
 9068 *Rationale:* Austin Group Defect Report(s) applied: 109.
 9069 See <http://austingroupbugs.net/view.php?id=109>
 9070
 9071
 9072 **Change Number: XSH/TC1/D5/0373 [195]**
 9073
 9074
 9075 On Page: 1281 Line: 42099 Section: mbtowc()
 9076
 9077 In the RETURN VALUE section, change from:
 9078
 9079 ... and may set errno ...
 9080
 9081 to:
 9082
 9083 ... and shall set errno ...
 9084
 9085 *Rationale:* Austin Group Defect Report(s) applied: 195.
 9086 See <http://austingroupbugs.net/view.php?id=195>
 9087

9088

9089 **Change Number: XSH/TC1/D5/0374** [110]

9090

9091

9092 On Page: 1284 Line: 42163 Section: memchr()

9093

9094 In the DESCRIPTION section, change from:

9095

9096 The memchr() function shall locate the first occurrence of c (converted
9097 to an unsigned char) in the initial n bytes (each interpreted as unsigned
9098 char) of the object pointed to by s.

9099

9100 to:

9101

9102 The memchr() function shall locate the first occurrence of c (converted
9103 to an unsigned char) in the initial n bytes (each interpreted as unsigned
9104 char) pointed to by s.

9105

9106 Add to DESCRIPTION as a new paragraph:

9107

9108 Implementations shall behave as if they read the memory byte by byte
9109 from the beginning of the bytes pointed to by s and stop at the first
9110 occurrence of c (if it is found in the initial n bytes).

9111

9112 In the RETURN VALUE section change from:

9113

9114 The memchr() function shall return a pointer to the located byte, or a
9115 null pointer if the byte does not occur in the object.

9116

9117 to:

9118

9119 The memchr() function shall return a pointer to the located byte, or a
9120 null pointer if the byte is not found.

9121

9122 *Rationale:* Austin Group Defect Report(s) applied: 110.9123 See <http://austingroupbugs.net/view.php?id=110>

9124

9125

9126 **Change Number: XSH/TC1/D5/0375** [461]

9127

9128

9129 On Page: 1289 Line: 42327 Section: mkdir()

9130

9131 In the DESCRIPTION section, change from:

9132

9133 ... the current working directory is used ...

9134

9135 to:

9136

9137 ... the current working directory shall be used ...

9138

9139 *Rationale:* Austin Group Defect Report(s) applied: 461.9140 See <http://austingroupbugs.net/view.php?id=461>

9141

9142

9143 **Change Number: XSH/TC1/D5/0376** [324]

9144

9145

9146 On Page: 1290 Line: 42346 Section: mkdir()

9147

9148 In the ERRORS section, for the [ENOTDIR] error, change from:

9149

9150 A component of the path prefix is not a directory.

9151
 9152 to:
 9153
 9154 A component of the path prefix names an existing file that is neither
 9155 a directory nor a symbolic link to a directory.
 9156
 9157 *Rationale:* Austin Group Defect Report(s) applied: 324.
 9158 See <http://austingroupbugs.net/view.php?id=324>
 9159 This is an editorial issue clarifying the intent of the
 9160 standard.
 9161
 9162
 9163 **Change Number: XSH/TC1/D5/0377 [277]**
 9164
 9165
 9166 On Page: 1290 Line: 42350 Section: mkdir()
 9167
 9168 In the ERRORS section, [EBADF] error, change from:
 9169
 9170 ... nor a valid file descriptor open for reading.
 9171
 9172 to:
 9173
 9174 ... nor a valid file descriptor open for reading or searching.
 9175
 9176 *Rationale:* Austin Group Defect Report(s) applied: 277.
 9177 See <http://austingroupbugs.net/view.php?id=277>
 9178
 9179
 9180 **Change Number: XSH/TC1/D5/0378 [278]**
 9181
 9182
 9183 On Page: 1290 Line: 42350 mkdir()
 9184
 9185 In the ERRORS section, add (after the [EBADF] error):
 9186
 9187 [ENOTDIR] The path argument is not an absolute path and fd is a file
 9188 descriptor associated with a non-directory file.
 9189
 9190 *Rationale:* Austin Group Defect Report(s) applied: 278.
 9191 See <http://austingroupbugs.net/view.php?id=278>
 9192
 9193
 9194 **Change Number: XSH/TC1/D5/0379 [278]**
 9195
 9196
 9197 On Page: 1290 Line: 42358 Section: mkdir()
 9198
 9199 In the ERRORS section, delete:
 9200
 9201 The mkdirat() function may fail if:
 9202
 9203 [ENOTDIR] The path argument is not an absolute path and fd is neither
 9204 AT_FDCWD nor a file descriptor associated with a directory.
 9205
 9206 *Rationale:* Austin Group Defect Report(s) applied: 278.
 9207 See <http://austingroupbugs.net/view.php?id=278>
 9208
 9209
 9210 **Change Number: XSH/TC1/D5/0380 [291]**
 9211
 9212
 9213 On Page: 1292 Line: 42420- Section: mkdtemp()

9214
9215 In the DESCRIPTION section, on L42420, L42426, L42428, and L42430
9216 change "filename" to "pathname".
9217
9218 *Rationale:* Austin Group Defect Report(s) applied: 291.
9219 See <http://austingroupbugs.net/view.php?id=291>
9220
9221
9222 **Change Number: XSH/TC1/D5/0381 [324]**
9223
9224
9225 On Page: 1293 Line: 42454 Section: mkdtemp()
9226
9227 In the ERRORS section, for the [ENOTDIR] error, change from:
9228
9229 A component of the path prefix is not a directory.
9230
9231 to:
9232
9233 A component of the path prefix names an existing file that is neither
9234 a directory nor a symbolic link to a directory.
9235
9236 *Rationale:* Austin Group Defect Report(s) applied: 324.
9237 See <http://austingroupbugs.net/view.php?id=324>
9238 This is an editorial issue clarifying the intent of the
9239 standard.
9240
9241
9242 **Change Number: XSH/TC1/D5/0382 [291]**
9243
9244
9245 On Page: 1293 Line: 42465 Section: mkdtemp()
9246
9247 In the EXAMPLES section, change "Filename" to "Pathname".
9248
9249 *Rationale:* Austin Group Defect Report(s) applied: 291.
9250 See <http://austingroupbugs.net/view.php?id=291>
9251
9252
9253 **Change Number: XSH/TC1/D5/0383 [461]**
9254
9255
9256 On Page: 1295 Line: 42523 Section: mkfifo()
9257
9258 In the DESCRIPTION section, change from:
9259
9260 ... the current working directory is used ...
9261
9262 to:
9263
9264 ... the current working directory shall be used ...
9265
9266 *Rationale:* Austin Group Defect Report(s) applied: 461.
9267 See <http://austingroupbugs.net/view.php?id=461>
9268
9269
9270 **Change Number: XSH/TC1/D5/0384 [146,435]**
9271
9272
9273 On Page: 1296 Line: 42537 Section: mkfifo()
9274
9275 In the ERRORS section, change from:
9276

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

9277 [ENOENT] A component of the path prefix specified by path does not name
 9278 an existing directory or path is an empty string.
 9279
 9280 to:
 9281
 9282 [ENOENT] A component of the path prefix of path does not name an
 9283 existing file or path is an empty string.
 9284
 9285 [ENOENT] or [ENOTDIR]
 9286 The path argument contains at least one non-`<slash>` character and ends
 9287 with one or more trailing `<slash>` characters. If path names an existing
 9288 file, an [ENOENT] error shall not occur.
 9289
 9290 *Rationale:* Austin Group Defect Report(s) applied: 146,435.
 9291 See <http://austingroupbugs.net/view.php?id=146>
 9292 See <http://austingroupbugs.net/view.php?id=435>
 9293
 9294
 9295 **Change Number: XSH/TC1/D5/0385** [324]
 9296
 9297
 9298 On Page: 1296 Line: 42541 Section: `mkfifo()`
 9299
 9300 In the ERRORS section, for the [ENOTDIR] error, change from:
 9301
 9302 A component of the path prefix is not a directory.
 9303
 9304 to:
 9305
 9306 A component of the path prefix names an existing file that is neither
 9307 a directory nor a symbolic link to a directory.
 9308
 9309 *Rationale:* Austin Group Defect Report(s) applied: 324.
 9310 See <http://austingroupbugs.net/view.php?id=324>
 9311 This is an editorial issue clarifying the intent of the
 9312 standard.
 9313
 9314
 9315 **Change Number: XSH/TC1/D5/0386** [278]
 9316
 9317
 9318 On Page: 1296 Line: 42547 `mkfifo()`
 9319
 9320 In the ERRORS section, add (after the [EBADF] error):
 9321
 9322 [ENOTDIR] The path argument is not an absolute path and fd is a file
 9323 descriptor associated with a non-directory file.
 9324
 9325 *Rationale:* Austin Group Defect Report(s) applied: 278.
 9326 See <http://austingroupbugs.net/view.php?id=278>
 9327
 9328
 9329 **Change Number: XSH/TC1/D5/0387** [278]
 9330
 9331
 9332 On Page: 1296 Line: 42555 Section: `mkfifo()`
 9333
 9334 In the ERRORS section, delete:
 9335
 9336 The `mkfifoat()` function may fail if:
 9337
 9338 [ENOTDIR] The path argument is not an absolute path and fd is neither
 9339 `AT_FDCWD` nor a file descriptor associated with a directory.

9340

9341 *Rationale:* Austin Group Defect Report(s) applied: 278.9342 See <http://austingroupbugs.net/view.php?id=278>

9343

9344

9345 **Change Number: XSH/TC1/D5/0388 [324]**

9346

9347

9348 On Page: 1299 Line: 42693 Section: `mknod()`

9349

9350 In the ERRORS section, for the [ENOTDIR] error, change from:

9351

9352 A component of the path prefix is not a directory.

9353

9354 to:

9355

9356 A component of the path prefix names an existing file that is neither
9357 a directory nor a symbolic link to a directory.

9358

9359 *Rationale:* Austin Group Defect Report(s) applied: 324.9360 See <http://austingroupbugs.net/view.php?id=324>

9361 This is an editorial issue clarifying the intent of the

9362 standard.

9363

9364

9365 **Change Number: XSH/TC1/D5/0389 [461]**

9366

9367

9368 On Page: 1299 Line: 42673 Section: `mknod()`

9369

9370 In the DESCRIPTION section, change from:

9371

9372 ... the current working directory is used ...

9373

9374 to:

9375

9376 ... the current working directory shall be used ...

9377

9378 *Rationale:* Austin Group Defect Report(s) applied: 461.9379 See <http://austingroupbugs.net/view.php?id=461>

9380

9381

9382 **Change Number: XSH/TC1/D5/0390 [146,435]**

9383

9384

9385 On Page: 1299 Line: 42689 Section: `mknod()`

9386

9387 In the ERRORS section, change from:

9388

9389 [ENOENT] A component of the path prefix specified by path does not name
9390 an existing directory or path is an empty string.

9391

9392 to:

9393

9394 [ENOENT] A component of the path prefix of path does not name an
9395 existing file or path is an empty string.

9396

9397 [ENOENT] or [ENOTDIR]

9398 The path argument contains at least one non-`<slash>` character and ends9399 with one or more trailing `<slash>` characters. If path names an existing

9400 file, an [ENOENT] error shall not occur.

9401

9402 *Rationale:* Austin Group Defect Report(s) applied: 146,435.

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

9403 See <http://austingroupbugs.net/view.php?id=146>
 9404 See <http://austingroupbugs.net/view.php?id=435>
 9405
 9406
 9407 **Change Number: XSH/TC1/D5/0391** [278]
 9408
 9409
 9410 On Page: 1300 Line: 42702 `mknod()`
 9411
 9412 In the ERRORS section, add (after the [EBADF] error):
 9413
 9414 [ENOTDIR] The path argument is not an absolute path and `fd` is a file
 9415 descriptor associated with a non-directory file.
 9416
 9417 *Rationale:* Austin Group Defect Report(s) applied: 278.
 9418 See <http://austingroupbugs.net/view.php?id=278>
 9419
 9420
 9421 **Change Number: XSH/TC1/D5/0392** [278]
 9422
 9423
 9424 On Page: 1300 Line: 42710 Section: `mknod()`
 9425
 9426 In the ERRORS section, delete:
 9427
 9428 The `mknodat()` function may fail if:
 9429
 9430 [ENOTDIR] The path argument is not an absolute path and `fd` is neither
 9431 `AT_FDCWD` nor a file descriptor associated with a directory.
 9432
 9433 *Rationale:* Austin Group Defect Report(s) applied: 278.
 9434 See <http://austingroupbugs.net/view.php?id=278>
 9435
 9436
 9437 **Change Number: XSH/TC1/D5/0393** [1040]
 9438
 9439
 9440 On Page: 1303 Line: 42798 Section: `mktime()`
 9441
 9442 In the RETURN VALUE section, change from:
 9443
 9444 ... shall return the value `(time_t)-1` [CX] and may set `errno` to
 9445 indicate the error [CX].
 9446
 9447 to:
 9448
 9449 ... shall return the value `(time_t)-1` [CX] and set `errno` to
 9450 indicate the error [CX].
 9451
 9452 On L42800, in the ERRORS section, change from:
 9453
 9454 The `mktime()` function may fail if ...
 9455
 9456 to:
 9457
 9458 The `mktime()` function shall fail if ...
 9459
 9460 *Rationale:* Austin Group Defect Report(s) applied: 104.
 9461 See <http://austingroupbugs.net/view.php?id=104>
 9462 The [EOVERFLOW] error is changed from a may fail to a shall
 9463 fail.
 9464
 9465

9466 **Change Number: XSH/TC1/D5/0394** [259]

9467

9468

9469 On Page: 1328 Line: 43672 Section: mq_open()

9470

9471 In the DESCRIPTION section (O_CREAT), after:

9472

9473 ... the message queue mq_maxmsg and mq_msgsize attributes shall be set

9474 to the values of the corresponding members in the mq_attr structure

9475 referred to by attr.

9476

9477 Add:

9478

9479 The values of the mq_flags and mq_curmsgs members of the mq_attr structure

9480 shall be ignored.

9481

9482 *Rationale:* Austin Group Defect Report(s) applied: 259.

9483 See <http://austingroupbugs.net/view.php?id=259>

9484

9485

9486 **Change Number: XSH/TC1/D5/0395** [345]

9487

9488

9489 On Page: 1342 Line: 44077 Section: msgctl()

9490

9491 In the DESCRIPTION section (for IPC_SET), add:

9492

9493 Also, the msg_ctime timestamp shall be set to the current time, as

9494 described in XSH Section 2.7.1.

9495

9496 *Rationale:* Austin Group Defect Report(s) applied: 345.

9497 See <http://austingroupbugs.net/view.php?id=345>

9498

9499

9500 **Change Number: XSH/TC1/D5/0396** [345]

9501

9502

9503 On Page: 1344 Line: 44145 Section: msgget()

9504

9505 In the DESCRIPTION section, change from:

9506

9507 ... set equal to the current time

9508

9509 to:

9510

9511 ... set to the current time, as described in XSH Section 2.7.1

9512

9513 *Rationale:* Austin Group Defect Report(s) applied: 345.

9514 See <http://austingroupbugs.net/view.php?id=345>

9515

9516

9517 **Change Number: XSH/TC1/D5/0397** [344]

9518

9519

9520 On Page: 1345 Line: 44174 Section: msgget()

9521

9522 In the SEE ALSO section, add ftok() prior to mq_close().

9523

9524 *Rationale:* Austin Group Defect Report(s) applied: 344.

9525 See <http://austingroupbugs.net/view.php?id=344>

9526

9527

9528 **Change Number: XSH/TC1/D5/0398** [345]

9529
 9530
 9531 On Page: 1347 Line: 44230 Section: msgrcv()
 9532
 9533 In the DESCRIPTION section, change from:
 9534
 9535 ... set equal to the current time
 9536
 9537 to:
 9538
 9539 ... set to the current time, as described in XSH Section 2.7.1
 9540
 9541 *Rationale:* Austin Group Defect Report(s) applied: 345.
 9542 See <http://austingroupbugs.net/view.php?id=345>
 9543
 9544
 9545 **Change Number: XSH/TC1/D5/0399** [421]
 9546
 9547
 9548 On Page: 1347 Line: 44234 Section: msgrcv()
 9549
 9550 In the RETURN VALUE section, change from:
 9551
 9552 ... msgrcv() shall return (ssize_t)-1, and ...
 9553
 9554 to:
 9555
 9556 ... msgrcv() shall return -1, and ...
 9557
 9558 *Rationale:* Austin Group Defect Report(s) applied: 421.
 9559 See <http://austingroupbugs.net/view.php?id=421>
 9560
 9561
 9562 **Change Number: XSH/TC1/D5/0400** [345]
 9563
 9564
 9565 On Page: 1350 Line: 44329 Section: msgsnd()
 9566
 9567 In the DESCRIPTION section, change from:
 9568
 9569 ... set equal to the current time
 9570
 9571 to:
 9572
 9573 ... set to the current time, as described in XSH Section 2.7.1
 9574
 9575 *Rationale:* Austin Group Defect Report(s) applied: 345.
 9576 See <http://austingroupbugs.net/view.php?id=345>
 9577
 9578
 9579 **Change Number: XSH/TC1/D5/0401** [359]
 9580
 9581
 9582 On Page: 1350 Line: 44342 Section: msgsnd() ERRORS
 9583
 9584 Remove the words "less than 0 or" from the description of [EINVAL].
 9585
 9586 *Rationale:* Austin Group Defect Report(s) applied: 359.
 9587 See <http://austingroupbugs.net/view.php?id=359>
 9588
 9589
 9590 **Change Number: XSH/TC1/D5/0402** [346,428]
 9591

9592
9593 On Page: 1362 Line: 44682 Section: nearbyint()
9594
9595 For nearbyint(), delete P1362 L44682-44685 (DESCRIPTION), L44691-44693
9596 (RETURN VALUE), and replace L44695-44700 (ERRORS) with:
9597
9598 No errors are defined.
9599
9600 In the RETURN VALUE section, at L44687 add:
9601
9602 [MX]The result shall have the same sign as x.[/MX]
9603
9604 In the APPLICATION USAGE section, change from:
9605
9606 On error, the expressions (math_errhandling & MATH_ERRNO) and
9607 (math_errhandling & MATH_ERREXCEPT) are independent of each other,
9608 but at least one of them must be non-zero.
9609
9610 to:
9611
9612 The integral value returned by these functions need not be expressible
9613 as an intmax_t. The return value should be tested before assigning it
9614 to an integer type to avoid the undefined results of an integer overflow.
9615
9616 *Rationale:* Austin Group Defect Report(s) applied: 346,428.
9617 See <http://austingroupbugs.net/view.php?id=346>
9618 See <http://austingroupbugs.net/view.php?id=428>
9619
9620
9621 **Change Number: XSH/TC1/D5/0403 [227]**
9622
9623
9624 On Page: 1364 Line: 44724 Section: newlocale()
9625
9626 In the DESCRIPTION section, change from:
9627
9628 It is unspecified whether the locale object pointed to by base shall be
9629 modified or freed and a new locale object created.
9630
9631 to:
9632
9633 It is unspecified whether the locale object pointed to by base shall be
9634 modified, or freed and a new locale object created.
9635
9636 *Rationale:* Austin Group Defect Report(s) applied: 227.
9637 See <http://austingroupbugs.net/view.php?id=227>
9638
9639
9640 **Change Number: XSH/TC1/D5/0404 [283]**
9641
9642
9643 On Page: 1364 Line: 44746 Section: newlocale()
9644
9645 In the DESCRIPTION section, change from:
9646
9647 The results are undefined if the base argument is the special locale
9648 object LC_GLOBAL_LOCALE.
9649
9650 to:
9651
9652 The behavior is undefined if the base argument is the special locale
9653 object LC_GLOBAL_LOCALE, or is not a valid locale object handle and is
9654 not (locale_t)0.

9655

9656 *Rationale:* Austin Group Defect Report(s) applied: 283.9657 See <http://austingroupbugs.net/view.php?id=283>

9658

9659

9660 **Change Number: XSH/TC1/D5/0405 [295]**

9661

9662

9663 On Page: 1365 Line: 44768, 44787 Section: newlocale()

9664

9665 In the EXAMPLES section, change both occurrences of "NULL" to

9666 "(locale_t)0" (without the quotes).

9667

9668 *Rationale:* Austin Group Defect Report(s) applied: 295.9669 See <http://austingroupbugs.net/view.php?id=295>

9670

9671

9672 **Change Number: XSH/TC1/D5/0406 [227]**

9673

9674

9675 On Page: 1366 Line: 44793 Section: newlocale()

9676

9677 In the APPLICATION USAGE section, change from:

9678

9679 Handles for locale objects created by the newlocale() function should

9680 be released by a corresponding call to freelocale().

9681

9682 to:

9683

9684 Handles for locale objects returned by the newlocale() function should

9685 either be released by a corresponding call to freelocale(), or be used

9686 as a base locale to another newlocale() call.

9687

9688 *Rationale:* Austin Group Defect Report(s) applied: 227.9689 See <http://austingroupbugs.net/view.php?id=227>

9690

9691

9692 **Change Number: XSH/TC1/D5/0407 [68]**

9693

9694

9695 On Page: 1367 Line: 44840 Section: nextafter()

9696

9697 In the RETURN VALUE section, change from:

9698

9699 [MX]If x!=y and the correct function value is subnormal, zero, or

9700 underflows, a range error shall occur, and either the correct function

9701 value (if representable) or 0.0 shall be returned.[/MX]

9702

9703 to:

9704

9705 [MX]If x!=y and the correct function value is subnormal, zero, or

9706 underflows, a range error shall occur, and[/MX] [MXX]the correct function

9707 value (if representable) or[/MXX] [MX]0.0 shall be returned.[/MX]

9708

9709 *Rationale:* Austin Group Defect Report(s) applied: 68.9710 See <http://austingroupbugs.net/view.php?id=68>

9711

9712

9713 **Change Number: XSH/TC1/D5/0408 [357]**

9714

9715

9716 On Page: 1368 Line: 44858 Section: nextafter()

9717

9718 In the APPLICATION USAGE section, add the following after L44858:
 9719
 9720 When <tgmath.h> is included, note that the return type of nextafter()
 9721 depends on the generic typing deduced from both arguments, while the
 9722 return type of nextttoward() depends only on the generic typing of the
 9723 first argument.
 9724
 9725 *Rationale:* Austin Group Defect Report(s) applied: 357.
 9726 See <http://austingroupbugs.net/view.php?id=357>
 9727
 9728
 9729 **Change Number: XSH/TC1/D5/0409 [403]**
 9730
 9731
 9732 On Page: 1369 Line: 44912 Section: nftw()
 9733
 9734 In the DESCRIPTION section, change from:
 9735
 9736 FTW_F The object is a file.
 9737
 9738 to:
 9739
 9740 FTW_F The object is a non-directory file.
 9741
 9742 *Rationale:* Austin Group Defect Report(s) applied: 403.
 9743 See <http://austingroupbugs.net/view.php?id=403>
 9744
 9745
 9746 **Change Number: XSH/TC1/D5/0410 [324]**
 9747
 9748
 9749 On Page: 1370 Line: 44944 Section: nftw()
 9750
 9751 In the ERRORS section, for the [ENOTDIR] error, change from:
 9752
 9753 A component of path is not a directory.
 9754
 9755 to:
 9756
 9757 A component of path names an existing file that is neither
 9758 a directory nor a symbolic link to a directory.
 9759
 9760 *Rationale:* Austin Group Defect Report(s) applied: 324.
 9761 See <http://austingroupbugs.net/view.php?id=324>
 9762 This is an editorial issue clarifying the intent of the
 9763 standard.
 9764
 9765
 9766 **Change Number: XSH/TC1/D5/0411 [403]**
 9767
 9768
 9769 On Page: 1371 Line: 44972 Section: nftw()
 9770
 9771 In the EXAMPLES section, change from:
 9772
 9773 (tflag == FTW_D) ? "d" : (tflag == FTW_DNR) ? "dnr" :
 9774 (tflag == FTW_DP) ? "dp" : (tflag == FTW_F) ? "f" :
 9775 (tflag == FTW_NS) ? "ns" : (tflag == FTW_SL) ? "sl" :
 9776 (tflag == FTW_SLN) ? "sln" : "???",
 9777
 9778 to:
 9779
 9780 (tflag == FTW_D) ? "d" : (tflag == FTW_DNR) ? "dnr" :

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

9781 (tflag == FTW_DP) ? "dp" : (tflag == FTW_F) ?
 9782 (S_ISBLK(sb->st_mode) ? "f b" :
 9783 S_ISCHR(sb->st_mode) ? "f c" :
 9784 S_ISFIFO(sb->st_mode) ? "f p" :
 9785 S_ISREG(sb->st_mode) ? "f r" :
 9786 S_ISSOCK(sb->st_mode) ? "f s" : "f ?") :
 9787 (tflag == FTW_NS) ? "ns" : (tflag == FTW_SL) ? "sl" :
 9788 (tflag == FTW_SLN) ? "sln" : "?",
 9789
 9790 *Rationale:* Austin Group Defect Report(s) applied: 403.
 9791 See <http://austingroupbugs.net/view.php?id=403>
 9792
 9793
 9794 **Change Number: XSH/TC1/D5/0412** [302]
 9795
 9796
 9797 On Page: 1375 Line: 45090 Section: nl_langinfo()
 9798
 9799 change from:
 9800
 9801 ... the locale of the process ...
 9802
 9803 to:
 9804
 9805 ... the current locale ...
 9806
 9807 *Rationale:* Austin Group Defect Report(s) applied: 302.
 9808 See <http://austingroupbugs.net/view.php?id=302>
 9809 These changes were overlooked during the revision when
 9810 per-thread locales were added.
 9811
 9812
 9813 **Change Number: XSH/TC1/D5/0413** [75]
 9814
 9815
 9816 On Page: 1375 Line: 45100 Section: nl_langinfo()
 9817
 9818 In the DESCRIPTION section, delete:
 9819
 9820 Calls to setlocale() with a category corresponding to the category of item
 9821 (see <langinfo.h>), or to the category LC_ALL, may overwrite the array
 9822 pointed to by the return value. Calls to uselocale() which change the
 9823 category corresponding to the category of item may overwrite the array
 9824 pointed to by the return value.
 9825
 9826 *Rationale:* Austin Group Defect Report(s) applied: 75.
 9827 See <http://austingroupbugs.net/view.php?id=75>
 9828
 9829
 9830 **Change Number: XSH/TC1/D5/0414** [283]
 9831
 9832
 9833 On Page: 1375 Line: 45104 Section: nl_langinfo()
 9834
 9835 In the DESCRIPTION section, add a new paragraph to the end of the section:
 9836
 9837 The behavior is undefined if the locale argument to nl_langinfo_l()
 9838 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 9839 object handle.
 9840
 9841 *Rationale:* Austin Group Defect Report(s) applied: 283.
 9842 See <http://austingroupbugs.net/view.php?id=283>
 9843

9844
9845 **Change Number: XSH/TC1/D5/0415 [75,402]**
9846
9847
9848 On Page: 1375 Line: 45109 Section: nl_langinfo()
9849
9850 In the RETURN VALUE section, change from:
9851
9852 This pointer may point to static data that may be overwritten on the
9853 next call to either function.
9854
9855 to:
9856
9857 The application shall not modify the string returned. The pointer
9858 returned by nl_langinfo() might be invalidated or the string content might
9859 be overwritten by a subsequent call to nl_langinfo() in any thread or to
9860 nl_langinfo_l() in the same thread or the initial thread, by subsequent
9861 calls to setlocale() with a category corresponding to the category of
9862 item (see <langinfo.h>) or the category LC_ALL, or by subsequent calls
9863 to uselocale() which change the category corresponding to the category
9864 of item. The pointer returned by nl_langinfo_l() might be invalidated
9865 or the string content might be overwritten by a subsequent call to
9866 nl_langinfo_l() in the same thread or to nl_langinfo() in any thread,
9867 or by subsequent calls to freelocale() or newlocale() which free or
9868 modify the locale object that was passed to nl_langinfo_l().
9869
9870 *Rationale:* Austin Group Defect Report(s) applied: 75,402.
9871 See <http://austingroupbugs.net/view.php?id=75>
9872 See <http://austingroupbugs.net/view.php?id=402>
9873
9874
9875 **Change Number: XSH/TC1/D5/0416 [283]**
9876
9877
9878 On Page: 1375 Line: 45111 Section: nl_langinfo()
9879
9880 In the ERRORS section, change from:
9881
9882 The nl_langinfo_l() function may fail if:
9883
9884 [EINVAL] locale is not a valid locale object handle.
9885
9886 to:
9887
9888 No errors are defined.
9889
9890 *Rationale:* Austin Group Defect Report(s) applied: 283.
9891 See <http://austingroupbugs.net/view.php?id=283>
9892
9893
9894 **Change Number: XSH/TC1/D5/0417 [402]**
9895
9896
9897 On Page: 1376 Line: 45126 Section: nl_langinfo()
9898
9899 In the RATIONALE section, change from:
9900
9901 None.
9902
9903 to:
9904
9905 The possible interactions between internal data used by nl_langinfo()
9906 and nl_langinfo_l() are complicated by the fact that nl_langinfo_l()

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

9907 must be thread-safe but `nl_langinfo()` need not be. The various
 9908 implementation choices are:
 9909

- 9910 1. `nl_langinfo_l()` and `nl_langinfo()` use separate buffers, or at
 9911 least one of them does not use an internal string buffer.
 9912 In this case there are no interactions.
- 9913 2. `nl_langinfo_l()` and `nl_langinfo()` share an internal per-thread
 9914 buffer. There can be interactions, but only in the same thread.
 9915
- 9916 3. `nl_langinfo_l()` uses an internal per-thread buffer, and
 9917 `nl_langinfo()` uses (in all threads) the same buffer that
 9918 `nl_langinfo_l()` uses in the initial thread. There can be
 9919 interactions, but only when `nl_langinfo_l()` is called in the
 9920 initial thread.
 9921

9922
 9923 *Rationale:* Austin Group Defect Report(s) applied: 402.
 9924 See <http://austingroupbugs.net/view.php?id=402>
 9925
 9926
 9927 **Change Number: XSH/TC1/D5/0418 [292]**
 9928
 9929
 9930 On Page: 1380 Line: 45209 Section: `open()`
 9931
 9932 In the DESCRIPTION section, change from:
 9933
 9934 `O_DIRECTORY` If path does not name a directory, fail and set `errno` to
 9935 `[ENOTDIR]`.
 9936
 9937 to:
 9938
 9939 `O_DIRECTORY` If path resolves to a non-directory file, fail and set `errno`
 9940 to `[ENOTDIR]`.
 9941
 9942 *Rationale:* Austin Group Defect Report(s) applied: 292.
 9943 See <http://austingroupbugs.net/view.php?id=292>
 9944 The standard is clear regarding symbolic links; however, there
 9945 is an overlap between the `[ENOENT]` and `[ENOTDIR]` error conditions. The
 9946 intention is that an `[ENOENT]` error should result when the pathname does
 9947 not resolve.
 9948
 9949
 9950 **Change Number: XSH/TC1/D5/0419 [141]**
 9951
 9952
 9953 On Page: 1380 Line: 45239 Section: `open()`
 9954
 9955 In the DESCRIPTION section (for `O_NONBLOCK`), change from:
 9956
 9957 Otherwise, the behavior of `O_NONBLOCK` is unspecified.
 9958
 9959 to:
 9960
 9961 Otherwise, the `O_NONBLOCK` flag shall not cause an error, but it is
 9962 unspecified whether the file status flags will include the `O_NONBLOCK`
 9963 flag.
 9964
 9965 *Rationale:* Austin Group Defect Report(s) applied: 141.
 9966 See <http://austingroupbugs.net/view.php?id=141>
 9967
 9968
 9969 **Change Number: XSH/TC1/D5/0420 [461]**

9970
9971
9972 On Page: 1382 Line: 45296 Section: open()
9973
9974 In the DESCRIPTION section, change from:
9975
9976 ... the current working directory is used ...
9977
9978 to:
9979
9980 ... the current working directory shall be used ...
9981
9982 *Rationale:* Austin Group Defect Report(s) applied: 461.
9983 See <http://austingroupbugs.net/view.php?id=461>
9984
9985
9986 **Change Number: XSH/TC1/D5/0421 [390]**
9987
9988
9989 On Page: 1382 Line: 45301 Section: open()
9990
9991 In the RETURN VALUE section, change from:
9992
9993 If - is returned,
9994
9995 to:
9996
9997 If -l is returned,
9998
9999 *Rationale:* Austin Group Defect Report(s) applied: 390.
10000 See <http://austingroupbugs.net/view.php?id=390>
10001
10002
10003 **Change Number: XSH/TC1/D5/0422 [146]**
10004
10005
10006 On Page: 1382 Line: 45322 Section: open()
10007
10008 In the ERRORS section, change from:
10009
10010 [ENOENT] O_CREAT is not set and the named file does not exist; or O_CREAT
10011 is set and either the path prefix does not exist or the path argument
10012 points to an empty string.
10013
10014 to:
10015
10016 [ENOENT] O_CREAT is not set and a component of path does not name an
10017 existing file, or O_CREAT is set and a component of the path prefix of
10018 path does not name an existing file, or path points to an empty string.
10019
10020 [ENOENT] or [ENOTDIR]
10021 O_CREAT is set, and the path argument contains at least one non-`<slash>`
10022 character and ends with one or more trailing `<slash>` characters. If path
10023 names an existing file, an [ENOENT] error shall not occur.
10024
10025 *Rationale:* Austin Group Defect Report(s) applied: 146.
10026 See <http://austingroupbugs.net/view.php?id=146>
10027
10028
10029 **Change Number: XSH/TC1/D5/0423 [324]**
10030
10031
10032 On Page: 1382 Line: 45329 Section: open()

10033
 10034 In the ERRORS section, for the [ENOTDIR] error, change from:
 10035
 10036 A component of the path prefix is not a directory; ...
 10037
 10038 to:
 10039
 10040 A component of the path prefix names an existing file that is neither
 10041 a directory nor a symbolic link to a directory; ...
 10042
 10043 *Rationale:* Austin Group Defect Report(s) applied: 324.
 10044 See <http://austingroupbugs.net/view.php?id=324>
 10045 This is an editorial issue clarifying the intent of the
 10046 standard.
 10047
 10048
 10049 **Change Number: XSH/TC1/D5/0424** [292]
 10050
 10051
 10052 On Page: 1382 Line: 45333 Section: open()
 10053
 10054 In the ERRORS section for the [ENOTDIR] error, change from:
 10055
 10056 ... or O_DIRECTORY was specified and the path argument does not name
 10057 a directory.
 10058
 10059 to:
 10060
 10061 ... or O_DIRECTORY was specified and the path argument resolves to a
 10062 non-directory file.
 10063
 10064 *Rationale:* Austin Group Defect Report(s) applied: 292.
 10065 See <http://austingroupbugs.net/view.php?id=292>
 10066 The standard is clear regarding symbolic links; however, there
 10067 is an overlap between the [ENOENT] and [ENOTDIR] error conditions. The
 10068 intention is that an [ENOENT] error should result when the pathname does
 10069 not resolve.
 10070
 10071
 10072 **Change Number: XSH/TC1/D5/0425** [278]
 10073
 10074
 10075 On Page: 1383 Line: 45348 open()
 10076
 10077 In the ERRORS section, add (after the [EBADF] error):
 10078
 10079 [ENOTDIR] The path argument is not an absolute path and fd is a file
 10080 descriptor associated with a non-directory file.
 10081
 10082 *Rationale:* Austin Group Defect Report(s) applied: 278.
 10083 See <http://austingroupbugs.net/view.php?id=278>
 10084
 10085
 10086 **Change Number: XSH/TC1/D5/0426** [278]
 10087
 10088
 10089 On Page: 1383 Line: 45364 Section: open()
 10090
 10091 In the ERRORS section, delete:
 10092
 10093 The openat() function may fail if:
 10094
 10095 [ENOTDIR] The path argument is not an absolute path and fd is neither

10096 AT_FDCWD nor a file descriptor associated with a directory.
10097
10098 *Rationale:* Austin Group Defect Report(s) applied: 278.
10099 See <http://austingroupbugs.net/view.php?id=278>
10100
10101
10102 **Change Number: XSH/TC1/D5/0427** [291]
10103
10104
10105 On Page: 1384 Line: 45377-45379,45409-45411 Section: open()
10106
10107 In the EXAMPLES section, change all instances of "filename" to "pathname".
10108
10109 *Rationale:* Austin Group Defect Report(s) applied: 291.
10110 See <http://austingroupbugs.net/view.php?id=291>
10111
10112
10113 **Change Number: XSH/TC1/D5/0428** [307]
10114
10115
10116 On Page: 1385 Line: 45427 Section: open()
10117
10118 In the RATIONALE section, add the following text to the end of the first
10119 paragraph (at the end of P1385 L45427):
10120
10121 Not all combinations of flags make sense. For example, using O_SEARCH |
10122 O_CREAT will successfully open a pre-existing directory for searching,
10123 but if there is no existing file by that name, then it is unspecified
10124 whether a regular file will be created. Likewise, if a non-directory
10125 file descriptor is successfully returned, it is unspecified whether that
10126 descriptor will have execute permissions as if by O_EXEC (note that it
10127 is unspecified whether O_EXEC and O_SEARCH have the same value).
10128
10129 And then move the entire paragraph from RATIONALE to APPLICATION USAGE
10130 after P1385 L45424.
10131
10132 *Rationale:* Austin Group Defect Report(s) applied: 307.
10133 See <http://austingroupbugs.net/view.php?id=307>
10134
10135
10136 **Change Number: XSH/TC1/D5/0429** [389,401]
10137
10138
10139 On Page: 1398 Line: 45791 Section: perror()
10140
10141 In the DESCRIPTION section, after L45791 add two new paragraphs, extending
10142 the CX shading to include them:
10143
10144 On error, perror() shall set the error indicator for the stream to which
10145 stderr points, and shall set errno to indicate the error.
10146
10147 Since no value is returned, an application wishing to check for error
10148 situations should call clearerr(stderr) before calling perror(), then
10149 if ferrror(stderr) returns non-zero, the value of errno indicates which
10150 error occurred.
10151
10152 *Rationale:* Austin Group Defect Report(s) applied: 389,401.
10153 See <http://austingroupbugs.net/view.php?id=389>
10154 See <http://austingroupbugs.net/view.php?id=401>
10155
10156
10157 **Change Number: XSH/TC1/D5/0430** [389]
10158

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10159
 10160 On Page: 1398 Line: 45795 Section: perror()
 10161
 10162 In the ERRORS section, change from:
 10163
 10164 No errors are defined.
 10165
 10166 to:
 10167
 10168 [CX]Refer to fputc().[/CX]
 10169
 10170 *Rationale:* Austin Group Defect Report(s) applied: 389.
 10171 See <http://austingroupbugs.net/view.php?id=389>
 10172
 10173
 10174 **Change Number: XSH/TC1/D5/0431 [389,401]**
 10175
 10176
 10177 On Page: 1398 Line: 45810 Section: perror()
 10178
 10179 In the APPLICATION USAGE section, change from:
 10180
 10181 None.
 10182
 10183 to:
 10184
 10185 Application writers may prefer to use alternative interfaces instead of
 10186 perror(), such as strerror_r() in combination with fprintf().
 10187
 10188 *Rationale:* Austin Group Defect Report(s) applied: 389,401.
 10189 See <http://austingroupbugs.net/view.php?id=389>
 10190 See <http://austingroupbugs.net/view.php?id=401>
 10191
 10192
 10193 **Change Number: XSH/TC1/D5/0432 [14]**
 10194
 10195
 10196 On Page: 1409 Line: 46171 Section: popen()
 10197
 10198 In the SEE ALSO section, add a reference to XSH Section 2.5.
 10199
 10200 *Rationale:* Austin Group Defect Report(s) applied: 14.
 10201 See <http://austingroupbugs.net/view.php?id=14>
 10202 This is an editorial improvement
 10203
 10204
 10205 **Change Number: XSH/TC1/D5/0433 [291]**
 10206
 10207
 10208 On Page: 1422 Line: 46551 Section: posix_spawn()
 10209
 10210 In the DESCRIPTION section, change "filename" to "filename string".
 10211
 10212 *Rationale:* Austin Group Defect Report(s) applied: 291.
 10213 See <http://austingroupbugs.net/view.php?id=291>
 10214
 10215
 10216 **Change Number: XSH/TC1/D5/0434 [173]**
 10217
 10218
 10219 On Page: 1423 Line: 46584 Section: posix_spawn()
 10220
 10221 In the DESCRIPTION section, after L46584 add a new paragraph:

10222
10223 If file descriptor 0, 1, or 2 would otherwise be closed in the new process
10224 image created by `posix_spawn()` or `posix_spawnp()`, implementations may
10225 open an unspecified file for the file descriptor in the new process
10226 image. If a standard utility or a conforming application is executed
10227 with file descriptor 0 not open for reading or with file descriptor
10228 1 or 2 not open for writing, the environment in which the utility or
10229 application is executed shall be deemed non-conforming, and consequently
10230 the utility or application might not behave as described in this standard.
10231
10232 *Rationale:* Austin Group Defect Report(s) applied: 173.
10233 See <http://austingroupbugs.net/view.php?id=173>
10234
10235
10236 **Change Number: XSH/TC1/D5/0435 [173]**
10237
10238
10239 On Page: 1425 Line: 46681 Section: `posix_spawn()`
10240
10241 In the APPLICATION USAGE section, add at the end of the section:
10242
10243 See also the APPLICATION USAGE section for `exec()`.
10244
10245 *Rationale:* Austin Group Defect Report(s) applied: 173.
10246 See <http://austingroupbugs.net/view.php?id=173>
10247
10248
10249 **Change Number: XSH/TC1/D5/0436 [418]**
10250
10251
10252 On Page: 1430 Line: 46893-46895 Section: `posix_spawn_file_actions_addclose()`
10253
10254 In the ERRORS section, change from:
10255
10256 These functions shall fail if:
10257
10258 [EBADF] The value specified by `fildev` is negative or greater than or
10259 equal to `{OPEN_MAX}`.
10260
10261 to:
10262
10263 The `posix_spawn_file_actions_addopen()` function shall fail if:
10264
10265 [EBADF] The value specified by `fildev` is negative or greater than or
10266 equal to `{OPEN_MAX}`.
10267
10268 The `posix_spawn_file_actions_addclose()` function shall fail if:
10269
10270 [EBADF] The value specified by `fildev` is negative.
10271
10272 *Rationale:* Austin Group Defect Report(s) applied: 418.
10273 See <http://austingroupbugs.net/view.php?id=418>
10274
10275
10276 **Change Number: XSH/TC1/D5/0437 [149]**
10277
10278
10279 On Page: 1431 Line: 46906 Section: `posix_spawn_file_actions_addclose()`
10280
10281 In the APPLICATION USAGE section, add a new paragraph at the end of
10282 the section:
10283
10284 Implementations may use file descriptors that must be inherited into

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10285 child processes for the child process to remain conforming, such as
 10286 for message catalog or tracing purposes. Therefore, an application that
 10287 calls `posix_spawn_file_actions_addclose()` with an arbitrary integer risks
 10288 non-conforming behavior, and this function can only portably be used to
 10289 close file descriptor values that the application has obtained through
 10290 explicit actions, or for the three file descriptors corresponding to the
 10291 standard file streams. In order to avoid a race condition of leaking an
 10292 unintended file descriptor into a child process, an application should
 10293 consider opening all file descriptors with the `FD_CLOEXEC` bit set unless
 10294 the file descriptor is intended to be inherited across `exec`.

10296 *Rationale:* Austin Group Defect Report(s) applied: 149.
 10297 See <http://austingroupbugs.net/view.php?id=149>

10298
 10299

10300 **Change Number: XSH/TC1/D5/0438 [291]**

10301
 10302

10303 On Page: 1431 Line: 46915 Section: `posix_spawn_file_actions_addclose()`

10304

10305 In the RATIONALE section, change "filenames" to "pathnames".

10306

10307 *Rationale:* Austin Group Defect Report(s) applied: 291.

10308 See <http://austingroupbugs.net/view.php?id=291>

10309

10310

10311 **Change Number: XSH/TC1/D5/0439 [418]**

10312

10313

10314 On Page: 1432 Line: 46954 Section: `posix_spawn_file_actions_addclose()`

10315

10316 In the RATIONALE section, add a new paragraph at the end of the section:

10317

10318 The `posix_spawn_file_actions_addclose()` function is not required to
 10319 check whether the file descriptor is less than `{OPEN_MAX}` because on
 10320 some implementations `{OPEN_MAX}` reflects the `RLIMIT_NOFILE` soft limit
 10321 and therefore calling `setrlimit()` to reduce this limit can result
 10322 in an `{OPEN_MAX}` value less than or equal to an already open file
 10323 descriptor. Applications need to be able to close such file descriptors
 10324 on `spawn`. On implementations where `{OPEN_MAX}` does not change, it is
 10325 recommended that `posix_spawn_file_actions_addclose()` should return `EBADF`
 10326 if `fd` is greater than or equal to `{OPEN_MAX}`.

10327

10328 *Rationale:* Austin Group Defect Report(s) applied: 418.

10329 See <http://austingroupbugs.net/view.php?id=418>

10330

10331

10332 **Change Number: XSH/TC1/D5/0440 [149]**

10333

10334

10335 On Page: 1433 Line: 46997 Section: `posix_spawn_file_actions_adddup2()`

10336

10337 In the APPLICATION USAGE section, add a new paragraph at the end of
 10338 the section:

10339

10340 Implementations may use file descriptors that must be inherited into
 10341 child processes for the child process to remain conforming, such as
 10342 for message catalog or tracing purposes. Therefore, an application
 10343 that calls `posix_spawn_file_actions_adddup2()` with an arbitrary integer
 10344 for `newfdes` risks non-conforming behavior, and this function can only
 10345 portably be used to overwrite file descriptor values that the application
 10346 has obtained through explicit actions, or for the three file descriptors
 10347 corresponding to the standard file streams. In order to avoid a race

10348 condition of leaking an unintended file descriptor into a child process,
10349 an application should consider opening all file descriptors with the
10350 FD_CLOEXEC bit set unless the file descriptor is intended to be inherited
10351 across exec.

10352

10353 *Rationale:* Austin Group Defect Report(s) applied: 149.

10354 See <http://austingroupbugs.net/view.php?id=149>

10355

10356

10357 **Change Number: XSH/TC1/D5/0441** [358]

10358

10359

10360 On Page: 1483 Line: 48027 Section: `posix_trace_shutdown()`

10361

10362 In the SYNOPSIS section, shade the declaration for `posix_trace_shutdown()`

10363 and mark with margin markers OB TRC.

10364

10365 *Rationale:* Austin Group Defect Report(s) applied: 358.

10366 See <http://austingroupbugs.net/view.php?id=358>

10367 Shading has stopped early in the synopsis. Shading is correct

10368 on P1507 (the pointer page).

10369

10370

10371 **Change Number: XSH/TC1/D5/0442** [119,428]

10372

10373

10374 On Page: 1516 Line: 48952 Section: `posix_typed_mem_open()`

10375

10376 In the ERRORS section, move the [ENAMETOOLONG] error from L48952 to

10377 become a may fail error after inserting at the end of the section:

10378

10379 The `posix_typed_mem_open()` function may fail if:

10380

10381 *Rationale:* Austin Group Defect Report(s) applied: 119,428.

10382 See <http://austingroupbugs.net/view.php?id=119>

10383 See <http://austingroupbugs.net/view.php?id=428>

10384

10385

10386 **Change Number: XSH/TC1/D5/0443** [68]

10387

10388

10389 On Page: 1518 Line: 49002 Section: `pow()`

10390

10391 In the RETURN VALUE section, change from:

10392

10393 If the correct value would cause underflow, and is not representable,

10394 a range error may occur, and [MX]either 0.0 (if supported), or [MX]

10395 an implementation-defined value shall be returned.

10396

10397 to:

10398

10399 If the correct value would cause underflow, [MXX]and is not

10400 representable [MXX], a range error may occur, and `pow()`, `powf()`, and

10401 `powl()` shall return [MXX]0.0, or [MXX] (if IEC 60559 Floating-Point is

10402 not supported) an implementation-defined value no greater in magnitude

10403 than `DBL_MIN`, `FLT_MIN`, and `LDBL_MIN`, respectively.

10404

10405 *Rationale:* Austin Group Defect Report(s) applied: 68.

10406 See <http://austingroupbugs.net/view.php?id=68>

10407

10408

10409 **Change Number: XSH/TC1/D5/0444** [148]

10410

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10411
 10412 On Page: 1518 Line: 49005-49008 Section: pow()
 10413
 10414 In the RETURN VALUE section, change from:
 10415
 10416 On systems that support the IEC 60559 Floating-Point option, a pole
 10417 error shall occur ...
 10418
 10419 to:
 10420
 10421 On systems that support the IEC 60559 Floating-Point option, if x is +/-0,
 10422 a pole error shall occur ...
 10423
 10424 *Rationale:* Austin Group Defect Report(s) applied: 148.
 10425 See <http://austingroupbugs.net/view.php?id=148>
 10426
 10427
 10428 **Change Number: XSH/TC1/D5/0445 [68]**
 10429
 10430
 10431 On Page: 1519 Line: 49025 Section: pow()
 10432
 10433 In the RETURN VALUE section, change the MX shading to MXX for:
 10434
 10435 If the correct value would cause underflow, and is representable,
 10436 a range error may occur and the correct value shall be returned.
 10437
 10438 *Rationale:* Austin Group Defect Report(s) applied: 68.
 10439 See <http://austingroupbugs.net/view.php?id=68>
 10440
 10441
 10442 **Change Number: XSH/TC1/D5/0446 [372]**
 10443
 10444
 10445 On Page: 1526 Line: 49241 Section: pselect()
 10446
 10447 In the ERRORS section, [EINTR] error, remove the XSI shading from:
 10448
 10449 If SA_RESTART has been set for the interrupting signal, it is
 10450 implementation-defined whether the function restarts or returns with
 10451 [EINTR].
 10452
 10453 *Rationale:* Austin Group Defect Report(s) applied: 372.
 10454 See <http://austingroupbugs.net/view.php?id=372>
 10455
 10456
 10457 **Change Number: XSH/TC1/D5/0447 [399,428]**
 10458
 10459
 10460 On Page: 1528 Line: 49331 Section: psginfo()
 10461
 10462 In the DESCRIPTION section, add three new paragraphs to the end of
 10463 the section:
 10464
 10465 The psginfo() and psignal() functions shall not change the setting of
 10466 errno if successful.
 10467
 10468 On error, the psginfo() and psignal() functions shall set the error
 10469 indicator for the stream to which stderr points, and shall set errno to
 10470 indicate the error.
 10471
 10472 Since no value is returned, an application wishing to check for error
 10473 situations should set errno to 0, then call psginfo() or psignal(),

10474 then check errno.
10475
10476 *Rationale:* Austin Group Defect Report(s) applied: 399,428.
10477 See <http://austingroupbugs.net/view.php?id=399>
10478 See <http://austingroupbugs.net/view.php?id=428>
10479
10480
10481 **Change Number: XSH/TC1/D5/0448** [399]
10482
10483
10484 On Page: 1528 Line: 49335 Section: psiginfo()
10485
10486 In the ERRORS section, change from:
10487
10488 No errors are defined.
10489
10490 to:
10491
10492 Refer to fputc().
10493
10494 *Rationale:* Austin Group Defect Report(s) applied: 399.
10495 See <http://austingroupbugs.net/view.php?id=399>
10496
10497
10498 **Change Number: XSH/TC1/D5/0449** [399,401]
10499
10500
10501 On Page: 1528 Line: 49339 Section: psiginfo()
10502
10503 In the APPLICATION USAGE section, change from:
10504
10505 None.
10506
10507 to:
10508
10509 As an alternative to setting errno to zero before the call and checking
10510 if it is non-zero afterwards, applications can use ferror() to detect
10511 whether psiginfo() or psignal() encountered an error.
10512
10513 An application wishing to use this method to check for error situations
10514 should call clearerr(stderr) before calling psiginfo() or psignal(),
10515 then if ferror(stderr) returns non-zero, the value of errno indicates
10516 which error occurred.
10517
10518 *Rationale:* Austin Group Defect Report(s) applied: 399,401.
10519 See <http://austingroupbugs.net/view.php?id=399>
10520 See <http://austingroupbugs.net/view.php?id=401>
10521
10522
10523 **Change Number: XSH/TC1/D5/0450** [314]
10524
10525
10526 On Page: 1540 Line: 49793 Section: pthread_attr_getinheritsched()
10527
10528 In the ERRORS section, insert before L49793:
10529
10530 The pthread_attr_setinheritsched() function shall fail if:
10531
10532 Move the [ENOTSUP] error from L49795 to after this new line.
10533
10534 *Rationale:* Austin Group Defect Report(s) applied: 314.
10535 See <http://austingroupbugs.net/view.php?id=314>
10536

10537
 10538 **Change Number: XSH/TC1/D5/0451** [314]
 10539
 10540
 10541 On Page: 1542 Line: 49864 Section: pthread_attr_getschedparam()
 10542
 10543 In the ERRORS section, insert before L49864:
 10544
 10545 The pthread_attr_setschedparam() function shall fail if:
 10546
 10547 Move the [ENOTSUP] error from L49866 to after this new line.
 10548
 10549 *Rationale:* Austin Group Defect Report(s) applied: 314.
 10550 See <http://austingroupbugs.net/view.php?id=314>
 10551
 10552
 10553 **Change Number: XSH/TC1/D5/0452** [314]
 10554
 10555
 10556 On Page: 1544 Line: 49923 Section: pthread_attr_getschedpolicy()
 10557
 10558 In the ERRORS section, insert before L49923:
 10559
 10560 The pthread_attr_setschedpolicy() function shall fail if:
 10561
 10562 Move the [ENOTSUP] error from L49925 to after this new line.
 10563
 10564 *Rationale:* Austin Group Defect Report(s) applied: 314.
 10565 See <http://austingroupbugs.net/view.php?id=314>
 10566
 10567
 10568 **Change Number: XSH/TC1/D5/0453** [314]
 10569
 10570
 10571 On Page: 1546 Line: 49986 Section: pthread_attr_getscope()
 10572
 10573 In the ERRORS section, insert before L49986:
 10574
 10575 The pthread_attr_setscope() function shall fail if:
 10576
 10577 Move the [ENOTSUP] error from L49988 to after this new line.
 10578
 10579 *Rationale:* Austin Group Defect Report(s) applied: 314.
 10580 See <http://austingroupbugs.net/view.php?id=314>
 10581
 10582 **Change Number: XSH/TC1/D5/0454** [229]
 10583
 10584
 10585 On Page: 1575-1576 Line: 50593 Section: pthread_cleanup_pop()
 10586
 10587 On L50593, in the EXAMPLES section, change from:
 10588
 10589 while ((l->lock_count < 0) && (l->waiting_writers != 0))
 10590
 10591 to:
 10592
 10593 while ((l->lock_count < 0) || (l->waiting_writers != 0))
 10594
 10595 On L50608 change:
 10596
 10597 pthread_mutex_unlock(l);
 10598
 10599 to:

10600
10601 pthread_mutex_unlock(&l->lock);
10602
10603 On L50617 change from:
10604
10605 * This only happens if we have been canceled.
10606
10607 to:
10608
10609 * This only happens if we have been canceled. If the
10610 * lock is not held by a writer, there may be readers who
10611 * were blocked because waiting_writers was positive; they
10612 * can now be unblocked.
10613
10614 On L50619 change from:
10615
10616 pthread_cond_broadcast(&l->wcond);
10617
10618 to:
10619
10620 pthread_cond_broadcast(&l->rcond);
10621
10622 On L50620 increase the indentation of the closing brace from zero <space>
10623 characters to four <space> characters.
10624
10625 On L50644 change from:
10626
10627 pthread_cond_broadcast(&l->rcond)
10628
10629 to:
10630
10631 pthread_cond_broadcast(&l->rcond);
10632
10633 Rationale: Austin Group Defect Report(s) applied: 229.
10634 See <http://austingroupbugs.net/view.php?id=229>
10635
10636
10637 **Change Number: XSH/TC1/D5/0455** [70]
10638
10639
10640 On Page: 1582 Line: 50877 Section: pthread_cond_destroy()
10641
10642 In the DESCRIPTION section, change from:
10643
10644 In cases where default condition variable attributes are appropriate,
10645 the macro PTHREAD_COND_INITIALIZER can be used to initialize condition
10646 variables that are statically allocated.
10647
10648 to:
10649
10650 In cases where default condition variable attributes are appropriate,
10651 the macro PTHREAD_COND_INITIALIZER can be used to initialize condition
10652 variables.
10653
10654 Rationale: Austin Group Defect Report(s) applied: 70.
10655 See <http://austingroupbugs.net/view.php?id=70>
10656
10657
10658 **Change Number: XSH/TC1/D5/0456** [91,286,437]
10659
10660
10661 On Page: 1587 Line: 51027-51034 Section: pthread_cond_timedwait()
10662

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10663 In the DESCRIPTION section, change from:
 10664
 10665 The pthread_cond_timedwait() function shall be equivalent to
 10666 pthread_cond_wait(), except that an error is returned if the absolute
 10667 time specified by abstime passes (that is, system time equals or exceeds
 10668 abstime) before the condition cond is signaled or broadcasted, or if
 10669 the absolute time specified by abstime has already been passed at the
 10670 time of the call.
 10671
 10672 The condition variable shall have a clock attribute which specifies the
 10673 clock that shall be used to measure the time specified by the abstime
 10674 argument. When such timeouts occur, pthread_cond_timedwait() shall
 10675 nonetheless release and re-acquire the mutex referenced by mutex. The
 10676 pthread_cond_timedwait() function is also a cancellation point.
 10677
 10678 to:
 10679
 10680 The pthread_cond_timedwait() function shall be equivalent to
 10681 pthread_cond_wait(), except that an error is returned if the absolute
 10682 time specified by abstime passes (that is, system time equals or exceeds
 10683 abstime) before the condition cond is signaled or broadcasted, or if
 10684 the absolute time specified by abstime has already been passed at the
 10685 time of the call. When such timeouts occur, pthread_cond_timedwait()
 10686 shall nonetheless release and re-acquire the mutex referenced by mutex,
 10687 and may consume a condition signal directed concurrently at the condition
 10688 variable.
 10689
 10690 The condition variable shall have a clock attribute which specifies
 10691 the clock that shall be used to measure the time specified by the
 10692 abstime argument. The pthread_cond_timedwait() function is also a
 10693 cancellation point.
 10694
 10695 *Rationale:* Austin Group Defect Report(s) applied: 91,286,437.
 10696 See <http://austingroupbugs.net/view.php?id=91>
 10697 See <http://austingroupbugs.net/view.php?id=286>
 10698 See <http://austingroupbugs.net/view.php?id=437>
 10699
 10700
 10701 **Change Number: XSH/TC1/D5/0457 [239]**
 10702
 10703
 10704 On Page: 1590 Line: 51186 Section: pthread_cond_timedwait()
 10705
 10706 In the RATIONALE section, change from:
 10707
 10708 if (rc == 0) setmystate(&t);
 10709
 10710 to:
 10711
 10712 if (rc == 0 || mypredicate(&t))
 10713 setmystate(&t);
 10714
 10715 *Rationale:* Austin Group Defect Report(s) applied: 239.
 10716 See <http://austingroupbugs.net/view.php?id=239>
 10717
 10718
 10719 **Change Number: XSH/TC1/D5/0458 [302]**
 10720
 10721
 10722 On Page: 1601 Line: 51456 Section: pthread_create()
 10723
 10724 In the DESCRIPTION section, change from:
 10725

10726 [XSI]The alternate stack shall not be inherited.[/XSI]
10727
10728 to:
10729
10730 The thread-local current locale [XSI]and the alternate stack[/XSI]
10731 shall not be inherited.
10732
10733 *Rationale:* Austin Group Defect Report(s) applied: 302.
10734 See <http://austingroupbugs.net/view.php?id=302>
10735 These changes were overlooked during the revision when
10736 per-thread locales were added.
10737
10738
10739 **Change Number: XSH/TC1/D5/0459 [314]**
10740
10741
10742 On Page: 1613 Line: 51845 Section: pthread_getschedparam()
10743
10744 In the ERRORS section, insert before L51845:
10745
10746 The pthread_setschedparam() function shall fail if:
10747
10748 Move the two [ENOTSUP] errors from L51848-51851 to after this new line.
10749
10750 *Rationale:* Austin Group Defect Report(s) applied: 314.
10751 See <http://austingroupbugs.net/view.php?id=314>
10752
10753
10754 **Change Number: XSH/TC1/D5/0460 [70,428]**
10755
10756
10757 On Page: 1628 Line: 52373 Section: pthread_mutex_init()
10758
10759 In the DESCRIPTION section, change from:
10760
10761 In cases where default mutex attributes are appropriate, the macro
10762 PTHREAD_MUTEX_INITIALIZER can be used to initialize mutexes that are
10763 statically allocated.
10764
10765 to:
10766
10767 In cases where default mutex attributes are appropriate, the macro
10768 PTHREAD_MUTEX_INITIALIZER can be used to initialize mutexes.
10769
10770 *Rationale:* Austin Group Defect Report(s) applied: 70,428.
10771 See <http://austingroupbugs.net/view.php?id=70>
10772 See <http://austingroupbugs.net/view.php?id=428>
10773
10774
10775 **Change Number: XSH/TC1/D5/0461 [121]**
10776
10777
10778 On Page: 1638 Line: 52690 Section: pthread_mutex_lock()
10779
10780 In the DESCRIPTION section, change from:
10781
10782 The mutex object referenced by mutex shall be locked by calling
10783 pthread_mutex_lock(). If the mutex is already locked, the calling
10784 thread shall block until the mutex becomes available. This operation
10785 shall return with the mutex object referenced by mutex in the locked
10786 state with the calling thread as its owner.
10787
10788 If the mutex type is PTHREAD_MUTEX_NORMAL, deadlock detection shall

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10789 not be provided. Attempting to relock the mutex causes deadlock. If
 10790 a thread attempts to unlock a mutex that it has not locked or a mutex
 10791 which is unlocked, undefined behavior results.
 10792

10793 If the mutex type is PTHREAD_MUTEX_ERRORCHECK, then error checking shall
 10794 be provided. If a thread attempts to relock a mutex that it has already
 10795 locked, an error shall be returned. If a thread attempts to unlock a
 10796 mutex that it has not locked or a mutex which is unlocked, an error
 10797 shall be returned.
 10798

10799 If the mutex type is PTHREAD_MUTEX_RECURSIVE, then the mutex shall
 10800 maintain the concept of a lock count. When a thread successfully acquires
 10801 a mutex for the first time, the lock count shall be set to one. Every
 10802 time a thread relocks this mutex, the lock count shall be incremented
 10803 by one. Each time the thread unlocks the mutex, the lock count shall be
 10804 decremented by one. When the lock count reaches zero, the mutex shall
 10805 become available for other threads to acquire. If a thread attempts
 10806 to unlock a mutex that it has not locked or a mutex which is unlocked,
 10807 an error shall be returned.
 10808

10809 If the mutex type is PTHREAD_MUTEX_DEFAULT, attempting to recursively
 10810 lock the mutex results in undefined behavior. Attempting to unlock the
 10811 mutex if it was not locked by the calling thread results in undefined
 10812 behavior. Attempting to unlock the mutex if it is not locked results in
 10813 undefined behavior.
 10814

10815 to:

10816
 10817 The mutex object referenced by mutex shall be locked by a call to
 10818 pthread_mutex_lock() that returns zero or EOWNERDEAD. If the mutex is
 10819 already locked by another thread, the calling thread shall block until the
 10820 mutex becomes available. This operation shall return with the mutex object
 10821 referenced by mutex in the locked state with the calling thread as its
 10822 owner. If a thread attempts to relock a mutex that it has already locked,
 10823 pthread_mutex_lock() shall behave as described in the Relock column of the
 10824 following table. If a thread attempts to unlock a mutex that it has not
 10825 locked or a mutex which is unlocked, pthread_mutex_unlock() shall behave
 10826 as described in the Unlock When Not Owner column of the following table.
 10827

10828

Mutex Type	Robustness	Relock	Unlock When Not Owner
normal	non-robust	deadlock	undefined behavior
normal	robust	deadlock	error returned
errorcheck	either	error returned	error returned
recursive	either	recursive (see below)	error returned
default	non-robust	undefined behavior[1]	undefined behavior[1]
default	robust	undefined behavior[1]	error returned

10847

10848 1. If the mutex type is PTHREAD_MUTEX_DEFAULT, the behavior of
 10849 pthread_mutex_lock() may correspond to one of the three other standard
 10850 mutex types as described in the table above. If it does not correspond
 10851 to one of those three, the behavior is undefined for the cases marked [1].

10852

10853 Where the table indicates recursive behavior, the mutex shall maintain
10854 the concept of a lock count. When a thread successfully acquires a
10855 mutex for the first time, the lock count shall be set to one. Every
10856 time a thread relocks this mutex, the lock count shall be incremented
10857 by one. Each time the thread unlocks the mutex, the lock count shall be
10858 decremented by one. When the lock count reaches zero, the mutex shall
10859 become available for other threads to acquire.

10860

10861 *Rationale:* Austin Group Defect Report(s) applied: 121.

10862 See <http://austingroupbugs.net/view.php?id=121>

10863

10864

10865 **Change Number: XSH/TC1/D5/0462** [92,428]

10866

10867

10868 On Page: 1639 Line: 52739 Section: pthread_mutex_lock()

10869

10870 In the RETURN VALUE section, change from:

10871

10872 pthread_mutex_lock() and pthread_mutex_unlock()

10873

10874 to

10875

10876 pthread_mutex_lock(), pthread_mutex_trylock(), and pthread_mutex_unlock()

10877

10878 Delete L52741-52742 (2nd para of RETURN VALUE).

10879

10880 *Rationale:* Austin Group Defect Report(s) applied: 92,428.

10881 See <http://austingroupbugs.net/view.php?id=92>

10882 See <http://austingroupbugs.net/view.php?id=428>

10883

10884

10885 **Change Number: XSH/TC1/D5/0463** [121]

10886

10887

10888 On Page: 1639 Line: 52763 Section: pthread_mutex_lock()

10889

10890 In the ERRORS section, change the [EPERM] error from:

10891

10892 The mutex type is PTHREAD_MUTEX_ERRORCHECK or the mutex is a robust mutex,
10893 and the current thread does not own the mutex.

10894

10895 to:

10896

10897 The mutex type is PTHREAD_MUTEX_ERRORCHECK or PTHREAD_MUTEX_RECURSIVE,
10898 or the mutex is a robust mutex, and the current thread does not own
10899 the mutex.

10900

10901 *Rationale:* Austin Group Defect Report(s) applied: 121.

10902 See <http://austingroupbugs.net/view.php?id=121>

10903

10904

10905 **Change Number: XSH/TC1/D5/0464** [121]

10906

10907

10908 On Page: 1661 Line: 53481 Section: pthread_mutexattr_gettype()

10909

10910 In the DESCRIPTION section, change from:

10911

10912 The type of mutex is contained in the type attribute of the mutex
10913 attributes. Valid mutex types include:

10914

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

10915 PTHREAD_MUTEX_NORMAL
 10916 This type of mutex does not detect deadlock. A thread attempting to
 10917 relock this mutex without first unlocking it shall deadlock. Attempting
 10918 to unlock a mutex locked by a different thread results in undefined
 10919 behavior. Attempting to unlock an unlocked mutex results in undefined
 10920 behavior.
 10921
 10922 PTHREAD_MUTEX_ERRORCHECK
 10923 This type of mutex provides error checking. A thread attempting to relock
 10924 this mutex without first unlocking it shall return with an error. A
 10925 thread attempting to unlock a mutex which another thread has locked
 10926 shall return with an error. A thread attempting to unlock an unlocked
 10927 mutex shall return with an error.
 10928
 10929 PTHREAD_MUTEX_RECURSIVE
 10930 A thread attempting to relock this mutex without first unlocking it shall
 10931 succeed in locking the mutex. The relocking deadlock which can occur
 10932 with mutexes of type PTHREAD_MUTEX_NORMAL cannot occur with this type
 10933 of mutex. Multiple locks of this mutex shall require the same number
 10934 of unlocks to release the mutex before another thread can acquire the
 10935 mutex. A thread attempting to unlock a mutex which another thread has
 10936 locked shall return with an error. A thread attempting to unlock an
 10937 unlocked mutex shall return with an error.
 10938
 10939 PTHREAD_MUTEX_DEFAULT
 10940 Attempting to recursively lock a mutex of this type results in undefined
 10941 behavior. Attempting to unlock a mutex of this type which was not locked
 10942 by the calling thread results in undefined behavior. Attempting to unlock
 10943 a mutex of this type which is not locked results in undefined behavior. An
 10944 implementation may map this mutex to one of the other mutex types.
 10945
 10946 to:
 10947
 10948 The type of mutex is contained in the type attribute of the mutex
 10949 attributes. Valid mutex types include:
 10950
 10951 PTHREAD_MUTEX_NORMAL
 10952 PTHREAD_MUTEX_ERRORCHECK
 10953 PTHREAD_MUTEX_RECURSIVE
 10954 PTHREAD_MUTEX_DEFAULT
 10955
 10956 The mutex type affects the behavior of calls which lock and unlock the
 10957 mutex. See pthread_mutex_lock() for details. An implementation may map
 10958 PTHREAD_MUTEX_DEFAULT to one of the other mutex types.
 10959
 10960 *Rationale:* Austin Group Defect Report(s) applied: 121.
 10961 See <http://austingroupbugs.net/view.php?id=121>
 10962
 10963
 10964 **Change Number: XSH/TC1/D5/0465 [70]**
 10965
 10966
 10967 On Page: 1671 Line: 53703 Section: pthread_rwlock_destroy()
 10968
 10969 In the DESCRIPTION section, change from:
 10970
 10971 In cases where default read-write lock attributes are appropriate, the
 10972 macro PTHREAD_RWLOCK_INITIALIZER can be used to initialize read-write
 10973 locks that are statically allocated.
 10974
 10975 to:
 10976
 10977 In cases where default read-write lock attributes are appropriate, the

10978 macro PTHREAD_RWLOCK_INITIALIZER can be used to initialize read-write
10979 locks.
10980
10981 *Rationale:* Austin Group Defect Report(s) applied: 70.
10982 See <http://austingroupbugs.net/view.php?id=70>
10983
10984
10985 **Change Number: XSH/TC1/D5/0466** [314]
10986
10987
10988 On Page: 1698 Line: 54396 Section: pthread_setschedprio()
10989
10990 In the ERRORS section, delete the [ENOTSUP] error.
10991
10992 *Rationale:* Austin Group Defect Report(s) applied: 314.
10993 See <http://austingroupbugs.net/view.php?id=314>
10994
10995
10996 **Change Number: XSH/TC1/D5/0467** [319]
10997
10998
10999 On Page: 1701 Line: 54469 Section: pthread_sigmask()
11000
11001 Change:
11002
11003 ... unless the signal was generated by the kill() function, the
11004 sigqueue() function, or the raise() function.
11005
11006 to:
11007
11008 ... unless the signal was generated by the action of another
11009 process, or by one of the functions kill(), pthread_kill(),
11010 raise(), or sigqueue().
11011
11012 *Rationale:* Austin Group Defect Report(s) applied: 319.
11013 See <http://austingroupbugs.net/view.php?id=319>
11014
11015
11016 **Change Number: XSH/TC1/D5/0468** [75]
11017
11018
11019 On Page: 1712 Line: 54752 Section: ptsname()
11020
11021 In the RETURN VALUE section, add a new paragraph at the end of the
11022 section:
11023
11024 The application shall not modify the string returned. The returned
11025 pointer might be invalidated or the string content might be overwritten
11026 by a subsequent call to ptsname().
11027
11028 On L54758, in the APPLICATION USAGE section, change from:
11029
11030 The value returned may point to a static data area that is overwritten
11031 by each call to ptsname().
11032
11033 to:
11034
11035 None.
11036
11037 *Rationale:* Austin Group Defect Report(s) applied: 75.
11038 See <http://austingroupbugs.net/view.php?id=75>
11039
11040

11041 **Change Number: XSH/TC1/D5/0469** [96]
11042
11043
11044 On Page: 1712 Line: 54760 Section: ptsname()
11045
11046 In the RATIONALE section, change from:
11047
11048 None.
11049
11050 to:
11051
11052 See RATIONALE for posix_openpt().
11053
11054 On L54764, in the SEE ALSO section, add posix_openpt().
11055
11056 *Rationale:* Austin Group Defect Report(s) applied: 96.
11057 See <http://austingroupbugs.net/view.php?id=96>
11058 Add reference to posix_openpt() to the RATIONALE and SEE
11059 ALSO sections.
11060
11061
11062 **Change Number: XSH/TC1/D5/0470** [14]
11063
11064
11065 On Page: 1713 Line: 54800 Section: putc()
11066
11067 In the SEE ALSO section, add a reference to XSH Section 2.5.
11068
11069 *Rationale:* Austin Group Defect Report(s) applied: 14.
11070 See <http://austingroupbugs.net/view.php?id=14>
11071 This is an editorial improvement
11072
11073
11074 **Change Number: XSH/TC1/D5/0471** [14]
11075
11076
11077 On Page: 1715 Line: 54834 Section: putchar()
11078
11079 In the SEE ALSO section, add a reference to XSH Section 2.5.
11080
11081 *Rationale:* Austin Group Defect Report(s) applied: 14.
11082 See <http://austingroupbugs.net/view.php?id=14>
11083 This is an editorial improvement
11084
11085
11086 **Change Number: XSH/TC1/D5/0472** [167]
11087
11088
11089 On Page: 1717 Line: 54880 Section: putenv()
11090
11091 In the APPLICATION USAGE section, change from:
11092
11093 The setenv() function is preferred over this function.
11094
11095 to:
11096
11097 The setenv() function is preferred over this function. One reason is
11098 that putenv() is optional and therefore less portable. Another is that
11099 using putenv() can slow down environment searches, as explained in the
11100 RATIONALE for getenv().
11101
11102 *Rationale:* Austin Group Defect Report(s) applied: 167.
11103 See <http://austingroupbugs.net/view.php?id=167>

11104
11105
11106 **Change Number: XSH/TC1/D5/0473** [167]
11107
11108
11109 On Page: 1717 Line: 54882 Section: putenv()
11110
11111 In the RATIONALE section, change from:
11112
11113 The standard developers noted that putenv() is the only function available
11114 to add to the environment without permitting memory leaks.
11115
11116 to:
11117
11118 Refer to the RATIONALE section in setenv().
11119
11120 *Rationale:* Austin Group Defect Report(s) applied: 167.
11121 See <http://austingroupbugs.net/view.php?id=167>
11122
11123
11124 **Change Number: XSH/TC1/D5/0474** [273,438]
11125
11126
11127 On Page: 1717 Line: 54855 Section: putenv()
11128
11129 In the DESCRIPTION section, remove the last sentence of the section
11130 (L54855-54856):
11131
11132 The space used by string is no longer used once a new string which
11133 defines name is passed to putenv().
11134
11135 *Rationale:* Austin Group Defect Report(s) applied: 273,438.
11136 See <http://austingroupbugs.net/view.php?id=273>
11137 See <http://austingroupbugs.net/view.php?id=438>
11138
11139
11140 **Change Number: XSH/TC1/D5/0475** [273]
11141
11142
11143 On Page: 1717 Line: 54879 Section: putenv()
11144
11145 In the APPLICATION USAGE section, add after L54879:
11146
11147 Although the space used by string is no longer used once a new string
11148 which defines name is passed to putenv(), if any thread in the application
11149 has used getenv() to retrieve a pointer to this variable, it should
11150 not be freed by calling free(). If the changed environment variable
11151 is one known by the system (such as the locale environment variables)
11152 the application should never free the buffer used by earlier calls to
11153 putenv() for the same variable.
11154
11155 *Rationale:* Austin Group Defect Report(s) applied: 273.
11156 See <http://austingroupbugs.net/view.php?id=273>
11157
11158
11159 **Change Number: XSH/TC1/D5/0476** [174,412]
11160
11161
11162 On Page: 1723 Line: 55077 Section: puts()
11163
11164 In the APPLICATION USAGE section, add to the end of the section:
11165
11166 This volume of POSIX.1-2008 requires that successful completion simply

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

11167 return a non-negative integer. There are at least three known different
 11168 implementation conventions for this requirement:
 11169
 11170 * Return a constant value.
 11171 * Return the last character written.
 11172 * Return the number of bytes written. Note that this implementation
 11173 convention cannot be adhered to for strings longer than {INT_MAX}
 11174 bytes as the value would not be representable in the return type of
 11175 the function. For backwards compatibility, implementations can return
 11176 the number of bytes for strings of up to INT_MAX bytes, and return
 11177 INT_MAX for all longer strings.
 11178
 11179 *Rationale:* Austin Group Defect Report(s) applied: 174,412.
 11180 See <http://austingroupbugs.net/view.php?id=174>
 11181 See <http://austingroupbugs.net/view.php?id=412>
 11182
 11183
 11184 **Change Number: XSH/TC1/D5/0477 [14]**
 11185
 11186
 11187 On Page: 1724 Line: 55083 Section: puts()
 11188
 11189 In the SEE ALSO section, add a reference to XSH Section 2.5.
 11190
 11191 *Rationale:* Austin Group Defect Report(s) applied: 14
 11192 See <http://austingroupbugs.net/view.php?id=14>
 11193 This is an editorial improvement
 11194
 11195
 11196 **Change Number: XSH/TC1/D5/0478 [14]**
 11197
 11198
 11199 On Page: 1726 Line: 55126 Section: putwc()
 11200
 11201 In the SEE ALSO section, add a reference to XSH Section 2.5.
 11202
 11203 *Rationale:* Austin Group Defect Report(s) applied: 14.
 11204 See <http://austingroupbugs.net/view.php?id=14>
 11205 This is an editorial improvement
 11206
 11207
 11208 **Change Number: XSH/TC1/D5/0479 [14]**
 11209
 11210
 11211 On Page: 1727 Line: 55157 Section: putwchar()
 11212
 11213 In the SEE ALSO section, add a reference to XSH Section 2.5.
 11214
 11215 *Rationale:* Austin Group Defect Report(s) applied: 14.
 11216 See <http://austingroupbugs.net/view.php?id=14>
 11217 This is an editorial improvement
 11218
 11219
 11220 **Change Number: XSH/TC1/D5/0480 [218]**
 11221
 11222
 11223 On Page: 1739 Line: 55487 Section: read()
 11224
 11225 In the ERRORS section, change from:
 11226
 11227 [EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the
 11228 thread would be delayed.
 11229

11230 to:
11231
11232 [EAGAIN] The file is neither a pipe, nor a FIFO, nor a socket, the
11233 O_NONBLOCK flag is set for the file descriptor, and the thread would be
11234 delayed in the read operation.
11235
11236 *Rationale:* Austin Group Defect Report(s) applied: 218.
11237 See <http://austingroupbugs.net/view.php?id=218>
11238
11239
11240 **Change Number: XSH/TC1/D5/0481 [79]**
11241
11242
11243 On Page: 1739 Line: 55497 Section: read()
11244
11245 In the ERRORS section [EIO], change from:
11246
11247 ... the process is ignoring or blocking the SIGTTIN signal, or the
11248 process group is orphaned.
11249
11250 to:
11251
11252 ... and either the calling thread is blocking SIGTTIN or the process is
11253 ignoring SIGTTIN or the process group of the process is orphaned.
11254
11255 *Rationale:* Austin Group Defect Report(s) applied: 79.
11256 See <http://austingroupbugs.net/view.php?id=79>
11257
11258
11259 **Change Number: XSH/TC1/D5/0482 [218]**
11260
11261
11262 On Page: 1739 Line: 55506 Section: read()
11263
11264 In the ERRORS section, before L55506, insert:
11265
11266 The pread() function shall fail if:
11267
11268 [EINVAL] The file is a regular file or block special file, and the offset
11269 argument is negative. The file pointer shall remain unchanged.
11270
11271 [ESPIPE] The file is a pipe, FIFO, or socket.
11272
11273 *Rationale:* Austin Group Defect Report(s) applied: 218.
11274 See <http://austingroupbugs.net/view.php?id=218>
11275
11276
11277 **Change Number: XSH/TC1/D5/0483 [218]**
11278
11279
11280 On Page: 1739 Line: 55507 Section: read()
11281
11282 In the ERRORS section, change from:
11283
11284 [EAGAIN] or [EWOLDBLOCK]
11285 The file descriptor is for a socket, is marked O_NONBLOCK, and no data
11286 is waiting to be received.
11287
11288 to:
11289
11290 [EAGAIN] The file is a pipe or FIFO, the O_NONBLOCK flag is set for the
11291 file descriptor, and the thread would be delayed in the read operation.
11292

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

11293 [EAGAIN] or [EWOULDBLOCK]
 11294 The file is a socket, the O_NONBLOCK flag is set for the file descriptor,
 11295 and the thread would be delayed in the read operation.
 11296
 11297 *Rationale:* Austin Group Defect Report(s) applied: 218.
 11298 See <http://austingroupbugs.net/view.php?id=218>
 11299
 11300
 11301 **Change Number: XSH/TC1/D5/0484 [218]**
 11302
 11303
 11304 On Page: 1740 Line: 55520 Section: read()
 11305
 11306 In the ERRORS section, delete L55520-55525 (the pread() "shall fail"
 11307 errors).
 11308
 11309 *Rationale:* Austin Group Defect Report(s) applied: 218.
 11310 See <http://austingroupbugs.net/view.php?id=218>
 11311
 11312
 11313 **Change Number: XSH/TC1/D5/0485 [218,428]**
 11314
 11315
 11316 On Page: 1742 Line: 55612 Section: read()
 11317
 11318 In the RATIONALE section, add a new paragraph to the end of the section:
 11319
 11320 This volume of POSIX.1-2008 intentionally does not specify any pread()
 11321 errors related to pipes, FIFOs, and sockets other than [ESPIPE].
 11322
 11323 *Rationale:* Austin Group Defect Report(s) applied: 218,428.
 11324 See <http://austingroupbugs.net/view.php?id=218>
 11325 See <http://austingroupbugs.net/view.php?id=428>
 11326
 11327
 11328 **Change Number: XSH/TC1/D5/0486 [75]**
 11329
 11330
 11331 On Page: 1744 Line: 55685 Section: readdir()
 11332
 11333 In the DESCRIPTION section, change from:
 11334
 11335 The pointer returned by readdir() points to data which may be overwritten
 11336 by another call to readdir() on the same directory stream. This data is
 11337 not overwritten by another call to readdir() on a different directory
 11338 stream.
 11339
 11340 to:
 11341
 11342 The application shall not modify the structure to which the return
 11343 value of readdir() points, nor any storage areas pointed to by pointers
 11344 within the structure. The returned pointer, and pointers within the
 11345 structure, might be invalidated or the structure or the storage areas
 11346 might be overwritten by a subsequent call to readdir() on the same
 11347 directory stream. They shall not be affected by a call to readdir()
 11348 on a different directory stream.
 11349
 11350 *Rationale:* Austin Group Defect Report(s) applied: 75.
 11351 See <http://austingroupbugs.net/view.php?id=75>
 11352
 11353
 11354 **Change Number: XSH/TC1/D5/0487 [120]**
 11355

11356
11357 On Page: 1749 Line: 55841 Section: readlink()
11358
11359 In the NAME section, change from:
11360
11361 readlink, readlinkat - read the contents of a symbolic link relative to
11362 a directory file descriptor
11363
11364 to:
11365
11366 readlink, readlinkat - read the contents of a symbolic link
11367
11368 *Rationale:* Austin Group Defect Report(s) applied: 120.
11369 See <http://austingroupbugs.net/view.php?id=120>
11370 A correction is made to the NAME section.
11371
11372
11373 **Change Number: XSH/TC1/D5/0488 [461]**
11374
11375
11376 On Page: 1749 Line: 55862 Section: readlink()
11377
11378 In the DESCRIPTION section, change from:
11379
11380 ... the current working directory is used ...
11381
11382 to:
11383
11384 ... the current working directory shall be used ...
11385
11386 *Rationale:* Austin Group Defect Report(s) applied: 461.
11387 See <http://austingroupbugs.net/view.php?id=461>
11388
11389
11390 **Change Number: XSH/TC1/D5/0489 [1430]**
11391
11392
11393 On Page: 1749 Line: 55865 Section: readlink()
11394
11395 In the RETURN VALUE section, change from:
11396
11397 Upon successful completion, readlink() shall return the count of bytes
11398 placed in the buffer. Otherwise, it shall return a value of -1, leave
11399 the buffer unchanged, and set errno to indicate the error.
11400
11401 Upon successful completion, the readlinkat() function shall return
11402 0. Otherwise, it shall return -1 and set errno to indicate the error.
11403
11404 to:
11405
11406 Upon successful completion, these functions shall return the count of
11407 bytes placed in the buffer. Otherwise, these functions shall return
11408 a value of -1, leave the buffer unchanged, and set errno to indicate
11409 the error.
11410
11411 *Rationale:* Austin Group Defect Report(s) applied: 143.
11412 See <http://austingroupbugs.net/view.php?id=143>
11413
11414
11415 **Change Number: XSH/TC1/D5/0490 [324]**
11416
11417
11418 On Page: 1749 Line: 55880 Section: readlink()

11419
 11420 In the ERRORS section, for the [ENOTDIR] error, change from:
 11421
 11422 A component of the path prefix is not a directory, ...
 11423
 11424 to:
 11425
 11426 A component of the path prefix names an existing file that is neither
 11427 a directory nor a symbolic link to a directory, ...
 11428
 11429 *Rationale:* Austin Group Defect Report(s) applied: 324.
 11430 See <http://austingroupbugs.net/view.php?id=324>
 11431 This is an editorial issue clarifying the intent of the
 11432 standard.
 11433
 11434
 11435 **Change Number: XSH/TC1/D5/0491 [278]**
 11436
 11437
 11438 On Page: 1750 Line: 55888 readlink()
 11439
 11440 In the ERRORS section, add (after the [EBADF] error):
 11441
 11442 [ENOTDIR] The path argument is not an absolute path and fd is a file
 11443 descriptor associated with a non-directory file.
 11444
 11445 *Rationale:* Austin Group Defect Report(s) applied: 278.
 11446 See <http://austingroupbugs.net/view.php?id=278>
 11447
 11448
 11449 **Change Number: XSH/TC1/D5/0492 [278]**
 11450
 11451
 11452 On Page: 1750 Line: 55896 Section: readlink()
 11453
 11454 In the ERRORS section, delete:
 11455
 11456 The readlinkat() function may fail if:
 11457
 11458 [ENOTDIR] The path argument is not an absolute path and fd is neither
 11459 AT_FDCWD nor a file descriptor associated with a directory.
 11460
 11461 *Rationale:* Austin Group Defect Report(s) applied: 278.
 11462 See <http://austingroupbugs.net/view.php?id=278>
 11463
 11464
 11465 **Change Number: XSH/TC1/D5/0493 [455]**
 11466
 11467
 11468 On Page: 1750 Line: 55904 Section: readlink()
 11469
 11470 In the EXAMPLES section, change from:
 11471
 11472 `ssize_t len;`
 11473
 11474 to:
 11475
 11476 `ssize_t len;`
 11477
 11478 *Rationale:* Austin Group Defect Report(s) applied: 455.
 11479 See <http://austingroupbugs.net/view.php?id=455>
 11480
 11481

11482 **Change Number: XSH/TC1/D5/0494** [151,231]
11483
11484
11485 On Page: 1750 Line: 55912 Section: readlink()
11486
11487 In the RATIONALE section, delete the first sentence:
11488
11489 Since POSIX.1-2008 does not require any association of file times with
11490 symbolic links, there is no requirement that file times be updated by
11491 readlink().
11492
11493 *Rationale:* Austin Group Defect Report(s) applied: 151,231.
11494 See <http://austingroupbugs.net/view.php?id=151>
11495 See <http://austingroupbugs.net/view.php?id=231>
11496
11497
11498 **Change Number: XSH/TC1/D5/0495** [400]
11499
11500
11501 On Page: 1754 Line: 56030 Section: realloc()
11502
11503 In the DESCRIPTION section, change from:
11504
11505 The realloc() function shall change the size of the memory object pointed
11506 to by ptr to the size specified by size. The contents of the object
11507 shall remain unchanged up to the lesser of the new and old sizes. If
11508 the new size of the memory object would require movement of the object,
11509 the space for the previous instantiation of the object is freed. If
11510 the new size is larger, the contents of the newly allocated portion of
11511 the object are unspecified. If size is 0 and ptr is not a null pointer,
11512 the object pointed to is freed.
11513
11514 to:
11515
11516 The realloc() function shall deallocate the old object pointed to by
11517 ptr and return a pointer to a new object that has the size specified by
11518 size. The contents of the new object shall be the same as that of the
11519 old object prior to deallocation, up to the lesser of the new and old
11520 sizes. Any bytes in the new object beyond the size of the old object
11521 have indeterminate values. If the size of the space requested is zero,
11522 the behavior shall be implementation-defined: either a null pointer is
11523 returned, or the behavior shall be as if the size were some nonzero value,
11524 except that the returned pointer shall not be used to access an object.
11525
11526 *Rationale:* Austin Group Defect Report(s) applied: 400.
11527 See <http://austingroupbugs.net/view.php?id=400>
11528
11529
11530 **Change Number: XSH/TC1/D5/0496** [400]
11531
11532
11533 On Page: 1754 Line: 56046 Section: realloc()
11534
11535 In the RETURN VALUE section, change from:
11536
11537 Upon successful completion with a size not equal to 0, realloc() shall
11538 return a pointer to the (possibly moved) allocated space. If size is 0,
11539 either a null pointer or a unique pointer that can be successfully passed
11540 to free() shall be returned. If there is not enough available memory,
11541 realloc() shall return a null pointer [CX] and set errno to [ENOMEM].[/CX]
11542
11543 to:
11544

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

11545 Upon successful completion, realloc() shall return a pointer to the
 11546 (possibly moved) allocated space. If size is 0, either:
 11547

- 11548 * A null pointer shall be returned [CX]and errno set to an
- 11549 implementation defined value[/CX].
- 11550 * A unique pointer that can be successfully passed to free() shall be
- 11551 returned, and the memory object pointed to by ptr shall be freed. The
- 11552 application shall ensure that the pointer is not used to access an object.
- 11553

11554 If there is not enough available memory, realloc() shall return a null
 11555 pointer [CX]and set errno to [ENOMEM][CX]. If realloc() returns a null
 11556 pointer [CX]and errno has been set to a ENOMEM[/CX], the memory referenced
 11557 by ptr shall not be changed.
 11558

11559 *Rationale:* Austin Group Defect Report(s) applied: 400.
 11560 See <http://austingroupbugs.net/view.php?id=400>
 11561

11562 **Change Number: XSH/TC1/D5/0497 [400]**
 11563

11564 On Page: 1754 Line: 56056 Section: realloc()
 11565

11566 In the APPLICATION USAGE section, change from:
 11567

11568 None.
 11569

11570 to
 11571

11572 The description of realloc() has been modified from previous versions of
 11573 this standard to align with C99. Previous versions explicitly permitted
 11574 a call to realloc(p, 0) to free the space pointed to by p and return
 11575 NULL. While this behavior could be interpreted as permitted by this
 11576 version of the standard, the C language committee have indicated that
 11577 this interpretation is incorrect. Applications should assume that if
 11578 realloc returns a null pointer, the space pointed to be p has not been
 11579 freed. Since this could lead to double-frees, implementations should
 11580 also set errno if a null pointer actually indicates a failure, and
 11581 applications should only free the space if errno was changed.
 11582

11583 *Rationale:* Austin Group Defect Report(s) applied: 400.
 11584 See <http://austingroupbugs.net/view.php?id=400>
 11585

11586 **Change Number: XSH/TC1/D5/0498 [400]**
 11587

11588 On Page: 1754 Line: 56060 Section: realloc()
 11589

11590 In the FUTURE DIRECTIONS section, change from:
 11591

11592 None.
 11593

11594 to:
 11595

11596 This standard defers to the C standard. While that standard currently has
 11597 language that might permit realloc(p, 0), where p is not a null pointer,
 11598 to free p while still returning a null pointer, the committee responsible
 11599 for that standard is considering clarifying the language to explicitly
 11600 prohibit that alternative.
 11601

11602 *Rationale:* Austin Group Defect Report(s) applied: 400.
 11603 See <http://austingroupbugs.net/view.php?id=400>
 11604

11608
11609
11610 **Change Number: XSH/TC1/D5/0499** [353]
11611
11612
11613 On Page: 1756 Line: 56099 Section: realpath()
11614
11615 In the ERRORS section, [EACCES] error, change from:
11616
11617 Read or search permission was denied for a component of file_name.
11618
11619 to:
11620
11621 Search permission was denied for a component of the path prefix of
11622 file_name.
11623
11624 *Rationale:* Austin Group Defect Report(s) applied: 353.
11625 See <http://austingroupbugs.net/view.php?id=353>
11626
11627
11628 **Change Number: XSH/TC1/D5/0500** [324]
11629
11630
11631 On Page: 1756 Line: 56108 Section: realpath()
11632
11633 In the ERRORS section, for the [ENOTDIR] error, change from:
11634
11635 A component of the path prefix is not a directory, ...
11636
11637 to:
11638
11639 A component of the path prefix names an existing file that is neither
11640 a directory nor a symbolic link to a directory, ...
11641
11642 *Rationale:* Austin Group Defect Report(s) applied: 324.
11643 See <http://austingroupbugs.net/view.php?id=324>
11644 This is an editorial issue clarifying the intent of the
11645 standard.
11646
11647
11648 **Change Number: XSH/TC1/D5/0501** [353]
11649
11650
11651 On Page: 1756 Line: 56112 Section: realpath()
11652
11653 After L56112 add a new "may fail" error:
11654
11655 [EACCES] The file_name argument does not begin with a <slash> and none
11656 of the symbolic links (if any) processed during pathname resolution of
11657 file_name had contents that began with a <slash>, and either search
11658 permission was denied for the current directory or read or search
11659 permission was denied for a directory above the current directory in
11660 the file hierarchy.
11661
11662 *Rationale:* Austin Group Defect Report(s) applied: 353.
11663 See <http://austingroupbugs.net/view.php?id=353>
11664
11665
11666 **Change Number: XSH/TC1/D5/0502** [462]
11667
11668
11669 On Page: 1760 Line: 56238 Section: recv()
11670

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

11671 In the APPLICATION USAGE section, change from:
 11672
 11673 The recv() function is equivalent to recvfrom() with a zero address_len
 11674 argument, and to read() if no flags are used.
 11675
 11676 to:
 11677
 11678 The recv() function is equivalent to recvfrom() with null pointer address
 11679 and address_len arguments, and to read() if the socket argument refers
 11680 to a socket and the flags argument is 0.
 11681
 11682 *Rationale:* Austin Group Defect Report(s) applied: 462.
 11683 See <http://austingroupbugs.net/view.php?id=462>
 11684
 11685
 11686 **Change Number: XSH/TC1/D5/0503 [464]**
 11687
 11688
 11689 On Page: 1761 Line: 56282 Section: recvfrom()
 11690
 11691 In the DESCRIPTION section, change from:
 11692
 11693 address_len Specifies the length of the sockaddr structure pointed to
 11694 by the address argument.
 11695
 11696 to:
 11697
 11698 address_len Either a null pointer, if address is a null pointer, or a
 11699 pointer to a socklen_t object which on input specifies the length of
 11700 the supplied sockaddr structure, and on output specifies the length of
 11701 the stored address.
 11702
 11703 *Rationale:* Austin Group Defect Report(s) applied: 464.
 11704 See <http://austingroupbugs.net/view.php?id=464>
 11705
 11706
 11707 **Change Number: XSH/TC1/D5/0504 [464]**
 11708
 11709
 11710 On Page: 1764 Line: 56385 Section: recvmsg()
 11711
 11712 In the DESCRIPTION section, change from:
 11713
 11714 In the msghdr structure, the msg_name and msg_namelen members specify
 11715 the source address if the socket is unconnected. If the socket is
 11716 connected, the msg_name and msg_namelen members shall be ignored.
 11717 The msg_name member may be a null pointer if no names are desired or
 11718 required.
 11719
 11720 to:
 11721
 11722 In the msghdr structure, the msg_name member may be a null pointer if the
 11723 source address is not required. Otherwise, if the socket is unconnected,
 11724 the msg_name member points to a sockaddr structure in which the source
 11725 address is to be stored, and the msg_namelen member on input specifies
 11726 the length of the supplied sockaddr structure and on output specifies
 11727 the length of the stored address. If the actual length of the address is
 11728 greater than the length of the supplied sockaddr structure, the stored
 11729 address shall be truncated. If the socket is connected, the msg_name
 11730 and msg_namelen members shall be ignored.
 11731
 11732 *Rationale:* Austin Group Defect Report(s) applied: 464.
 11733 See <http://austingroupbugs.net/view.php?id=464>

11734

11735

11736 **Change Number: XSH/TC1/D5/0505 [305]**

11737

11738

11739 On Page: 1769 Line: 56541 Section: regcomp()

11740

11741 In the DESCRIPTION section, change from:

11742

11743 The following constants are defined as error return values:

11744

11745 to:

11746

11747 The following constants are defined as the minimum set of error return

11748 values, although other errors listed as implementation extensions in

11749 <regex.h> are possible:

11750

11751 *Rationale:* Austin Group Defect Report(s) applied: 305.11752 See <http://austingroupbugs.net/view.php?id=305>

11753

11754

11755 **Change Number: XSH/TC1/D5/0506 [320]**

11756

11757

11758 On Page: 1774 Line: 56764 Section: remainder()

11759

11760 In the RETURN VALUE section, change from:

11761

11762 ... and either a NaN (if supported), or an implementation-defined

11763 value shall be returned.

11764

11765 to:

11766

11767 ... and a NaN shall be returned.

11768

11769 *Rationale:* Austin Group Defect Report(s) applied: 320.11770 See <http://austingroupbugs.net/view.php?id=320>

11771

11772

11773 **Change Number: XSH/TC1/D5/0507 [320]**

11774

11775

11776 On Page: 1779 Line: 56883 Section: remquo()

11777

11778 In the RETURN VALUE section, change from:

11779

11780 ... and either a NaN (if supported), or an implementation-defined

11781 value shall be returned.

11782

11783 to:

11784

11785 ... and a NaN shall be returned.

11786

11787 *Rationale:* Austin Group Defect Report(s) applied: 320.11788 See <http://austingroupbugs.net/view.php?id=320>

11789

11790

11791 **Change Number: XSH/TC1/D5/0508 [324]**

11792

11793

11794 On Page: 1783 Line: 57005 Section: rename()

11795

11796 In the ERRORS section, for the [ENOTDIR] error, change from:

11797
11798 A component of either path prefix is not a directory; ...
11799
11800 to:
11801
11802 A component of either path prefix names an existing file that is neither
11803 a directory nor a symbolic link to a directory; ...
11804
11805 *Rationale:* Austin Group Defect Report(s) applied: 324.
11806 See <http://austingroupbugs.net/view.php?id=324>
11807 This is an editorial issue clarifying the intent of the
11808 standard.
11809
11810
11811 **Change Number: XSH/TC1/D5/0509 [147]**
11812
11813
11814 On Page: 1783 Line: 57009 Section: rename()
11815
11816 In the ERRORS section, change the [ENOTDIR] error from:
11817
11818 ... or the new argument names a nonexistent file, contains at least
11819 one non-`<slash>` character, and ends with one or more trailing
11820 `<slash>` characters.
11821
11822 to:
11823
11824 ... or the old argument names an existing non-directory file and the
11825 new argument names a nonexistent file, contains at least one non-`<slash>`
11826 character, and ends with one or more trailing `<slash>` characters; or the
11827 new argument names an existing non-directory file, contains at least
11828 one non-`<slash>` character, and ends with one or more trailing `<slash>`
11829 characters.
11830
11831 *Rationale:* Austin Group Defect Report(s) applied: 147.
11832 See <http://austingroupbugs.net/view.php?id=147>
11833
11834
11835 **Change Number: XSH/TC1/D5/0510 [379]**
11836
11837
11838 On Page: 1783 Line: 57039 Section: rename()
11839
11840 In the ERRORS section, [ETXTBSY] error, change from:
11841
11842 The file to be renamed is a pure procedure (shared text) file that is
11843 being executed.
11844
11845 to:
11846
11847 The file named by new exists and is the last directory entry to a pure
11848 procedure (shared text) file that is being executed.
11849
11850 *Rationale:* Austin Group Defect Report(s) applied: 379.
11851 See <http://austingroupbugs.net/view.php?id=379>
11852
11853
11854 **Change Number: XSH/TC1/D5/0511 [278]**
11855
11856
11857 On Page: 1783 Line: 57030 Section: rename()
11858
11859 In the ERRORS section, add (after the [EBADF] error):

11860
11861 [ENOTDIR] The old or new argument is not an absolute path and oldfd or
11862 newfd, respectively, is a file descriptor associated with a non-directory
11863 file.
11864
11865 *Rationale:* Austin Group Defect Report(s) applied: 278.
11866 See <http://austingroupbugs.net/view.php?id=278>
11867
11868
11869 **Change Number: XSH/TC1/D5/0512 [278]**
11870
11871
11872 On Page: 1783 Line: 57041 Section: rename()
11873
11874 In the ERRORS section, delete:
11875
11876 The renameat() function may fail if:
11877
11878 [ENOTDIR] The old argument is not an absolute path and oldfd is neither
11879 AT_FDCWD nor a file descriptor associated with a directory, or the new
11880 argument is not an absolute path and newfd is neither AT_FDCWD nor a
11881 file descriptor associated with a directory.
11882
11883 *Rationale:* Austin Group Defect Report(s) applied: 278.
11884 See <http://austingroupbugs.net/view.php?id=278>
11885
11886
11887 **Change Number: XSH/TC1/D5/0513 [14]**
11888
11889
11890 On Page: 1786 Line: 57142 Section: rewind()
11891
11892 In the SEE ALSO section, add a reference to XSH Section 2.5.
11893
11894 *Rationale:* Austin Group Defect Report(s) applied: 14.
11895 See <http://austingroupbugs.net/view.php?id=14>
11896 This is an editorial improvement.
11897
11898
11899 **Change Number: XSH/TC1/D5/0514 [346]**
11900
11901
11902 On Page: 1788 Line: 57201 Section: rint()
11903
11904 In the DESCRIPTION section, add to the paragraph at P1788 L57201:
11905
11906 If the current rounding mode rounds towards zero, then rint() shall be
11907 equivalent to trunc(). [MX]If the current rounding mode rounds towards
11908 nearest, then rint() differs from round() in that halfway cases are
11909 rounded to even rather than away from zero.[/MX]
11910
11911 *Rationale:* Austin Group Defect Report(s) applied: 346.
11912 See <http://austingroupbugs.net/view.php?id=346>
11913
11914
11915 **Change Number: XSH/TC1/D5/0515 [346]**
11916
11917
11918 On Page: 1788 Line: 57211 Section: rint()
11919
11920 In the RETURN VALUE section, at L57211, add:
11921
11922 [MX]The result shall have the same sign as x.[/MX]

11923
 11924 *Rationale:* Austin Group Defect Report(s) applied: 346.
 11925 See <http://austingroupbugs.net/view.php?id=346>
 11926
 11927
 11928 **Change Number: XSH/TC1/D5/0516 [346]**
 11929
 11930
 11931 On Page: 1788 Line: 57214-57216 Section: rint()
 11932
 11933 In the RETURN VALUE section, delete L57214-57216.
 11934
 11935 *Rationale:* Austin Group Defect Report(s) applied: 346.
 11936 See <http://austingroupbugs.net/view.php?id=346>
 11937
 11938
 11939 **Change Number: XSH/TC1/D5/0517 [346]**
 11940
 11941
 11942 On Page: 1788 Line: 57218-57223 Section: rint()
 11943
 11944 In the ERRORS section, replace the Range Error on L57218-57223 with:
 11945
 11946 No errors are defined.
 11947
 11948 *Rationale:* Austin Group Defect Report(s) applied: 346.
 11949 See <http://austingroupbugs.net/view.php?id=346>
 11950
 11951
 11952 **Change Number: XSH/TC1/D5/0518 [346]**
 11953
 11954
 11955 On Page: 1789 Line: 57227-57228 Section: rint()
 11956
 11957 In the APPLICATION USAGE section, change from:
 11958
 11959 On error, the expressions (matherrhandling & MATH_ERRNO) and
 11960 (math_errhandling & MATH_ERREXCEPT) are independent of each other,
 11961 but at least one of them must be non-zero.
 11962
 11963 to:
 11964
 11965 The integral value returned by these functions need not be expressible
 11966 as an intmax_t. The return value should be tested before assigning it
 11967 to an integer type to avoid the undefined results of an integer overflow.
 11968
 11969 *Rationale:* Austin Group Defect Report(s) applied: 346.
 11970 See <http://austingroupbugs.net/view.php?id=346>
 11971
 11972
 11973 **Change Number: XSH/TC1/D5/0519 [324]**
 11974
 11975
 11976 On Page: 1791 Line: 57290 Section: rmdir()
 11977
 11978 In the ERRORS section, for the [ENOTDIR] error, change from:
 11979
 11980 A component of path is not a directory.
 11981
 11982 to:
 11983
 11984 A component of path names an existing file that is neither
 11985 a directory nor a symbolic link to a directory.

11986
11987 *Rationale:* Austin Group Defect Report(s) applied: 324.
11988 See <http://austingroupbugs.net/view.php?id=324>
11989 This is an editorial issue clarifying the intent of the
11990 standard.
11991
11992
11993 **Change Number: XSH/TC1/D5/0520 [346]**
11994
11995
11996 On Page: 1793 Line: 57367 Section: round()
11997
11998 For round(), delete P1793 L57367-57370 (DESCRIPTION), L57375-57377
11999 (RETURN VALUE), and replace L57379-57384 (ERRORS) with:
12000
12001 No errors are defined.
12002
12003 At L57372 (RETURN VALUE), add:
12004
12005 [MX]The result shall have the same sign as x.[/MX]
12006
12007 Replace L57388-57389 (APPLICATION USAGE) with:
12008
12009 The integral value returned by these functions need not be expressible
12010 as an intmax_t. The return value should be tested before assigning it
12011 to an integer type to avoid the undefined results of an integer overflow.
12012
12013 These functions may raise the inexact floating-point exception if the
12014 result differs in value from the argument.
12015
12016 *Rationale:* Austin Group Defect Report(s) applied: 346.
12017 See <http://austingroupbugs.net/view.php?id=346>
12018
12019
12020 **Change Number: XSH/TC1/D5/0521 [68]**
12021
12022
12023 On Page: 1795 Line: 57424 Section: scalbln()
12024
12025 In the RETURN VALUE section, change from:
12026
12027 If the correct value would cause underflow, and is not representable,
12028 a range error may occur, and [MX]either 0.0 (if supported), or[/MX]
12029 an implementation-defined value shall be returned.
12030
12031 to:
12032
12033 If the correct value would cause underflow, [MXX]and is not
12034 representable[/MXX], a range error may occur, and scalbln(),
12035 scalblnf(), scalblnl(), scalbn(), scalbnf(), and scalbnl() shall return
12036 [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is not supported)
12037 an implementation-defined value no greater in magnitude than DBL_MIN,
12038 FLT_MIN, LDBL_MIN, DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.
12039
12040 *Rationale:* Austin Group Defect Report(s) applied: 68.
12041 See <http://austingroupbugs.net/view.php?id=68>
12042
12043
12044 **Change Number: XSH/TC1/D5/0522 [68]**
12045
12046
12047 On Page: 1795 Line: 57429 Section: scalbln()
12048

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

12049 In the RETURN VALUE section, change the MX shading to MXX for:
12050
12051 If the correct value would cause underflow, and is representable,
12052 a range error may occur and the correct value shall be returned.
12053
12054 *Rationale:* Austin Group Defect Report(s) applied: 68.
12055 See <http://austingroupbugs.net/view.php?id=68>
12056
12057
12058 **Change Number: XSH/TC1/D5/0523 [37]**
12059
12060
12061 On Page: 1812 Line: 57909 Section: sem_close()
12062
12063 In the APPLICATION USAGE section, change from:
12064
12065 The sem_close() function is part of the Semaphores option and need not
12066 be provided on all implementations.
12067
12068 to:
12069
12070 None.
12071
12072 *Rationale:* Austin Group Defect Report(s) applied: 37.
12073 See <http://austingroupbugs.net/view.php?id=37>
12074
12075
12076 **Change Number: XSH/TC1/D5/0524 [37]**
12077
12078
12079 On Page: 1814 Line: 57950 Section: sem_destroy()
12080
12081 In the APPLICATION USAGE section, change from:
12082
12083 The sem_destroy() function is part of the Semaphores option and need
12084 not be provided on all implementations.
12085
12086 to:
12087
12088 None.
12089
12090 *Rationale:* Austin Group Defect Report(s) applied: 37.
12091 See <http://austingroupbugs.net/view.php?id=37>
12092
12093
12094 **Change Number: XSH/TC1/D5/0525 [37]**
12095
12096
12097 On Page: 1816 Line: 57992 Section: sem_getvalue()
12098
12099 In the APPLICATION USAGE section, change from:
12100
12101 The sem_getvalue() function is part of the Semaphores option and need
12102 not be provided on all implementations.
12103
12104 to:
12105
12106 None.
12107
12108 *Rationale:* Austin Group Defect Report(s) applied: 37.
12109 See <http://austingroupbugs.net/view.php?id=37>
12110
12111

12112 **Change Number: XSH/TC1/D5/0526** [37]
12113
12114
12115 On Page: 1818 Line: 58049 Section: sem_init()
12116
12117 In the APPLICATION USAGE section, change from:
12118
12119 The sem_init() function is part of the Semaphores option and need not
12120 be provided on all implementations.
12121
12122 to:
12123
12124 None.
12125
12126 *Rationale:* Austin Group Defect Report(s) applied: 37.
12127 See <http://austingroupbugs.net/view.php?id=37>
12128
12129
12130 **Change Number: XSH/TC1/D5/0527** [37]
12131
12132
12133 On Page: 1822 Line: 58156 Section: sem_open()
12134
12135 In the APPLICATION USAGE section, change from:
12136
12137 The sem_open() function is part of the Semaphores option and need not
12138 be provided on all implementations.
12139
12140 to:
12141
12142 None.
12143
12144 *Rationale:* Austin Group Defect Report(s) applied: 37.
12145 See <http://austingroupbugs.net/view.php?id=37>
12146
12147
12148 **Change Number: XSH/TC1/D5/0528** [37]
12149
12150
12151 On Page: 1823 Line: 58223 Section: sem_post()
12152
12153 In the APPLICATION USAGE section, change from:
12154
12155 The sem_post() function is part of the Semaphores option and need not
12156 be provided on all implementations.
12157
12158 to:
12159
12160 None.
12161
12162 *Rationale:* Austin Group Defect Report(s) applied: 37.
12163 See <http://austingroupbugs.net/view.php?id=37>
12164
12165
12166 **Change Number: XSH/TC1/D5/0529** [138]
12167
12168
12169 On Page: 1826 Line: 58299 Section: sem_timedwait()
12170
12171 In the EXAMPLES section, change the example code for the handler()
12172 function from:
12173
12174 static void

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

```

12175 handler(int sig)
12176 {
12177     write(STDOUT_FILENO, "sem_post() from handler\n", 24);
12178     if (sem_post(&sem) == -1) {
12179         write(STDERR_FILENO, "sem_post() failed\n", 18);
12180         _exit(EXIT_FAILURE);
12181     }
12182 }
12183
12184 to:
12185
12186 static void
12187 handler(int sig)
12188 {
12189     int sav_errno = errno;
12190     static const char info_msg[] = "sem_post() from handler\n";
12191     write(STDOUT_FILENO, info_msg, sizeof info_msg - 1);
12192     if (sem_post(&sem) == -1) {
12193         static const char err_msg[] = "sem_post() failed\n";
12194         write(STDERR_FILENO, err_msg, sizeof err_msg - 1);
12195         _exit(EXIT_FAILURE);
12196     }
12197     errno = sav_errno;
12198 }
12199
12200 Rationale: Austin Group Defect Report(s) applied: 138.
12201 See http://austingroupbugs.net/view.php?id=138
12202
12203
12204 Change Number: XSH/TC1/D5/0530 [37]
12205
12206
12207 On Page: 1828 Line: 58405 Section: sem_trywait()
12208
12209 In the APPLICATION USAGE section, delete
12210
12211 The sem_trywait() and sem_wait() functions are part of the Semaphores
12212 option and need not be provided on all implementations.
12213
12214 Rationale: Austin Group Defect Report(s) applied: 37.
12215 See http://austingroupbugs.net/view.php?id=37
12216
12217
12218 Change Number: XSH/TC1/D5/0531 [37]
12219
12220
12221 On Page: 1830 Line: 58462 Section: sem_unlink()
12222
12223 In the APPLICATION USAGE section, change from:
12224
12225 The sem_unlink() function is part of the Semaphores option and need not
12226 be provided on all implementations.
12227
12228 to:
12229
12230 None.
12231
12232 Rationale: Austin Group Defect Report(s) applied: 37.
12233 See http://austingroupbugs.net/view.php?id=37
12234
12235
12236 Change Number: XSH/TC1/D5/0532 [345]
12237

```

12238
12239 On Page: 1833 Line: 58514 Section: semctl()
12240
12241 In the DESCRIPTION section, at L58514 [SETVAL], add a sentence before
12242 "Requires alter permission":
12243
12244 Also, the sem_ctime timestamp shall be set to the current time, as
12245 described in XSH Section 2.7.1.
12246
12247 *Rationale:* Austin Group Defect Report(s) applied: 345.
12248 See <http://austingroupbugs.net/view.php?id=345>
12249
12250
12251 **Change Number: XSH/TC1/D5/0533 [345]**
12252
12253
12254 On Page: 1833 Line: 58526 Section: semctl()
12255
12256 In the DESCRIPTION section, at L58526 [SETALL], add a sentence before
12257 "Requires alter permission":
12258
12259 Also, the sem_ctime timestamp shall be set to the current time, as
12260 described in XSH Section 2.7.1.
12261
12262 *Rationale:* Austin Group Defect Report(s) applied: 345.
12263 See <http://austingroupbugs.net/view.php?id=345>
12264
12265
12266 **Change Number: XSH/TC1/D5/0534 [345]**
12267
12268
12269 On Page: 1834 Line: 58541 Section: semctl()
12270
12271 In the DESCRIPTION section, after L58541 [IPC_SET], add a sentence:
12272
12273 The sem_ctime timestamp shall be set to the current time, as described
12274 in XSH Section 2.7.1.
12275
12276 *Rationale:* Austin Group Defect Report(s) applied: 345.
12277 See <http://austingroupbugs.net/view.php?id=345>
12278
12279
12280 **Change Number: XSH/TC1/D5/0535 [335]**
12281
12282
12283 On Page: 1835 Line: 58574 Section: semctl()
12284
12285 In the EXAMPLES section, change from:
12286
12287 None.
12288
12289 to:
12290
12291 Refer to semop() (on page 1841).
12292
12293 *Rationale:* Austin Group Defect Report(s) applied: 335.
12294 See <http://austingroupbugs.net/view.php?id=335>
12295
12296
12297 **Change Number: XSH/TC1/D5/0536 [335,439]**
12298
12299
12300 On Page: 1837 Line: 58642–58701 Section: semget()

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

12301
 12302 In the EXAMPLES section, replace the entire example on L58642-58701 with:
 12303
 12304 Refer to semop() (on page 1841).
 12305
 12306 On Page: 1838 Line: 58710 Section: semget()
 12307
 12308 In the FUTURE DIRECTIONS section, change from:
 12309
 12310 None.
 12311
 12312 to:
 12313
 12314 A future version may require that the value of the semval, sempid, semncnt, and
 12315 semzcnt members of all semaphores in a semaphore set be initialized to zero
 12316 when a call to semget() creates a semaphore set. Many semaphore implementations
 12317 already do this and it greatly simplifies what an application must do to
 12318 initialize a semaphore set.
 12319
 12320 *Rationale:* Austin Group Defect Report(s) applied: 335,439.
 12321 See <http://austingroupbugs.net/view.php?id=335>
 12322 See <http://austingroupbugs.net/view.php?id=439>
 12323
 12324
 12325 **Change Number: XSH/TC1/D5/0537 [344]**
 12326
 12327
 12328 On Page: 1838 Line: 58712 Section: semget()
 12329
 12330 In the SEE ALSO section, add ftok() prior to semctl().
 12331
 12332 *Rationale:* Austin Group Defect Report(s) applied: 344.
 12333 See <http://austingroupbugs.net/view.php?id=344>
 12334
 12335
 12336 **Change Number: XSH/TC1/D5/0538 [329,429]**
 12337
 12338
 12339 On Page: 1839 Line: 58742 Section: semop()
 12340
 12341 In the DESCRIPTION section, change the type of the sem_num member of
 12342 the sembuf structure in the description of semop() from:
 12343
 12344 short
 12345
 12346 to:
 12347
 12348 unsigned short
 12349
 12350 *Rationale:* Austin Group Defect Report(s) applied: 329,429.
 12351 See <http://austingroupbugs.net/view.php?id=329>
 12352 See <http://austingroupbugs.net/view.php?id=429>
 12353
 12354
 12355 **Change Number: XSH/TC1/D5/0539 [345,428]**
 12356
 12357
 12358 On Page: 1840 Line: 58793 Section: semop()
 12359
 12360 In the DESCRIPTION section, after L58793, add a sentence:
 12361
 12362 Also, the sem_otime timestamp shall be set to the current time, as
 12363 described in XSH Section 2.7.1.

12364
 12365 *Rationale:* Austin Group Defect Report(s) applied: 345,428.
 12366 See <http://austingroupbugs.net/view.php?id=345>
 12367 See <http://austingroupbugs.net/view.php?id=428>
 12368
 12369
 12370 **Change Number: XSH/TC1/D5/0540** [329,429]
 12371
 12372
 12373 On Page: 1840 Line: 58804 Section: semop()
 12374
 12375 In the ERRORS section, for the [EFBIG] error, delete the text:
 12376
 12377 less than 0 or.
 12378
 12379 *Rationale:* Austin Group Defect Report(s) applied: 329,429.
 12380 See <http://austingroupbugs.net/view.php?id=329>
 12381 See <http://austingroupbugs.net/view.php?id=429>
 12382
 12383
 12384 **Change Number: XSH/TC1/D5/0541** [335]
 12385
 12386
 12387 On Page: 1841-1842 Line: 58825 Section: semop()
 12388
 12389 In the EXAMPLES section, after L58825 insert:
 12390
 12391 /* Code to initialize semid. */
 12392 ...
 12393
 12394 At L58841, change:
 12395
 12396 uses a call to semop() to initialize it
 12397
 12398 to:
 12399
 12400 uses calls to semctl() and semop() to initialize it
 12401
 12402 After L58844, add a paragraph:
 12403
 12404 Processes that obtain semid without creating it check that sem_otime is
 12405 non-zero, to ensure that the creating process has completed the semop()
 12406 initialization.
 12407
 12408 Delete L58848 (<sys/types.h>), L58850 (<sys/ipc.h>), L58854 (<unistd.h>),
 12409 and L58856-58858 (<pwd.h>, <fcntl.h>, and <limits.h>).
 12410
 12411 At L58861, delete ", pfd, fv", leaving just "int semid;".
 12412
 12413 Delete L58863-58866 (lgn, filename, outstat, pw), and replace them with:
 12414
 12415 union semun {
 12416 int val;
 12417 struct semid_ds *buf;
 12418 unsigned short *array;
 12419 } arg;
 12420 struct semid_ds ds;
 12421
 12422 After L58878, insert:
 12423
 12424 arg.val = 0;
 12425
 12426 Replace L58882 with:

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

```

12427
12428 if (semctl(semid, 0, SETVAL, arg) == -1
12429     || semop(semid, &sbuf, 1) == -1) {
12430
12431 After L58889, insert:
12432
12433 goto check_init;
12434
12435 After L58894, insert:
12436
12437 else
12438 {
12439     /* Check that semid has completed initialization. */
12440     /* An application may use a retry loop at this point rather than exiting.
12441     */
12442     check_init:
12443     arg.buf = &ds;
12444     if (semctl(semid, 0, IPC_STAT, arg) < 0) {
12445         perror("IPC error 3: semctl"); exit(1);
12446     }
12447     if (ds.sem_otime == 0) {
12448         perror("IPC error 4: semctl"); exit(1);
12449     }
12450 }
12451
12452 Rationale: Austin Group Defect Report(s) applied: 335.
12453 See http://austingroupbugs.net/view.php?id=335
12454
12455 Change Number: XSH/TC1/D5/0542 [291,429]
12456
12457
12458 On Page: 1842 Line: 58864 Section: semop()
12459
12460 In the EXAMPLES section, delete "char filename[PATH_MAX+1]".
12461
12462 Rationale: Austin Group Defect Report(s) applied: 291,429.
12463 See http://austingroupbugs.net/view.php?id=291
12464 See http://austingroupbugs.net/view.php?id=429
12465 This declaration is unused in this example.
12466
12467
12468 Change Number: XSH/TC1/D5/0543 [463]
12469
12470 On Page: 1845 Line: 58990 Section: send()
12471
12472 In the APPLICATION USAGE section, change from:
12473
12474 The send() function is equivalent to sendto() with a null pointer dest_len
12475 argument, and to write() if no flags are used.
12476
12477 to:
12478
12479 If the socket argument refers to a connection-mode socket, the send()
12480 function is equivalent to sendto() (with any value for the dest_addr and
12481 dest_len arguments, as they are ignored in this case). If the socket
12482 argument refers to a socket and the flags argument is 0, the send()
12483 function is equivalent to write().
12484
12485 Rationale: Austin Group Defect Report(s) applied: 463.
12486 See http://austingroupbugs.net/view.php?id=463
12487
12488
12489

```

12490
12491 **Change Number: XSH/TC1/D5/0544 [324]**
12492
12493
12494 On Page: 1848 Line: 59085 Section: sendmsg()
12495
12496 In the ERRORS section, for the [ENOTDIR] error, change from:
12497
12498 A component of the path prefix of the pathname in the socket address is
12499 not a directory, ...
12500
12501 to:
12502
12503 A component of the path prefix of the pathname in the socket address
12504 names an existing file that is neither a directory nor a symbolic link
12505 to a directory, ...
12506
12507 *Rationale:* Austin Group Defect Report(s) applied: 324.
12508 See <http://austingroupbugs.net/view.php?id=324>
12509 This is an editorial issue clarifying the intent of the
12510 standard.
12511
12512
12513 **Change Number: XSH/TC1/D5/0545 [324]**
12514
12515
12516 On Page: 1852 Line: 59210 Section: sendto()
12517
12518 In the ERRORS section, for the [ENOTDIR] error, change from:
12519
12520 A component of the path prefix of the pathname in the socket address is
12521 not a directory, ...
12522
12523 to:
12524
12525 A component of the path prefix of the pathname in the socket address
12526 names an existing file that is neither a directory nor a symbolic link
12527 to a directory, ...
12528
12529 *Rationale:* Austin Group Defect Report(s) applied: 324.
12530 See <http://austingroupbugs.net/view.php?id=324>
12531 This is an editorial issue clarifying the intent of the
12532 standard.
12533
12534
12535 **Change Number: XSH/TC1/D5/0546 [397]**
12536
12537
12538 On Page: 1855 Line: 59282 Section: setbuf()
12539
12540 In the ERRORS section, change from:
12541
12542 No errors are defined.
12543
12544 to:
12545
12546 Although the setvbuf() interface may set errno in defined ways, the
12547 value of errno after a call to setbuf() is unspecified.
12548
12549 *Rationale:* Austin Group Defect Report(s) applied: 397.
12550 See <http://austingroupbugs.net/view.php?id=397>
12551 Existing implementations do not work to preserve errno values
12552 in the case of no error.

12553

12554

12555 **Change Number: XSH/TC1/D5/0547** [397]

12556

12557

12558 On Page: 1855 Line: 59289 Section: setbuf()

12559

12560 In the APPLICATION USAGE section, add a new paragraph at the end of
12561 the section:

12562

12563 Since errno is not required to be unchanged on success, in order to
12564 correctly detect and possibly recover from errors, applications should
12565 use setvbuf() instead of setbuf().

12566

12567 *Rationale:* Austin Group Defect Report(s) applied: 397.12568 See <http://austingroupbugs.net/view.php?id=397>

12569

12570

12571 **Change Number: XSH/TC1/D5/0548** [14]

12572

12573

12574 On Page: 1855 Line: 59295 Section: setbuf()

12575

12576 In the SEE ALSO section, add a reference to XSH Section 2.5.

12577

12578 *Rationale:* Austin Group Defect Report(s) applied: 14.12579 See <http://austingroupbugs.net/view.php?id=14>

12580

12581

12582

12583 **Change Number: XSH/TC1/D5/0549** [167]

12584

12585

12586 On Page: 1857 Line: 59347 Section: setenv()

12587

12588 In the DESCRIPTION section, remove the text:

12589

12590 If the application modifies environ or the pointers to which it points,
12591 the behavior of setenv() is undefined.

12592

12593 *Rationale:* Austin Group Defect Report(s) applied: 167.12594 See <http://austingroupbugs.net/view.php?id=167>

12595

12596 The text relating to undefined behavior on modification
12597 of environ is removed as this is now covered by general text added in
12598 XBD 8.1.

12599

12600

12601 **Change Number: XSH/TC1/D5/0550** [185]

12602

12603

12604 On Page: 1857 Line: 59356 Section: setenv()

12605

12606 In the ERRORS section, change from:

12607

12608 [EINVAL] The name argument is a null pointer, points to an empty string,
12609 or points to a string containing an '=' character.

12610

12611 to:

12612

12613 [EINVAL] The envname argument points to an empty string or points to a
12614 string containing an '=' character.

12615

12616 *Rationale:* Austin Group Defect Report(s) applied: 185.

12616 See <http://austingroupbugs.net/view.php?id=185>
12617
12618
12619 **Change Number: XSH/TC1/D5/0551 [167]**
12620
12621
12622 On Page: 1858 Line: 59381 Section: `setenv()`
12623
12624 In the RATIONALE section, add to the end of the section:
12625
12626 See also the RATIONALE section in `getenv()`.
12627
12628 *Rationale:* Austin Group Defect Report(s) applied: 167.
12629 See <http://austingroupbugs.net/view.php?id=167>
12630
12631
12632 **Change Number: XSH/TC1/D5/0552 [38]**
12633
12634
12635 On Page: 1858 Line: 59385 Section: `setenv()`
12636
12637 In the SEE ALSO section, add "`putenv()`".
12638
12639 *Rationale:* Austin Group Defect Report(s) applied: 38.
12640 See <http://austingroupbugs.net/view.php?id=38>
12641
12642
12643 **Change Number: XSH/TC1/D5/0553 [302]**
12644
12645
12646 On Page: 1868 Line: 59590 Section: `setlocale()`
12647
12648 In the DESCRIPTION section, change from:
12649
12650 The `setlocale()` function selects the appropriate piece of the locale of
12651 the process, as specified by the category and locale arguments, and may
12652 be used to change or query the entire locale of the process or portions
12653 thereof. The value `LC_ALL` for category names the entire locale of the
12654 process; other values for category name only a part of the locale of
12655 the process:
12656
12657 to:
12658
12659 The `setlocale()` function selects the appropriate piece of the global
12660 locale, as specified by the category and locale arguments, and can be
12661 used to change or query the entire global locale or portions thereof. The
12662 value `LC_ALL` for category names the entire global locale; other values
12663 for category name only a part of the global locale:
12664
12665 *Rationale:* Austin Group Defect Report(s) applied: 302.
12666 See <http://austingroupbugs.net/view.php?id=302>
12667 These changes were overlooked during the revision when
12668 per-thread locales were added.
12669
12670
12671 **Change Number: XSH/TC1/D5/0554 [303]**
12672
12673
12674 On Page: 1868 Line: 59597 Section: `setlocale()`
12675
12676 In the DESCRIPTION section, replace the description of `LC_MESSAGES` with:
12677
12678 `LC_MESSAGES` Affects the affirmative and negative response expressions

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

12679 returned by `nl_langinfo()` and the way message catalogs are located. It
 12680 may also affect the behavior of functions that return or write message
 12681 strings.
 12682
 12683 (All within the CX shading; no XSI shading anywhere.)
 12684
 12685 *Rationale:* Austin Group Defect Report(s) applied: 303.
 12686 See <http://austingroupbugs.net/view.php?id=303>
 12687
 12688
 12689 **Change Number: XSH/TC1/D5/0555 [302]**
 12690
 12691
 12692 On Page: 1868 Line: 59608 Section: `setlocale()`
 12693
 12694 In the DESCRIPTION section, change from:
 12695
 12696 If `setlocale()` is not invoked, the POSIX locale is the default at entry
 12697 to `main()`.
 12698
 12699 to:
 12700
 12701 The POSIX locale is the default global locale at entry to `main()`.
 12702
 12703 *Rationale:* Austin Group Defect Report(s) applied: 302.
 12704 See <http://austingroupbugs.net/view.php?id=302>
 12705 These changes were overlooked during the revision when
 12706 per-thread locales were added.
 12707
 12708
 12709 **Change Number: XSH/TC1/D5/0556 [302]**
 12710
 12711
 12712 On Page: 1868 Line: 59615 Section: `setlocale()`
 12713
 12714 In the DESCRIPTION section, change from:
 12715
 12716 Used to direct `setlocale()` to query the current internationalized
 12717 environment and return the name of the locale.
 12718
 12719 to:
 12720
 12721 Directs `setlocale()` to query the current global locale setting and
 12722 return the name of the locale if category is not `LC_ALL`, or a string
 12723 which encodes the locale name(s) for all of the individual categories
 12724 if category is `LC_ALL`.
 12725
 12726 *Rationale:* Austin Group Defect Report(s) applied: 302.
 12727 See <http://austingroupbugs.net/view.php?id=302>
 12728 These changes were overlooked during the revision when
 12729 per-thread locales were added.
 12730
 12731
 12732 **Change Number: XSH/TC1/D5/0557 [302]**
 12733
 12734
 12735 On Page: 1868 Line: 59617 Section: `setlocale()`
 12736
 12737 In the DESCRIPTION section, change from:
 12738
 12739 Setting all of the categories of the locale of the process is similar
 12740 to successively setting each individual category of the locale of the
 12741 process, except that all error checking is done before any actions

12742 are performed. To set all the categories of the locale of the process,
12743 `setlocale()` is invoked as:
12744
12745 to:
12746
12747 Setting all of the categories of the global locale is similar to
12748 successively setting each individual category of the global locale, except
12749 that all error checking is done before any actions are performed. To set
12750 all the categories of the global locale, `setlocale()` can be invoked as:
12751
12752 *Rationale:* Austin Group Defect Report(s) applied: 302.
12753 See <http://austingroupbugs.net/view.php?id=302>
12754 These changes were overlooked during the revision when
12755 per-thread locales were added.
12756
12757
12758 **Change Number: XSH/TC1/D5/0558 [302]**
12759
12760
12761 On Page: 1869 Line: 59625 Section: `setlocale()`
12762
12763 In the DESCRIPTION section, change from:
12764
12765 ... the locale of the process ...
12766
12767 to:
12768 ... the global locale ...
12769
12770
12771 *Rationale:* Austin Group Defect Report(s) applied: 302.
12772 See <http://austingroupbugs.net/view.php?id=302>
12773 These changes were overlooked during the revision when
12774 per-thread locales were added.
12775
12776
12777 **Change Number: XSH/TC1/D5/0559 [302]**
12778
12779
12780 On Page: 1869 Line: 59629 Section: `setlocale()`
12781
12782 In the DESCRIPTION section, change from:
12783
12784 The locale state is common to all threads within a process.
12785
12786 to:
12787
12788 [CX]The global locale established using `setlocale()` shall only be used
12789 in threads for which no current locale has been set using `uselocale()`
12790 or whose current locale has been set to the global locale using
12791 `uselocale(LC_GLOBAL_LOCALE).`[/CX]
12792
12793 *Rationale:* Austin Group Defect Report(s) applied: 302.
12794 See <http://austingroupbugs.net/view.php?id=302>
12795 These changes were overlooked during the revision when
12796 per-thread locales were added.
12797
12798
12799 **Change Number: XSH/TC1/D5/0560 [288]**
12800
12801
12802 On Page: 1869 Line: 59629 Section: `setlocale()`
12803
12804 In the DESCRIPTION section, add a new paragraph to the end of the section:

12805
 12806 The implementation shall behave as if no function defined in this volume
 12807 of POSIX.1-200x calls `setlocale()`.
 12808
 12809 *Rationale:* Austin Group Defect Report(s) applied: 288.
 12810 See <http://austingroupbugs.net/view.php?id=288>
 12811
 12812
 12813 **Change Number: XSH/TC1/D5/0561 [302]**
 12814
 12815
 12816 On Page: 1869 Line: 59632 Section: `setlocale()`
 12817
 12818 In the RETURN VALUE section, change from:
 12819
 12820 ... the locale of the process is not changed.
 12821
 12822 to:
 12823
 12824 ... the global locale shall not be changed.
 12825
 12826 *Rationale:* Austin Group Defect Report(s) applied: 302.
 12827 See <http://austingroupbugs.net/view.php?id=302>
 12828 These changes were overlooked during the revision when
 12829 per-thread locales were added.
 12830
 12831
 12832 **Change Number: XSH/TC1/D5/0562 [302]**
 12833
 12834
 12835 On Page: 1869 Line: 59634 Section: `setlocale()`
 12836
 12837 In the RETURN VALUE section, change from:
 12838
 12839 A null pointer for locale causes `setlocale()` to return a pointer to
 12840 the string associated with the category for the current locale of the
 12841 process. The locale of the process shall not be changed.
 12842
 12843 to:
 12844
 12845 A null pointer for locale shall cause `setlocale()` to return a pointer
 12846 to the string associated with the specified category for the current
 12847 global locale. The global locale shall not be changed.
 12848
 12849 *Rationale:* Austin Group Defect Report(s) applied: 302.
 12850 See <http://austingroupbugs.net/view.php?id=302>
 12851 These changes were overlooked during the revision when
 12852 per-thread locales were added.
 12853
 12854
 12855 **Change Number: XSH/TC1/D5/0563 [302]**
 12856
 12857
 12858 On Page: 1869 Line: 59637 Section: `setlocale()`
 12859
 12860 In the RETURN VALUE section, change from:
 12861
 12862 ... the locale of the process ...
 12863
 12864 to:
 12865
 12866 ... the global locale ...
 12867

12868 *Rationale:* Austin Group Defect Report(s) applied: 302.
12869 See <http://austingroupbugs.net/view.php?id=302>
12870 These changes were overlooked during the revision when
12871 per-thread locales were added.
12872
12873
12874 **Change Number: XSH/TC1/D5/0564** [302]
12875
12876
12877 On Page: 1869 Line: 59645 Section: `setlocale()`
12878
12879 In the APPLICATION USAGE section, change from:
12880
12881 ... the locale of the process ...
12882
12883 to:
12884
12885 ... the global locale ...
12886
12887 *Rationale:* Austin Group Defect Report(s) applied: 302.
12888 See <http://austingroupbugs.net/view.php?id=302>
12889 These changes were overlooked during the revision when
12890 per-thread locales were added.
12891
12892
12893 **Change Number: XSH/TC1/D5/0565** [302]
12894
12895
12896 On Page: 1869 Line: 59649 Section: `setlocale()`
12897
12898 In the APPLICATION USAGE section, change from:
12899
12900 Internationalized programs must call `setlocale()` to initiate a specific
12901 language operation. This can be done by calling `setlocale()` as follows:
12902
12903 to:
12904
12905 Internationalized programs can initiate language operation according
12906 to environment variable settings (see XBD Section 8.2) by calling
12907 `setlocale()` as follows:
12908
12909 *Rationale:* Austin Group Defect Report(s) applied: 302.
12910 See <http://austingroupbugs.net/view.php?id=302>
12911 These changes were overlooked during the revision when
12912 per-thread locales were added.
12913
12914
12915 **Change Number: XSH/TC1/D5/0566** [302]
12916
12917
12918 On Page: 1869 Line: 59655 Section: `setlocale()`
12919
12920 In the RATIONALE section, insert a new paragraph at the beginning of the
12921 section:
12922
12923 References to the international environment or locale in the following
12924 text relate to the global locale for the process. This can be overridden
12925 for individual threads using `uselocale()`.
12926
12927 *Rationale:* Austin Group Defect Report(s) applied: 302.
12928 See <http://austingroupbugs.net/view.php?id=302>
12929 These changes were overlooked during the revision when
12930 per-thread locales were added.

12931
 12932
 12933 **Change Number: XSH/TC1/D5/0567 [288]**
 12934
 12935
 12936 On Page: 1869 Line: 59637 Section: setlocale()
 12937
 12938 In the RETURN VALUE section, change to:
 12939
 12940 The application shall not modify the string returned. [CX]The returned
 12941 string pointer might be invalidated or[/CX] the string content might be
 12942 overwritten by a subsequent call to setlocale().
 12943
 12944 *Rationale:* Austin Group Defect Report(s) applied: 288.
 12945 See <http://austingroupbugs.net/view.php?id=288>
 12946
 12947
 12948 **Change Number: XSH/TC1/D5/0568 [288]**
 12949
 12950
 12951 On Page: 1869 Line: 59653 Section: setlocale()
 12952
 12953 In the APPLICATION USAGE section, add a new paragraph to the end of
 12954 the section:
 12955
 12956 In order to make use of different locale settings while multiple
 12957 threads are running, applications should use uselocale() in preference
 12958 to setlocale().
 12959
 12960 *Rationale:* Austin Group Defect Report(s) applied: 288.
 12961 See <http://austingroupbugs.net/view.php?id=288>
 12962
 12963
 12964 **Change Number: XSH/TC1/D5/0569 [303]**
 12965
 12966
 12967 On Page: 1871 Line: 59717 Section: setlocale()
 12968
 12969 In the SEE ALSO section, add catopen(), perror(), and psignal().
 12970
 12971 *Rationale:* Austin Group Defect Report(s) applied: 303.
 12972 See <http://austingroupbugs.net/view.php?id=303>
 12973
 12974
 12975 **Change Number: XSH/TC1/D5/0570 [421]**
 12976
 12977
 12978 On Page: 1887 Line: 60039 Section: setsid()
 12979
 12980 In the RETURN VALUE section, change from:
 12981
 12982 Otherwise, it shall return (pid_t)-1 and ...
 12983
 12984 to:
 12985
 12986 Otherwise, it shall return -1 and ...
 12987
 12988 *Rationale:* Austin Group Defect Report(s) applied: 421.
 12989 See <http://austingroupbugs.net/view.php?id=421>
 12990
 12991
 12992 **Change Number: XSH/TC1/D5/0571 [369]**
 12993

12994
12995 On Page: 1889 Line: 60092 Section: setsockopt()
12996
12997 Insert a new sentence between the two sentences in the last paragraph
12998 of the DESCRIPTION:
12999
13000 If option_name is equal to SO_RCVTIMEO or SO_SNDTIMEO and the
13001 implementation supports setting the option, it is unspecified whether
13002 the struct timeval pointed to by option_value is stored as provided by
13003 this function or is rounded up to align with the resolution of the clock
13004 being used.
13005
13006 *Rationale:* Austin Group Defect Report(s) applied: 369.
13007 See <http://austingroupbugs.net/view.php?id=369>
13008
13009
13010 **Change Number: XSH/TC1/D5/0572 [345]**
13011
13012
13013 On Page: 1905 Line: 60590 Section: shmatt()
13014
13015 In the RETURN VALUE section, after L60590, add a sentence:
13016
13017 Also, the shm_atime timestamp shall be set to the current time, as
13018 described in XSH Section 2.7.1.
13019
13020 *Rationale:* Austin Group Defect Report(s) applied: 345.
13021 See <http://austingroupbugs.net/view.php?id=345>
13022
13023
13024 **Change Number: XSH/TC1/D5/0573 [345]**
13025
13026
13027 On Page: 1907 Line: 60648 Section: shmattl()
13028
13029 In the DESCRIPTION section [IPC_SET], after L60648, add a sentence:
13030
13031 Also, the shm_ctime timestamp shall be set to the current time, as
13032 described in XSH Section 2.7.1.
13033
13034 *Rationale:* Austin Group Defect Report(s) applied: 345.
13035 See <http://austingroupbugs.net/view.php?id=345>
13036
13037
13038 **Change Number: XSH/TC1/D5/0574 [345]**
13039
13040
13041 On Page: 1909 Line: 60710 Section: shmattd()
13042
13043 In the RETURN VALUE section, after L60710, add a sentence:
13044
13045 Also, the shm_dtime timestamp shall be set to the current time, as
13046 described in XSH Section 2.7.1.
13047
13048 *Rationale:* Austin Group Defect Report(s) applied: 345.
13049 See <http://austingroupbugs.net/view.php?id=345>
13050
13051
13052 **Change Number: XSH/TC1/D5/0575 [345]**
13053
13054
13055 On Page: 1911 Line: 60760 Section: shmget()
13056

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

13057 In the DESCRIPTION section, change from:
13058
13059 ... set equal to the current time
13060
13061 to:
13062
13063 ... set to the current time, as described in XSH Section 2.7.1
13064
13065 *Rationale:* Austin Group Defect Report(s) applied: 345.
13066 See <http://austingroupbugs.net/view.php?id=345>
13067
13068
13069 **Change Number: XSH/TC1/D5/0576 [363]**
13070
13071
13072 On Page: 1911 Line: 60774-60776 Section: shmget()
13073
13074 In the ERRORS section, [EINVAL] error:
13075
13076 Remove the words "and size is not 0" from the description of [EINVAL].
13077
13078 *Rationale:* Austin Group Defect Report(s) applied: 363.
13079 See <http://austingroupbugs.net/view.php?id=363>
13080 The shmget() function shall fail with error [EINVAL] if "No
13081 shared memory segment is to be created and a shared memory segment exists
13082 for key but the size of the segment associated with it is less than size
13083 and size is not 0". However, if the size of the segment is less than
13084 the size argument, then the size argument is guaranteed to be non-zero,
13085 as both are unsigned values. Thus, the last condition is redundant and
13086 should be removed.
13087
13088
13089 **Change Number: XSH/TC1/D5/0577 [344]**
13090
13091
13092 On Page: 1912 Line: 60797 Section: shmget()
13093
13094 In the SEE ALSO section, add ftok() prior to shmat().
13095
13096 *Rationale:* Austin Group Defect Report(s) applied: 344.
13097 See <http://austingroupbugs.net/view.php?id=344>
13098
13099
13100 **Change Number: XSH/TC1/D5/0578 [66]**
13101
13102
13103 On Page: 1919 Line: 61038 Section: sigaction()
13104
13105 In the APPLICATION USAGE section, remove the paragraph:
13106
13107 If the signal occurs other than as the result of calling abort(), kill(),
13108 or raise(), the behavior is undefined if the signal handler calls any
13109 function in the standard library other than one of the functions listed
13110 in the table of sync-signal-safe functions in Section 2.4.3, or refers
13111 to any object other than errno with static storage duration other than by
13112 assigning a value to a static storage duration variable of type volatile
13113 sig_atomic_t. Unless all signal handlers have errno set on return as it
13114 was on entry, the value of errno is unspecified.
13115
13116 *Rationale:* Austin Group Defect Report(s) applied: 66.
13117 See <http://austingroupbugs.net/view.php?id=66>
13118
13119

13120 **Change Number: XSH/TC1/D5/0579** [140]
 13121
 13122
 13123 On Page: 1919 Line: 61064 Section: sigaction()
 13124
 13125 In the APPLICATION USAGE section, add a new a paragraph at the end of
 13126 the section:
 13127
 13128 See also the rationale for Realtime Signal Generation and Delivery in
 13129 Section B.2.4.2.
 13130
 13131 *Rationale:* Austin Group Defect Report(s) applied: 140.
 13132 See <http://austingroupbugs.net/view.php?id=140>
 13133
 13134
 13135 **Change Number: XSH/TC1/D5/0580** [275]
 13136
 13137
 13138 On Page: 1937 Line: 61633 Section: signal()
 13139
 13140 In the DESCRIPTION section, change from:
 13141
 13142 If the signal occurs as the result of calling the abort(), raise(),
 13143 [CX]kill(), pthread_kill(), or sigqueue()[/CX] function, the signal
 13144 handler shall not call the raise() function.
 13145
 13146 to:
 13147
 13148 The C Standard places a restriction on applications relating to the use of
 13149 raise() from signal handlers. [CX]This restriction does not apply to POSIX
 13150 applications, as POSIX.1-2008 requires raise() to be async-signal-safe
 13151 (see Section 2.4.3 Signal Actions).[/CX]
 13152
 13153 *Rationale:* Austin Group Defect Report(s) applied: 275.
 13154 See <http://austingroupbugs.net/view.php?id=275>
 13155
 13156
 13157 **Change Number: XSH/TC1/D5/0581** [66]
 13158
 13159
 13160 On Page: 1937 Line: 61635 Section: signal()
 13161
 13162 In the DESCRIPTION section, change from:
 13163
 13164 If the signal occurs other than as the result of calling abort(),
 13165 raise(), [CX]kill(), pthread_kill(), or sigqueue()[/CX], the behavior
 13166 is undefined if the signal handler refers to any object with static
 13167 storage duration other than by assigning a value to an object declared as
 13168 volatile sig_atomic_t, or if the signal handler calls any function in the
 13169 standard library other than one of the functions listed in Section 2.4.
 13170 Furthermore, if such a call fails, the value of errno is unspecified.
 13171
 13172 to:
 13173
 13174 [CX]If the process is multi-threaded[/CX], or if the process is single
 13175 threaded and a signal handler is executed other than as the result of:
 13176
 13177 * The process calling abort(), raise(), [CX]kill(), pthread_kill(),
 13178 or sigqueue()[/CX] to generate a signal that is not blocked
 13179
 13180 * [CX]A pending signal being unblocked and being delivered before the
 13181 call that unblocked it returns[/CX]
 13182

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

13183 the behavior is undefined if the signal handler refers to any object
 13184 [CX]other than errno[/CX] with static storage duration other than by
 13185 assigning a value to an object declared as volatile sig_atomic_t, or if
 13186 the signal handler calls any function defined in this standard other than
 13187 [CX]one of the functions listed in Section 2.4[/CX].

13188

13189 *Rationale:* Austin Group Defect Report(s) applied: 66.

13190 See <http://austingroupbugs.net/view.php?id=66>

13191

13192

13193 **Change Number: XSH/TC1/D5/0582** [105]

13194

13195

13196 On Page: 1937 Line: 61644 Section: signal()

13197

13198 In the DESCRIPTION section, add a new paragraph at the end of the
 13199 DESCRIPTION:

13200

13201 The signal() function shall not change the setting of errno if successful.

13202

13203 *Rationale:* Austin Group Defect Report(s) applied: 105.

13204 See <http://austingroupbugs.net/view.php?id=105>

13205

13206

13207 **Change Number: XSH/TC1/D5/0583** [392]

13208

13209

13210 On Page: 1952 Line: 62027 Section: sigtimedwait()

13211

13212 In the APPLICATION USAGE section, add a new paragraph to the end of the
 13213 section:

13214

13215 Note that in order to ensure generated signals are queued and signal
 13216 values passed to sigqueue() are available in si_value, applications
 13217 which use sigwaitinfo() or sigtimedwait() need to set the SA_SIGINFO
 13218 flag for each signal in the set (see 2.4). This means setting each
 13219 signal to be handled by a three argument signal catching function,
 13220 even if the handler will never be called. It is not possible (portably)
 13221 to set a signal handler to SIG_DFL while setting the SA_SIGINFO flag,
 13222 because assigning to the sa_handler member of struct sigaction instead
 13223 of the sa_sigaction member would result in undefined behavior, and
 13224 SIG_DFL need not be assignment compatible with sa_sigaction. Even if
 13225 an assignment of SIG_DFL to sa_sigaction is accepted by the compiler,
 13226 the implementation need not treat this value as special - it could just
 13227 be taken as the address of a signal catching function.

13228

13229 *Rationale:* Austin Group Defect Report(s) applied: 392.

13230 See <http://austingroupbugs.net/view.php?id=392>

13231

13232

13233 **Change Number: XSH/TC1/D5/0584** [76]

13234

13235

13236 On Page: 1955 Line: 62165 Section: sigwait()

13237

13238 In the RATIONALE section, change from:

13239

13240 After some consideration, threads were allowed to use semaphores and
 13241 sem_post() was defined to be async-signal and async-cancel-safe.

13242

13243 to:

13244

13245 After some consideration, threads were allowed to use semaphores and

13246 sem_post() was defined to be async-signal-safe.
13247
13248 *Rationale:* Austin Group Defect Report(s) applied: 76.
13249 See <http://austingroupbugs.net/view.php?id=76>
13250
13251
13252 **Change Number: XSH/TC1/D5/0585 [68]**
13253
13254
13255 On Page: 1958 Line: 62216 Section: sin()
13256
13257 In the RETURN VALUE section, change from:
13258
13259 [MX]If x is subnormal, a range error may occur and x should be
13260 returned.[/MX]
13261
13262 to:
13263
13264 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
13265 be returned.[/MXX]
13266
13267 [MX]If x is not returned, sin(), sinf(), and sinl() shall return
13268 an implementation-defined value no greater in magnitude than DBL_MIN,
13269 FLT_MIN, and LDBL_MIN, respectively.[/MX]
13270
13271 *Rationale:* Austin Group Defect Report(s) applied: 68.
13272 See <http://austingroupbugs.net/view.php?id=68>
13273
13274
13275 **Change Number: XSH/TC1/D5/0586 [320]**
13276
13277
13278 On Page: 1958 Line: 62217 Section: sin()
13279
13280 In the RETURN VALUE section, change from:
13281
13282 ... and either a NaN (if supported), or an implementation-defined
13283 value shall be returned.
13284
13285 to:
13286
13287 ... and a NaN shall be returned.
13288
13289 *Rationale:* Austin Group Defect Report(s) applied: 320.
13290 See <http://austingroupbugs.net/view.php?id=320>
13291
13292
13293 **Change Number: XSH/TC1/D5/0587 [68]**
13294
13295
13296 On Page: 1960 Line: 62285 Section: sinh()
13297
13298 In the RETURN VALUE section, change from:
13299
13300 [MX]If x is subnormal, a range error may occur and x should be
13301 returned.[/MX]
13302
13303 to:
13304
13305 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
13306 be returned.[/MXX]
13307
13308 [MX]If x is not returned, sinh(), sinhf(), and sinhl() shall return

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

13309 an implementation-defined value no greater in magnitude than DBL_MIN,
 13310 FLT_MIN, and LDBL_MIN, respectively. [/MX]

13311
 13312 *Rationale:* Austin Group Defect Report(s) applied: 68.
 13313 See <http://austingroupbugs.net/view.php?id=68>

13314
 13315
 13316 **Change Number: XSH/TC1/D5/0588 [320]**

13317
 13318
 13319 On Page: 1973 Line: 62659 Section: sqrt()
 13320

13321 In the RETURN VALUE section, change from:
 13322
 13323 ... and either a NaN (if supported), or an implementation-defined
 13324 value shall be returned.

13325
 13326 to:
 13327
 13328 ... and a NaN shall be returned.

13329
 13330 *Rationale:* Austin Group Defect Report(s) applied: 320.
 13331 See <http://austingroupbugs.net/view.php?id=320>

13332
 13333
 13334 **Change Number: XSH/TC1/D5/0589 [302]**

13335
 13336 On Page: 1985 Line: 62813 Section: strcasecmp()
 13337

13338
 13339 In the DESCRIPTION section, change from:
 13340
 13341 ... current locale of the process ...

13342
 13343 to:
 13344
 13345 ... current locale ...

13346
 13347 *Rationale:* Austin Group Defect Report(s) applied: 302.
 13348 See <http://austingroupbugs.net/view.php?id=302>
 13349 These changes were overlooked during the revision when
 13350 per-thread locales were added.

13351
 13352
 13353 **Change Number: XSH/TC1/D5/0590 [294]**

13354
 13355 On Page: 1985 Line: 62817 Section: strcasecmp()
 13356

13357
 13358 In the DESCRIPTION section, change from:
 13359

13360 When the LC_CTYPE category of the current locale is from the POSIX locale,
 13361 strcasecmp() and strncasecmp() shall ...

13362
 13363 to:
 13364
 13365 When the LC_CTYPE category of the locale being used is from the POSIX
 13366 locale, these functions shall ...

13367
 13368 *Rationale:* Austin Group Defect Report(s) applied: 294.
 13369 See <http://austingroupbugs.net/view.php?id=294>

13370
 13371

13372 **Change Number: XSH/TC1/D5/0591** [283]
13373
13374
13375 On Page: 1985 Line: 62819 Section: strcasecmp()
13376
13377 In the DESCRIPTION section, add a new paragraph to the end of the section:
13378
13379 The behavior is undefined if the locale argument to strcasecmp_l() or
13380 strncasecmp_l() is the special locale object LC_GLOBAL_LOCALE or is not
13381 a valid locale object handle.
13382
13383 *Rationale:* Austin Group Defect Report(s) applied: 283.
13384 See <http://austingroupbugs.net/view.php?id=283>
13385
13386
13387 **Change Number: XSH/TC1/D5/0592** [283]
13388
13389
13390 On Page: 1985 Line: 62829 Section: strcasecmp()
13391
13392 In the ERRORS section, change from:
13393
13394 The strcasecmp_l() and strncasecmp_l() functions may fail if:
13395
13396 [EINVAL] locale is not a valid locale object handle.
13397
13398 to:
13399
13400 No errors are defined.
13401
13402 *Rationale:* Austin Group Defect Report(s) applied: 283.
13403 See <http://austingroupbugs.net/view.php?id=283>
13404
13405
13406 **Change Number: XSH/TC1/D5/0593** [283]
13407
13408
13409 On Page: 1991 Line: 62993 Section: strcoll()
13410
13411 In the DESCRIPTION section, add a new paragraph to the end of the section:
13412
13413 [CX]The behavior is undefined if the locale argument to strcoll_l()
13414 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
13415 object handle.[/CX]
13416
13417 *Rationale:* Austin Group Defect Report(s) applied: 283.
13418 See <http://austingroupbugs.net/view.php?id=283>
13419
13420
13421 **Change Number: XSH/TC1/D5/0594** [283]
13422
13423
13424 On Page: 1991 Line: 63007 Section: strcoll()
13425
13426 In the ERRORS section, delete:
13427
13428 The strcoll_l() function may fail if:
13429
13430 [EINVAL] Locale is not a valid locale object handle.
13431
13432 *Rationale:* Austin Group Defect Report(s) applied: 283.
13433 See <http://austingroupbugs.net/view.php?id=283>
13434

13435

13436 **Change Number: XSH/TC1/D5/0595 [75]**

13437

13438

13439 On Page: 1999 Line: 63237 Section: strerror()

13440

13441 In the DESCRIPTION section, change from:

13442

13443 The string pointed to shall not be modified by the application.

13444 The string may be overwritten by a subsequent call to strerror().

13445 [CX]The string may be overwritten by a subsequent call to strerror_l()

13446 in the same thread.[/CX]

13447

13448 to:

13449

13450 The application shall not modify the string returned. [CX]The returned

13451 string pointer might be invalidated or[/CX] the string content might be

13452 overwritten by a subsequent call to strerror(), [CX]or by a subsequent

13453 call to strerror_l() in the same thread[/CX].

13454

13455 *Rationale:* Austin Group Defect Report(s) applied: 75.13456 See <http://austingroupbugs.net/view.php?id=75>

13457

13458

13459 **Change Number: XSH/TC1/D5/0596 [447]**

13460

13461

13462 On Page: 1999 Line: 63244 Section: strerror()

13463

13464 In the DESCRIPTION section, change from:

13465

13466 The strerror() and strerror_l() functions shall not change the setting

13467 of errno if successful.

13468

13469 Since no return value is reserved to indicate an error, an application

13470 wishing to check for error situations should set errno to 0, then call

13471 strerror(), then check errno.

13472

13473 to:

13474

13475 The strerror() and strerror_l() functions shall not change the setting

13476 of errno if successful.

13477

13478 Since no return value is reserved to indicate an error of strerror(), an

13479 application wishing to check for error situations should set errno to 0,

13480 then call strerror(), then check errno. Similarly, since strerror_l()

13481 is required to return a string for some errors, an application wishing

13482 to check for all error situations should set errno to 0, then call

13483 strerror_l(), then check errno.

13484

13485 *Rationale:* Austin Group Defect Report(s) applied: 447.13486 See <http://austingroupbugs.net/view.php?id=447>

13487

13488

13489 **Change Number: XSH/TC1/D5/0597 [382,428]**

13490

13491

13492 On Page: 1999 Line: 63253 Section: strerror()

13493

13494 In the DESCRIPTION section, change from (retaining the CX shading):

13495

13496 If the value of errno is a valid error number, the message string shall

13497 indicate what error occurred; otherwise, if these functions complete

13498 successfully, the message string shall indicate that an unknown error
13499 occurred.
13500
13501 to:
13502
13503 If the value of `errno` is a valid error number, the message string
13504 shall indicate what error occurred; if the value of `errno` is zero,
13505 the message string shall either be an empty string or indicate that no
13506 error occurred; otherwise, if these functions complete successfully,
13507 the message string shall indicate that an unknown error occurred.
13508
13509 *Rationale:* Austin Group Defect Report(s) applied: 382,428.
13510 See <http://austingroupbugs.net/view.php?id=382>
13511 See <http://austingroupbugs.net/view.php?id=428>
13512 While the standard does not currently require it, the intent
13513 was that `strerror(0)` should be required to succeed, and there are existing
13514 applications that rely on this behavior.
13515
13516 **Change Number: XSH/TC1/D5/0598 [283]**
13517
13518
13519
13520 On Page: 1999 Line: 63255 Section: `strerror()`
13521
13522 In the DESCRIPTION section, add a new paragraph to the end of the section:
13523
13524 [CX]The behavior is undefined if the locale argument to `strerror_l()`
13525 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
13526 object handle.[/CX]
13527
13528 *Rationale:* Austin Group Defect Report(s) applied: 283.
13529 See <http://austingroupbugs.net/view.php?id=283>
13530
13531
13532 **Change Number: XSH/TC1/D5/0599 [382,428]**
13533
13534
13535 On Page: 2000 Line: 63267 Section: `strerror()`
13536
13537 In the ERRORS section for the first [EINVAL] error, change from:
13538
13539 ... is not a valid error number.
13540
13541 to:
13542
13543 ... is neither a valid error number nor zero.
13544
13545 *Rationale:* Austin Group Defect Report(s) applied: 382,428.
13546 See <http://austingroupbugs.net/view.php?id=382>
13547 See <http://austingroupbugs.net/view.php?id=428>
13548 While the standard does not currently require it, the intent
13549 was that `strerror(0)` should be required to succeed, and there are existing
13550 applications that rely on this behavior.
13551
13552
13553 **Change Number: XSH/TC1/D5/0600 [283]**
13554
13555
13556 On Page: 2000 Line: 63268 Section: `strerror()`
13557
13558 In the ERRORS section, delete:
13559
13560 The `strerror_l()` function may fail if:

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

13561
 13562 [EINVAL] locale is not a valid locale object handle.
 13563
 13564 *Rationale:* Austin Group Defect Report(s) applied: 283.
 13565 See <http://austingroupbugs.net/view.php?id=283>
 13566
 13567
 13568 **Change Number: XSH/TC1/D5/0601** [382,428]
 13569
 13570
 13571 On Page: 2000 Line: 63303 Section: strerror()
 13572
 13573 In the RATIONALE section, add a new paragraph to the end of the section:
 13574
 13575 Some applications rely on being able to set errno to 0 before calling
 13576 a function with no reserved value to indicate an error, then call
 13577 strerror(errno) afterwards to detect whether an error occurred
 13578 (because errno changed) or indicate success (because errno remained
 13579 zero). This usage pattern requires that strerror(0) succeed with useful
 13580 results. Previous versions of the standard did not specify the behavior
 13581 when errnum is zero.
 13582
 13583 *Rationale:* Austin Group Defect Report(s) applied: 382,428.
 13584 See <http://austingroupbugs.net/view.php?id=382>
 13585 See <http://austingroupbugs.net/view.php?id=428>
 13586 While the standard does not currently require it, the intent
 13587 was that strerror(0) should be required to succeed, and there are existing
 13588 applications that rely on this behavior.
 13589
 13590
 13591 **Change Number: XSH/TC1/D5/0602** [302]
 13592
 13593
 13594 On Page: 2004 Line: 63413 Section: strfmon()
 13595
 13596 change from:
 13597
 13598 ... the locale of the process..
 13599
 13600 to:
 13601
 13602 ... the current locale..
 13603
 13604 *Rationale:* Austin Group Defect Report(s) applied: 302.
 13605 See <http://austingroupbugs.net/view.php?id=302>
 13606 These changes were overlooked during the revision when
 13607 per-thread locales were added.
 13608
 13609
 13610 **Change Number: XSH/TC1/D5/0603** [283]
 13611
 13612
 13613 On Page: 2004 Line: 63418 Section: strfmon()
 13614
 13615 In the DESCRIPTION section, add a new paragraph to the end of the section:
 13616
 13617 The behavior is undefined if the locale argument to strfmon_l()
 13618 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 13619 object handle.
 13620
 13621 *Rationale:* Austin Group Defect Report(s) applied: 283.
 13622 See <http://austingroupbugs.net/view.php?id=283>
 13623

13624

13625 **Change Number: XSH/TC1/D5/0604** [283]

13626

13627

13628 On Page: 2004 Line: 63427 Section: strfmon()

13629

13630 In the ERRORS section, delete:

13631

13632 The strfmon_l() function may fail if:

13633

13634 [EINVAL] locale is not a valid locale object handle.

13635

13636 *Rationale:* Austin Group Defect Report(s) applied: 283.13637 See <http://austingroupbugs.net/view.php?id=283>

13638

13639

13640 **Change Number: XSH/TC1/D5/0605** [283]

13641

13642

13643 On Page: 2011 Line: 63682 Section: strftime()

13644

13645 In the DESCRIPTION section, add a new paragraph to the end of the section:

13646

13647 [CX]The behavior is undefined if the locale argument to strftime_l()

13648 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale

13649 object handle. [/CX]

13650

13651 *Rationale:* Austin Group Defect Report(s) applied: 283.13652 See <http://austingroupbugs.net/view.php?id=283>

13653

13654

13655 **Change Number: XSH/TC1/D5/0606** [283]

13656

13657

13658 On Page: 2011 Line: 63689 Section: strftime()

13659

13660 In the ERRORS section, change from:

13661

13662 The strftime_l() function may fail if:

13663

13664 [EINVAL] locale is not a valid locale object handle.

13665

13666 to:

13667

13668 No errors are defined.

13669

13670 *Rationale:* Austin Group Defect Report(s) applied: 283.13671 See <http://austingroupbugs.net/view.php?id=283>

13672

13673

13674 **Change Number: XSH/TC1/D5/0607** [193]

13675

13676

13677 On Page: 2014 Line: 63830 Section: strftime()

13678

13679 In the RATIONALE section, in the Conversion Specification column,
13680 change from:

13681

13682 "%05%Y"

13683

13684 to:

13685

13686 "%05Y"

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

13687
 13688 *Rationale:* Austin Group Defect Report(s) applied: 193.
 13689 See <http://austingroupbugs.net/view.php?id=193>
 13690
 13691
 13692 **Change Number: XSH/TC1/D5/0608 [193]**
 13693
 13694
 13695 On Page: 2014 Line: 63832 Section: strftime()
 13696
 13697 In the RATIONALE section, in the Conversion Specification column,
 13698 change the 'l' after the 'y' in %3C%yl+12345 to a column separator
 13699 (a ! in tbl formatting notation).
 13700
 13701 *Rationale:* Austin Group Defect Report(s) applied: 193.
 13702 See <http://austingroupbugs.net/view.php?id=193>
 13703
 13704
 13705 **Change Number: XSH/TC1/D5/0609 [75]**
 13706
 13707
 13708 On Page: 2032 Line: 64344 Section: strsignal()
 13709
 13710 In the DESCRIPTION section, change from:
 13711
 13712 The string pointed to shall not be modified by the application, but may
 13713 be overwritten by a subsequent call to strsignal() or setlocale().
 13714
 13715 to:
 13716
 13717 The application shall not modify the string returned. The returned
 13718 pointer might be invalidated or the string content might be overwritten
 13719 by a subsequent call to strsignal() or setlocale().
 13720
 13721 *Rationale:* Austin Group Defect Report(s) applied: 75.
 13722 See <http://austingroupbugs.net/view.php?id=75>
 13723
 13724
 13725 **Change Number: XSH/TC1/D5/0610 [302]**
 13726
 13727
 13728 On Page: 2036 Line: 64492 Section: strtod()
 13729
 13730 change from:
 13731
 13732 ... the locale of the process ...
 13733
 13734 to:
 13735
 13736 ... the current locale ...
 13737
 13738 *Rationale:* Austin Group Defect Report(s) applied: 302.
 13739 See <http://austingroupbugs.net/view.php?id=302>
 13740 These changes were overlooked during the revision when
 13741 per-thread locales were added.
 13742
 13743
 13744 **Change Number: XSH/TC1/D5/0611 [94]**
 13745
 13746
 13747 On Page: 2036 Line: 64498 Section: strtod()
 13748
 13749 In the DESCRIPTION section, change from:

13750

13751 If the subject sequence is empty or does not have the expected form,
13752 no conversion shall be performed; the value of str is stored in the ...

13753

13754 to:

13755

13756 If the subject sequence is empty or does not have the expected form,
13757 no conversion shall be performed; the value of nptr is stored in the ...

13758

13759 *Rationale:* Austin Group Defect Report(s) applied: 94.

13760 See <http://austingroupbugs.net/view.php?id=94>

13761 Correct a reference to the nptr argument in the case when

13762 no conversion is performed.

13763

13764

13765 **Change Number: XSH/TC1/D5/0612 [105]**

13766

13767

13768 On Page: 2036 Line: 64500 Section: strtod()

13769

13770 In the DESCRIPTION section, remove the CX shading from the paragraphs:

13771

13772 The strtod() function shall not change the setting of errno if successful.

13773

13774 Since 0 is returned on error and is also a valid return on success, an
13775 application wishing to check for error situations should set errno to 0,
13776 then call strtod(), strttof(), or strtold(), then check errno.

13777

13778 and then change the first paragraph from:

13779

13780 The strtod() function shall not change the setting of errno if successful.

13781

13782 to:

13783

13784 These functions shall not change the setting of errno if successful.

13785

13786 *Rationale:* Austin Group Defect Report(s) applied: 105.

13787 See <http://austingroupbugs.net/view.php?id=105>

13788

13789

13790 **Change Number: XSH/TC1/D5/0613 [453]**

13791

13792

13793 On Page: 2039 Line: 64605 Section: strtoumax()

13794

13795 In the RETURN VALUE section, change from:

13796

13797 If no conversion could be performed, zero shall be returned.

13798

13799 to:

13800

13801 If no conversion could be performed, 0 shall be returned [CX]and errno
13802 may be set to [EINVAL][CX]. [CX]If the value of base is not supported,
13803 0 shall be returned and errno shall be set to [EINVAL].[CX]

13804

13805 *Rationale:* Austin Group Defect Report(s) applied: 453.

13806 See <http://austingroupbugs.net/view.php?id=453>

13807

13808

13809 **Change Number: XSH/TC1/D5/0614 [453]**

13810

13811

13812 On Page: 2039 Line: 64610 Section: strtoumax()

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

13813
 13814 In the ERRORS section, change from:
 13815
 13816 These functions shall fail if:
 13817
 13818 [ERANGE] The value to be returned is not representable.
 13819
 13820 These functions may fail if:
 13821
 13822 [EINVAL] The value of base is not supported.
 13823
 13824 to:
 13825
 13826 These functions shall fail if:
 13827
 13828 [EINVAL] [CX]The value of base is not supported.[/CX]
 13829
 13830 [ERANGE] The value to be returned is not representable.
 13831
 13832 These functions may fail if:
 13833
 13834 [EINVAL] No conversion could be performed.
 13835
 13836 *Rationale:* Austin Group Defect Report(s) applied: 453.
 13837 See <http://austingroupbugs.net/view.php?id=453>
 13838
 13839
 13840 On Page: 2039 Line: 64617 Section: strtoumax()
 13841
 13842 In the APPLICATION USAGE section, change from:
 13843
 13844 None.
 13845
 13846 to:
 13847
 13848 Since the value of *endptr is unspecified if the value of base is not
 13849 supported, applications should either ensure that base has a supported
 13850 value (0 or between 2 and 36) before the call, or check for an [EINVAL]
 13851 error before examining *endptr.
 13852
 13853 *Rationale:* Austin Group Defect Report(s) applied: 453.
 13854 See <http://austingroupbugs.net/view.php?id=453>
 13855
 13856
 13857 **Change Number: XSH/TC1/D5/0615 [177]**
 13858
 13859
 13860 On Page: 2041 Line: 64700-64701 Section: strtok()
 13861
 13862 In the EXAMPLES section, change from:
 13863
 13864 Breaking a Line
 13865
 13866 The following example uses strtok() to break a line into two character
 13867 strings separated by any combination of <space>, <tab>, or <newline>
 13868 characters.
 13869
 13870 #include <string.h>
 13871 ...
 13872 struct element {
 13873 char *key;
 13874 char *data;
 13875 };

```
13876 ...
13877 char line[LINE_MAX];
13878 char *key, *data;
13879 ...
13880 key = strtok(line, " \n");
13881 data = strtok(NULL, " \n");
13882 ...
13883
13884 to:
13885
13886 Find First two Fields in a Buffer
13887
13888 The following example uses strtok() to find two character strings (a
13889 key and data associated with that key) separated by any combination of
13890 <space>, <tab>, or <newline> characters at the start of the array of
13891 characters pointed to by buffer.
13892
13893 #include <string.h>
13894 ...
13895 char *buffer;
13896 ...
13897 struct element {
13898     char *key;
13899     char *data;
13900 } e;
13901 ...
13902 // Load the buffer...
13903 ...
13904 // Get the key and its data...
13905 e.key = strtok(buffer, " \t\n");
13906 e.data = strtok(NULL, " \t\n");
13907 // Process the rest of the contents of the buffer...
13908 ...
13909
13910 Rationale: Austin Group Defect Report(s) applied: 177.
13911 See http://austingroupbugs.net/view.php?id=177
13912
13913
13914 Change Number: XSH/TC1/D5/0616 [453]
13915
13916
13917 On Page: 2044 Line: 64778 Section: strtol()
13918
13919 In the DESCRIPTION section, change from:
13920
13921 ... the value of str is stored ...
13922
13923 to:
13924
13925 ... the value of str shall be stored ...
13926
13927 Rationale: Austin Group Defect Report(s) applied: 453.
13928 See http://austingroupbugs.net/view.php?id=453
13929
13930
13931 Change Number: XSH/TC1/D5/0617 [105]
13932
13933
13934 On Page: 2044 Line: 64780 Section: strtol()
13935
13936 In the DESCRIPTION section, remove the CX shading from the paragraphs:
13937
13938 The strtol() function shall not change the setting of errno if successful.
```

13939
 13940 Since 0, {LONG_MIN} or {LLONG_MIN}, and {LONG_MAX} or {LLONG_MAX} are
 13941 returned on error and are also valid returns on success, an application
 13942 wishing to check for error situations should set errno to 0, then call
 13943 strtol() or strtoll(), then check errno.
 13944
 13945 and then change the first paragraph from:
 13946
 13947 The strtol() function shall not change the setting of errno if successful.
 13948
 13949 to:
 13950
 13951 These functions shall not change the setting of errno if successful.
 13952
 13953 *Rationale:* Austin Group Defect Report(s) applied: 105.
 13954 See <http://austingroupbugs.net/view.php?id=105>
 13955
 13956 **Change Number: XSH/TC1/D5/0618 [453]**
 13957
 13958
 13959 On Page: 2044 Line: 64785 Section: strtol()
 13960
 13961 In the RETURN VALUE section, change from:
 13962
 13963 If no conversion could be performed, 0 shall be returned [CX]and
 13964 errno may be set to [EINVAL][CX].
 13965
 13966 to:
 13967
 13968 If no conversion could be performed, 0 shall be returned [CX]and
 13969 errno may be set to [EINVAL][CX].
 13970
 13971 [CX]If the value of base is not supported, 0 shall be returned and
 13972 errno shall be set to [EINVAL].[/CX]
 13973
 13974 *Rationale:* Austin Group Defect Report(s) applied: 453.
 13975 See <http://austingroupbugs.net/view.php?id=453>
 13976
 13977
 13978 **Change Number: XSH/TC1/D5/0619 [453]**
 13979
 13980
 13981 On Page: 2044 Line: 64791 Section: strtol()
 13982
 13983 In the ERRORS section, change from:
 13984
 13985 These functions shall fail if:
 13986
 13987 [ERANGE] The value to be returned is not representable.
 13988
 13989 These functions may fail if:
 13990
 13991 [EINVAL] [CX] The value of base is not supported. [/CX]
 13992
 13993 to:
 13994
 13995 These functions shall fail if:
 13996
 13997 [EINVAL] [CX] The value of base is not supported. [/CX]
 13998
 13999 [ERANGE] The value to be returned is not representable.
 14000
 14001

14002 These functions may fail if:
14003
14004 [EINVAL] No conversion could be performed.
14005
14006 *Rationale:* Austin Group Defect Report(s) applied: 453.
14007 See <http://austingroupbugs.net/view.php?id=453>
14008
14009
14010 **Change Number: XSH/TC1/D5/0620 [453]**
14011
14012
14013 On Page: 2044 Line: 64798 Section: strtol()
14014
14015 In the APPLICATION USAGE section, change from:
14016
14017 None.
14018
14019 to:
14020
14021 Since the value of *endptr is unspecified if the value of base is
14022 not supported, applications should either ensure that base has a
14023 supported value (0 or between 2 and 36) before the call, or check
14024 for an [EINVAL] error before examining *endptr.
14025
14026 *Rationale:* Austin Group Defect Report(s) applied: 453.
14027 See <http://austingroupbugs.net/view.php?id=453>
14028
14029
14030 **Change Number: XSH/TC1/D5/0621 [105]**
14031
14032
14033 On Page: 2049 Line: 64884 Section: strtoul()
14034
14035 In the DESCRIPTION section, remove the CX shading from the paragraphs:
14036
14037 The strtoul() function shall not change the setting of errno if successful.
14038
14039 Since 0, {ULONG_MAX}, and {ULLONG_MAX} are returned on error and are
14040 also valid returns on success, an application wishing to check for error
14041 situations should set errno to 0, then call strtoul() or strtoull(),
14042 then check errno.
14043
14044 and then change the first paragraph from:
14045
14046 The strtoul() function shall not change the setting of errno if successful.
14047
14048 to:
14049
14050 These functions shall not change the setting of errno if successful.
14051
14052 *Rationale:* Austin Group Defect Report(s) applied: 105.
14053 See <http://austingroupbugs.net/view.php?id=105>
14054
14055
14056 **Change Number: XSH/TC1/D5/0622 [453]**
14057
14058
14059 On Page: 2049 Line: 64889 Section: strtoul()
14060
14061 In the RETURN VALUE section, change from:
14062
14063 If no conversion could be performed, 0 shall be returned [CX]and
14064 errno may be set to [EINVAL][CX].

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

14065

14066 to:

14067

14068 If no conversion could be performed, 0 shall be returned [CX]and
 14069 errno may be set to [EINVAL][CX].

14070

14071 [CX]If the value of base is not supported, 0 shall be returned and
 14072 errno shall be set to [EINVAL].[/CX]

14073

14074 *Rationale:* Austin Group Defect Report(s) applied: 453.
 14075 See <http://austingroupbugs.net/view.php?id=453>

14076

14077

14078 **Change Number: XSH/TC1/D5/0623 [453]**

14079

14080

14081 On Page: 2049 Line: 64902 Section: strtoul()

14082

14083 In the APPLICATION USAGE section, change from:

14084

14085 None.

14086

14087 to:

14088

14089 Since the value of *endptr is unspecified if the value of base is
 14090 not supported, applications should either ensure that base has a
 14091 supported value (0 or between 2 and 36) before the call, or check
 14092 for an [EINVAL] error before examining *endptr.

14093

14094 *Rationale:* Austin Group Defect Report(s) applied: 453.
 14095 See <http://austingroupbugs.net/view.php?id=453>

14096

14097

14098 **Change Number: XSH/TC1/D5/0624 [283]**

14099

14100

14101 On Page: 2052 Line: 64952 Section: strxfrm()

14102

14103 In the DESCRIPTION section, add a new paragraph to the end of the section:

14104

14105 [CX]The behavior is undefined if the locale argument to strxfrm_l()
 14106 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 14107 object handle.[/CX]

14108

14109 *Rationale:* Austin Group Defect Report(s) applied: 283.
 14110 See <http://austingroupbugs.net/view.php?id=283>

14111

14112

14113 **Change Number: XSH/TC1/D5/0625 [283]**

14114

14115

14116 On Page: 2052 Line: 64963 Section: strxfrm()

14117

14118 In the ERRORS section, delete:

14119

14120 The strxfrm_l() function may fail if:

14121

14122 [EINVAL] locale is not a valid locale object handle.

14123

14124 *Rationale:* Austin Group Defect Report(s) applied: 283.
 14125 See <http://austingroupbugs.net/view.php?id=283>

14126

14127

14128 **Change Number: XSH/TC1/D5/0626** [302]
14129
14130
14131 On Page: 2052 Line: 64969 Section: `strxfrm()`
14132
14133 change from:
14134
14135 ... the locale of the process ...
14136
14137 to:
14138
14139 ... the current locale ...
14140
14141 *Rationale:* Austin Group Defect Report(s) applied: 302.
14142 See <http://austingroupbugs.net/view.php?id=302>
14143 These changes were overlooked during the revision when
14144 per-thread locales were added.
14145
14146
14147 **Change Number: XSH/TC1/D5/0627** [146,428]
14148
14149
14150 On Page: 2057 Line: 65056 Section: `symlink()`
14151
14152 In the DESCRIPTION section, add a new paragraph after the existing
14153 third paragraph:
14154
14155 If path2 names a symbolic link, `symlink()` shall fail and set `errno` to
14156 `[EEXIST]`.
14157
14158 *Rationale:* Austin Group Defect Report(s) applied: 146,428.
14159 See <http://austingroupbugs.net/view.php?id=146>
14160 See <http://austingroupbugs.net/view.php?id=428>
14161
14162
14163 **Change Number: XSH/TC1/D5/0628** [461]
14164
14165
14166 On Page: 2057 Line: 65077 Section: `symlink()`
14167
14168 In the DESCRIPTION section, change from:
14169
14170 ... the current working directory is used ...
14171
14172 to:
14173
14174 ... the current working directory shall be used ...
14175
14176 *Rationale:* Austin Group Defect Report(s) applied: 461.
14177 See <http://austingroupbugs.net/view.php?id=461>
14178
14179
14180 **Change Number: XSH/TC1/D5/0629** [146,428]
14181
14182
14183 On Page: 2058 Line: 65087 Section: `symlink()`
14184
14185 In the ERRORS section, change the `[EEXIST]` error from:
14186
14187 `[EEXIST]` The path2 argument names an existing file or symbolic link.
14188
14189 to:
14190

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

14191 [EEXIST] The path2 argument names an existing file.
 14192
 14193 *Rationale:* Austin Group Defect Report(s) applied: 146,428.
 14194 See <http://austingroupbugs.net/view.php?id=146>
 14195 See <http://austingroupbugs.net/view.php?id=428>
 14196
 14197
 14198 **Change Number: XSH/TC1/D5/0630** [146,428,436]
 14199
 14200
 14201 On Page: 2058 Line: 65095 Section: symlink()
 14202
 14203 In the ERRORS section, change the [ENOENT] error from:
 14204
 14205 [ENOENT] A component of path2 does not name an existing file or
 14206 path2 is an empty string.
 14207
 14208 to:
 14209
 14210 [ENOENT] A component of the path prefix of path2 does not name an
 14211 existing file or path2 is an empty string.
 14212
 14213 [ENOENT] or [ENOTDIR]
 14214 The path2 argument contains at least one non-`<slash>` character and ends
 14215 with one or more trailing `<slash>` characters. If path2 names an existing
 14216 file, an [ENOENT] error shall not occur.
 14217
 14218 *Rationale:* Austin Group Defect Report(s) applied: 146,428,436.
 14219 See <http://austingroupbugs.net/view.php?id=146>
 14220 See <http://austingroupbugs.net/view.php?id=428>
 14221 See <http://austingroupbugs.net/view.php?id=436>
 14222
 14223
 14224 **Change Number: XSH/TC1/D5/0631** [324]
 14225
 14226
 14227 On Page: 2058 Line: 65102 Section: symlink()
 14228
 14229 In the ERRORS section, for the [ENOTDIR] error, change from:
 14230
 14231 A component of the path prefix of path2 is not a directory.
 14232
 14233 to:
 14234
 14235 A component of the path prefix of path2 names an existing file that is
 14236 neither a directory nor a symbolic link to a directory.
 14237
 14238 *Rationale:* Austin Group Defect Report(s) applied: 324.
 14239 See <http://austingroupbugs.net/view.php?id=324>
 14240 This is an editorial issue clarifying the intent of the
 14241 standard.
 14242
 14243
 14244 **Change Number: XSH/TC1/D5/0632** [278]
 14245
 14246
 14247 On Page: 2058 Line: 65108 Section: symlink()
 14248
 14249 In the ERRORS section, add (after the [EBADF] error):
 14250
 14251 [ENOTDIR] The path2 argument is not an absolute path and fd is a file
 14252 descriptor associated with a non-directory file.
 14253

14254 *Rationale:* Austin Group Defect Report(s) applied: 278.
14255 See <http://austingroupbugs.net/view.php?id=278>
14256
14257
14258 **Change Number: XSH/TC1/D5/0633 [278]**
14259
14260
14261 On Page: 2058 Line: 65116 Section: symlink()
14262
14263 In the ERRORS section, delete:
14264
14265 The symlinkat() function may fail if:
14266
14267 [ENOTDIR] The path2 argument is not an absolute path and fd is neither
14268 AT_FDCWD nor a file descriptor associated with a directory.
14269
14270 *Rationale:* Austin Group Defect Report(s) applied: 278.
14271 See <http://austingroupbugs.net/view.php?id=278>
14272
14273
14274 **Change Number: XSH/TC1/D5/0634 [151]**
14275
14276
14277 On Page: 2059 Line: 65129 Section: symlink()
14278
14279 In the RATIONALE section, delete the first sentence:
14280
14281 Since POSIX.1-2008 does not require any association of file times with
14282 symbolic links, there is no requirement that file times be updated
14283 by symlink().
14284
14285 *Rationale:* Austin Group Defect Report(s) applied: 151.
14286 See <http://austingroupbugs.net/view.php?id=151>
14287
14288
14289 **Change Number: XSH/TC1/D5/0635 [68]**
14290
14291
14292 On Page: 2074 Line: 65706 Section: tan()
14293
14294 In the RETURN VALUE section, change from:
14295
14296 If the correct value would cause underflow, and is not representable,
14297 a range error may occur, and [MX]either 0.0 (if supported), or[/MX]
14298 an implementation-defined value shall be returned.
14299
14300 to:
14301
14302 If the correct value would cause underflow, [MXX]and is not
14303 representable[/MXX], a range error may occur, and tan(), tanf(), and
14304 tanl() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point is
14305 not supported) an implementation-defined value no greater in magnitude
14306 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.
14307
14308 *Rationale:* Austin Group Defect Report(s) applied: 68.
14309 See <http://austingroupbugs.net/view.php?id=68>
14310
14311
14312 **Change Number: XSH/TC1/D5/0636 [68]**
14313
14314
14315 On Page: 2074 Line: 65710 Section: tan()
14316

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

14317 In the RETURN VALUE section, change from:
 14318
 14319 [MX]If x is subnormal, a range error may occur and x should be
 14320 returned.[/MX]
 14321
 14322 to:
 14323
 14324 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
 14325 be returned.[/MXX]
 14326
 14327 [MX]If x is not returned, tan(), tanf(), and tanl() shall return
 14328 an implementation-defined value no greater in magnitude than DBL_MIN,
 14329 FLT_MIN, and LDBL_MIN, respectively.[/MX]
 14330
 14331 *Rationale:* Austin Group Defect Report(s) applied: 68.
 14332 See <http://austingroupbugs.net/view.php?id=68>
 14333
 14334
 14335 **Change Number: XSH/TC1/D5/0637 [68]**
 14336
 14337
 14338 On Page: 2074 Line: 65713 Section: tan()
 14339
 14340 In the RETURN VALUE section, change the MX shading to MXX for:
 14341
 14342 If the correct value would cause underflow, and is representable,
 14343 a range error may occur and the correct value shall be returned.
 14344
 14345 *Rationale:* Austin Group Defect Report(s) applied: 68.
 14346 See <http://austingroupbugs.net/view.php?id=68>
 14347
 14348
 14349 **Change Number: XSH/TC1/D5/0638 [68]**
 14350
 14351
 14352 On Page: 2076 Line: 65792 Section: tanh()
 14353
 14354 In the RETURN VALUE section, change from:
 14355
 14356 [MX]If x is subnormal, a range error may occur and x should be
 14357 returned.[/MX]
 14358
 14359 to:
 14360
 14361 [MX]If x is subnormal, a range error may occur[/MX] [MXX]and x should
 14362 be returned.[/MXX]
 14363
 14364 [MX]If x is not returned, tanh(), tanhf(), and tanhl() shall return
 14365 an implementation-defined value no greater in magnitude than DBL_MIN,
 14366 FLT_MIN, and LDBL_MIN, respectively.[/MX]
 14367
 14368 *Rationale:* Austin Group Defect Report(s) applied: 68.
 14369 See <http://austingroupbugs.net/view.php?id=68>
 14370
 14371
 14372 **Change Number: XSH/TC1/D5/0639 [79]**
 14373
 14374
 14375 On Page: 2079 Line: 65840 Section: tcdrain()
 14376
 14377 In the DESCRIPTION section change from:
 14378
 14379 If the calling process is blocking or ignoring SIGTTOU signals ...

14380
14381 to:
14382
14383 If the calling thread is blocking SIGTTOU signals or the process is
14384 ignoring SIGTTOU signals ...
14385
14386 *Rationale:* Austin Group Defect Report(s) applied: 79.
14387 See <http://austingroupbugs.net/view.php?id=79>
14388
14389
14390 **Change Number: XSH/TC1/D5/0640 [79]**
14391
14392
14393 On Page: 2079 Line: 65848 Section: tcdrain()
14394
14395 In the ERRORS section, add:
14396
14397 [EIO] The process group of the writing process is orphaned, the calling
14398 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14399
14400 *Rationale:* Austin Group Defect Report(s) applied: 79.
14401 See <http://austingroupbugs.net/view.php?id=79>
14402
14403
14404 **Change Number: XSH/TC1/D5/0641 [79]**
14405
14406
14407 On Page: 2079 Line: 65850 Section: tcdrain()
14408
14409 In the ERRORS section, delete:
14410
14411 The [...] function may fail if:
14412
14413 [EIO] The process group of the writing process is orphaned, and the
14414 writing process is not ignoring or blocking SIGTTOU.
14415
14416 *Rationale:* Austin Group Defect Report(s) applied: 79.
14417 See <http://austingroupbugs.net/view.php?id=79>
14418
14419
14420 **Change Number: XSH/TC1/D5/0642 [79]**
14421
14422
14423 On Page: 2081 Line: 65895 Section: tcflow()
14424
14425 In the DESCRIPTION section change from:
14426
14427 If the calling process is blocking or ignoring SIGTTOU signals ...
14428
14429 to:
14430
14431 If the calling thread is blocking SIGTTOU signals or the process is
14432 ignoring SIGTTOU signals ...
14433
14434 *Rationale:* Austin Group Defect Report(s) applied: 79.
14435 See <http://austingroupbugs.net/view.php?id=79>
14436
14437
14438 **Change Number: XSH/TC1/D5/0643 [79]**
14439
14440
14441 On Page: 2081 Line: 65903 Section: tcflow()
14442

IEEE Std 1003.1™-2008/Cor 1-2013
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 1

14443 In the ERRORS section, add:
14444
14445 [EIO] The process group of the writing process is orphaned, the calling
14446 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14447
14448 *Rationale:* Austin Group Defect Report(s) applied: 79.
14449 See <http://austingroupbugs.net/view.php?id=79>
14450
14451
14452 **Change Number: XSH/TC1/D5/0644 [79]**
14453
14454
14455 On Page: 2081 Line: 65905 Section: tcflow()
14456
14457 In the ERRORS section, delete:
14458
14459 The [...] function may fail if:
14460
14461 [EIO] The process group of the writing process is orphaned, and the
14462 writing process is not ignoring or blocking SIGTTOU.
14463
14464 *Rationale:* Austin Group Defect Report(s) applied: 79.
14465 See <http://austingroupbugs.net/view.php?id=79>
14466
14467
14468 **Change Number: XSH/TC1/D5/0645 [79]**
14469
14470
14471 On Page: 2083 Line: 65943 Section: tcflush()
14472
14473 In the DESCRIPTION section change from:
14474
14475 If the calling process is blocking or ignoring SIGTTOU signals ...
14476
14477 to:
14478
14479 If the calling thread is blocking SIGTTOU signals or the process is
14480 ignoring SIGTTOU signals ...
14481
14482 *Rationale:* Austin Group Defect Report(s) applied: 79.
14483 See <http://austingroupbugs.net/view.php?id=79>
14484
14485
14486 **Change Number: XSH/TC1/D5/0646 [79]**
14487
14488
14489 On Page: 2083 Line: 65951 Section: tcflush()
14490
14491 In the ERRORS section, add:
14492
14493 [EIO] The process group of the writing process is orphaned, the calling
14494 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14495
14496 *Rationale:* Austin Group Defect Report(s) applied: 79.
14497 See <http://austingroupbugs.net/view.php?id=79>
14498
14499
14500 **Change Number: XSH/TC1/D5/0647 [79]**
14501
14502
14503 On Page: 2083 Line: 65953 Section: tcflush()
14504
14505 In the ERRORS section, delete:

14506
14507 The [...] function may fail if:
14508
14509 [EIO] The process group of the writing process is orphaned, and the
14510 writing process is not ignoring or blocking SIGTTOU.
14511
14512 *Rationale:* Austin Group Defect Report(s) applied: 79.
14513 See <http://austingroupbugs.net/view.php?id=79>
14514
14515
14516 **Change Number: XSH/TC1/D5/0648** [421]
14517
14518
14519 On Page: 2089 Line: 66089 Section: tcgetsid()
14520
14521 In the RETURN VALUE section, change from:
14522
14523 Otherwise, a value of (pid_t)-1 shall be returned, ...
14524
14525 to:
14526
14527 Otherwise, a value of -1 shall be returned, ...
14528
14529 *Rationale:* Austin Group Defect Report(s) applied: 421.
14530 See <http://austingroupbugs.net/view.php?id=421>
14531
14532
14533 **Change Number: XSH/TC1/D5/0649** [79]
14534
14535
14536 On Page: 2090 Line: 66132 Section: tcsendbreak()
14537
14538 In the DESCRIPTION section change from:
14539
14540 If the calling process is blocking or ignoring SIGTTOU signals ...
14541
14542 to:
14543
14544 If the calling thread is blocking SIGTTOU signals or the process is
14545 ignoring SIGTTOU signals ...
14546
14547 *Rationale:* Austin Group Defect Report(s) applied: 79.
14548 See <http://austingroupbugs.net/view.php?id=79>
14549
14550
14551 **Change Number: XSH/TC1/D5/0650** [79]
14552
14553
14554 On Page: 2090 Line: 66139 Section: tcsendbreak()
14555
14556 In the ERRORS section, add:
14557
14558 [EIO] The process group of the writing process is orphaned, the calling
14559 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14560
14561 *Rationale:* Austin Group Defect Report(s) applied: 79.
14562 See <http://austingroupbugs.net/view.php?id=79>
14563
14564
14565 **Change Number: XSH/TC1/D5/0651** [79]
14566
14567
14568 On Page: 2090 Line: 66141 Section: tcsendbreak()

14569
14570 In the ERRORS section, delete:
14571
14572 The [...] function may fail if:
14573
14574 [EIO] The process group of the writing process is orphaned, and the
14575 writing process is not ignoring or blocking SIGTTOU.
14576
14577 *Rationale:* Austin Group Defect Report(s) applied: 79.
14578 See <http://austingroupbugs.net/view.php?id=79>
14579
14580
14581 **Change Number: XSH/TC1/D5/0652 [79]**
14582
14583
14584 On Page: 2092 Line: 66203 Section: tcsetattr()
14585
14586 In the DESCRIPTION section change from:
14587
14588 If the calling process is blocking or ignoring SIGTTOU signals ...
14589
14590 to:
14591
14592 If the calling thread is blocking SIGTTOU signals or the process is
14593 ignoring SIGTTOU signals ...
14594
14595 *Rationale:* Austin Group Defect Report(s) applied: 79.
14596 See <http://austingroupbugs.net/view.php?id=79>
14597
14598
14599 **Change Number: XSH/TC1/D5/0653 [79]**
14600
14601
14602 On Page: 2093 Line: 66215 Section: tcsetattr()
14603
14604 In the ERRORS section, add:
14605
14606 [EIO] The process group of the writing process is orphaned, the calling
14607 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14608
14609 *Rationale:* Austin Group Defect Report(s) applied: 79.
14610 See <http://austingroupbugs.net/view.php?id=79>
14611
14612
14613 **Change Number: XSH/TC1/D5/0654 [79]**
14614
14615
14616 On Page: 2093 Line: 66217 Section: tcsetattr()
14617
14618 In the ERRORS section, delete:
14619
14620 The [...] function may fail if:
14621
14622 [EIO] The process group of the writing process is orphaned, and the
14623 writing process is not ignoring or blocking SIGTTOU.
14624
14625 *Rationale:* Austin Group Defect Report(s) applied: 79.
14626 See <http://austingroupbugs.net/view.php?id=79>
14627
14628
14629 **Change Number: XSH/TC1/D5/0655 [79]**
14630
14631

14632 On Page: 2095 Line: 66280 Section: tcsetpgrp()
14633
14634 In the DESCRIPTION section change from:
14635
14636 If the calling process is blocking or ignoring SIGTTOU signals ...
14637
14638 to:
14639
14640 If the calling thread is blocking SIGTTOU signals or the process is
14641 ignoring SIGTTOU signals ...
14642
14643 *Rationale:* Austin Group Defect Report(s) applied: 79.
14644 See <http://austingroupbugs.net/view.php?id=79>
14645
14646
14647 **Change Number: XSH/TC1/D5/0656 [79]**
14648
14649
14650 On Page: 2095 Line: 66288 Section: tcsetpgrp()
14651
14652 In the ERRORS section, add:
14653
14654 [EIO] The process group of the writing process is orphaned, the calling
14655 thread is not blocking SIGTTOU, and the process is not ignoring SIGTTOU.
14656
14657 *Rationale:* Austin Group Defect Report(s) applied: 79.
14658 See <http://austingroupbugs.net/view.php?id=79>
14659
14660
14661 **Change Number: XSH/TC1/D5/0657 [291]**
14662
14663
14664 On Page: 2102 Line: 66538-66539 Section: tempnam()
14665
14666 In the EXAMPLES section, change both instances of "filename" to
14667 "pathname".
14668
14669 *Rationale:* Austin Group Defect Report(s) applied: 291.
14670 See <http://austingroupbugs.net/view.php?id=291>
14671
14672
14673 **Change Number: XSH/TC1/D5/0658 [137]**
14674
14675
14676 On Page: 2102 Line: 66543 Section: tempnam()
14677
14678 In the EXAMPLES section, change from:
14679
14680 char *directory = "/tmp";
14681 char *fileprefix = "file";
14682
14683 to:
14684
14685 const char *directory = "/tmp";
14686 const char *fileprefix = "file";
14687
14688 *Rationale:* Austin Group Defect Report(s) applied: 137.
14689 See <http://austingroupbugs.net/view.php?id=137>
14690
14691
14692 **Change Number: XSH/TC1/D5/0659 [137]**
14693
14694

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

14695 On Page: 2103 Line: 66559 Section: tempnam()
 14696
 14697 In the SEE ALSO section, add mkdtemp() and mkstemp().
 14698
 14699 *Rationale:* Austin Group Defect Report(s) applied: 137.
 14700 See <http://austingroupbugs.net/view.php?id=137>
 14701
 14702
 14703 **Change Number: XSH/TC1/D5/0660 [68]**
 14704
 14705
 14706 On Page: 2105 Line: 66603 Section: tgamma()
 14707
 14708 In the RETURN VALUE section, add a new section after L66603:
 14709
 14710 If the correct value would cause underflow, [MXX]and is not
 14711 representable[/MXX], a range error may occur, and tgamma(), tgammaf(),
 14712 and tgammal() shall return [MXX]0.0, or[/MXX] (if IEC 60559 Floating-Point
 14713 is not supported) an implementation-defined value no greater in magnitude
 14714 than DBL_MIN, FLT_MIN, and LDBL_MIN, respectively.
 14715
 14716 [MXX]If the correct value would cause underflow, and is representable,
 14717 a range error may occur and the correct value shall be returned.[/MXX]
 14718
 14719 *Rationale:* Austin Group Defect Report(s) applied: 68
 14720 See <http://austingroupbugs.net/view.php?id=68>
 14721
 14722
 14723 **Change Number: XSH/TC1/D5/0661 [320]**
 14724
 14725
 14726 On Page: 2105 Line: 66606 Section: tgamma()
 14727
 14728 In the RETURN VALUE section, change from:
 14729
 14730 ... and either a NaN (if supported), or an implementation-defined
 14731 value shall be returned.
 14732
 14733 to:
 14734
 14735 ... and a NaN shall be returned.
 14736
 14737 *Rationale:* Austin Group Defect Report(s) applied: 320.
 14738 See <http://austingroupbugs.net/view.php?id=320>
 14739
 14740
 14741 **Change Number: XSH/TC1/D5/0662 [68]**
 14742
 14743
 14744 On Page: 2106 Line: 66635 Section: tgamma()
 14745
 14746 In the ERRORS section, add to the end of the "may fails" errors:
 14747
 14748 Range Error The result underflows.
 14749
 14750 If the integer expression (math_errhandling & MATH_ERRNO) is non-zero,
 14751 then errno shall be set to [ERANGE]. If the integer expression
 14752 (math_errhandling & MATH_ERREXCEPT) is non-zero, then the underflow
 14753 floating-point exception shall be raised.
 14754
 14755 *Rationale:* Austin Group Defect Report(s) applied: 68.
 14756 See <http://austingroupbugs.net/view.php?id=68>
 14757

14758
14759 **Change Number: XSH/TC1/D5/0663** [106]
14760
14761
14762 On Page: 2107 Line: 66673 Section: time()
14763
14764 In the ERRORS section, change from:
14765
14766 No errors are defined.
14767
14768 to:
14769
14770 The time() function may fail if:
14771
14772 [Eoverflow] The number of seconds since the Epoch will not fit in an
14773 object of type time_t.
14774
14775 with CX shading.
14776
14777 *Rationale:* Austin Group Defect Report(s) applied: 106.
14778 See <http://austingroupbugs.net/view.php?id=106>
14779
14780
14781 **Change Number: XSH/TC1/D5/0664** [350]
14782
14783
14784 On Page: 2108 Line: 66712 Section: time()
14785
14786 In the RATIONALE section, change from:
14787
14788 The time() function returns a value in seconds (type time_t) while
14789 times() returns a set of values in clock ticks (type clock_t).
14790 Some historical implementations, such as 4.3 BSD, have mechanisms
14791 capable of returning more precise times (see below). A generalized
14792 timing scheme to unify these various timing mechanisms has been
14793 proposed but not adopted.
14794
14795 to:
14796
14797 The time() function returns a value in seconds while clock_gettime()
14798 and gettimeofday() return a struct timespec (seconds and nanoseconds)
14799 and struct timeval (seconds and microseconds), respectively, and are
14800 therefore capable of returning more precise times. The times() function
14801 is also capable of more precision than time() as it returns a value in
14802 clock ticks, although it returns the elapsed time since an arbitrary
14803 point such as system boot time, not since the epoch.
14804
14805 *Rationale:* Austin Group Defect Report(s) applied: 350.
14806 See <http://austingroupbugs.net/view.php?id=350>
14807
14808
14809 **Change Number: XSH/TC1/D5/0665** [106]
14810
14811
14812 On Page: 2108 Line: 66718 Section: time()
14813
14814 In the RATIONALE section, add after the 2nd paragraph a new paragraph:
14815
14816 On some systems the time() function is implemented using a system call
14817 that does not return an error condition in addition to the return
14818 value. On these systems it is impossible to differentiate between
14819 valid and invalid return values and hence overflow conditions cannot be
14820 reliably detected.

14821

14822 *Rationale:* Austin Group Defect Report(s) applied: 106.

14823 See <http://austingroupbugs.net/view.php?id=106>

14824

14825

14826 **Change Number: XSH/TC1/D5/0666 [350]**

14827

14828

14829 On Page: 2108 Line: 66724 Section: time()

14830

14831 In the RATIONALE section, delete the sentence:

14832

14833 4.3 BSD includes time() only as an alternate function to the more

14834 flexible gettimeofday() function.

14835

14836 *Rationale:* Austin Group Defect Report(s) applied: 350.

14837 See <http://austingroupbugs.net/view.php?id=350>

14838

14839

14840 **Change Number: XSH/TC1/D5/0667 [350]**

14841

14842

14843 On Page: 2108 Line: 66732 Section: time()

14844

14845 In the SEE ALSO section, add clock_gettime(), times(), utimes(), and

14846 utimensat() to the SEE ALSO list.

14847

14848 *Rationale:* Austin Group Defect Report(s) applied: 350.

14849 See <http://austingroupbugs.net/view.php?id=350>

14850

14851

14852 **Change Number: XSH/TC1/D5/0668 [14]**

14853

14854

14855 On Page: 2122 Line: 67175 Section: tmpfile()

14856

14857 In the SEE ALSO section, add a reference to XSH Section 2.5.

14858

14859 *Rationale:* Austin Group Defect Report(s) applied: 14.

14860 See <http://austingroupbugs.net/view.php?id=14>

14861 This is an editorial improvement

14862

14863

14864 **Change Number: XSH/TC1/D5/0669 [291]**

14865

14866

14867 On Page: 2123 Line: 67207 Section: tmpnam()

14868

14869 In the DESCRIPTION section, change from:

14870

14871 ... valid filename and that is not the same as the name of an existing

14872 file.

14873

14874 to:

14875

14876 ... valid pathname that does not name an existing file.

14877

14878 *Rationale:* Austin Group Defect Report(s) applied: 291.

14879 See <http://austingroupbugs.net/view.php?id=291>

14880

14881

14882 **Change Number: XSH/TC1/D5/0670 [291,429]**

14883

14884
14885 On Page: 2123 Line: 67227-67235 Section: tmpnam()
14886
14887 In the EXAMPLES and APPLICATION USAGE sections, change all instances of
14888 "filename" to "pathname".
14889
14890 *Rationale:* Austin Group Defect Report(s) applied: 291,429.
14891 See <http://austingroupbugs.net/view.php?id=291>
14892 See <http://austingroupbugs.net/view.php?id=429>
14893
14894
14895 **Change Number: XSH/TC1/D5/0671 [283]**
14896
14897
14898 On Page: 2126 Line: 67304 Section: tolower()
14899
14900 In the DESCRIPTION section, add a new paragraph to the end of the section:
14901
14902 [CX]The behavior is undefined if the locale argument to tolower_l()
14903 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
14904 object handle. [/CX]
14905
14906 *Rationale:* Austin Group Defect Report(s) applied: 283.
14907 See <http://austingroupbugs.net/view.php?id=283>
14908
14909
14910 **Change Number: XSH/TC1/D5/0672 [283]**
14911
14912
14913 On Page: 2126 Line: 67310 Section: tolower()
14914
14915 In the ERRORS section, change from:
14916
14917 The tolower_l() function may fail if:
14918
14919 [EINVAL] locale is not a valid locale object handle.
14920
14921 to:
14922
14923 No errors are defined.
14924
14925 *Rationale:* Austin Group Defect Report(s) applied: 283.
14926 See <http://austingroupbugs.net/view.php?id=283>
14927
14928
14929 **Change Number: XSH/TC1/D5/0673 [283]**
14930
14931
14932 On Page: 2127 Line: 67347 Section: toupper()
14933
14934 In the DESCRIPTION section, add a new paragraph to the end of the section:
14935
14936 [CX]The behavior is undefined if the locale argument to toupper_l()
14937 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
14938 object handle. [/CX]
14939
14940 *Rationale:* Austin Group Defect Report(s) applied: 283.
14941 See <http://austingroupbugs.net/view.php?id=283>
14942
14943
14944 **Change Number: XSH/TC1/D5/0674 [283]**
14945
14946

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

14947 On Page: 2127 Line: 67352 Section: toupper()
 14948
 14949 In the ERRORS section, change from:
 14950
 14951 The toupper_l() function may fail if:
 14952
 14953 [EINVAL] locale is not a valid locale object handle.
 14954
 14955 to:
 14956
 14957 No errors are defined.
 14958
 14959 *Rationale:* Austin Group Defect Report(s) applied: 283.
 14960 See <http://austingroupbugs.net/view.php?id=283>
 14961
 14962
 14963 **Change Number: XSH/TC1/D5/0675 [302]**
 14964
 14965
 14966 On Page: 2129 Line: 67386 Section: towctrans()
 14967
 14968 In the DESCRIPTION section, change from:
 14969
 14970 ... current locale of the process ...
 14971
 14972 to:
 14973
 14974 ... current locale ...
 14975
 14976 *Rationale:* Austin Group Defect Report(s) applied: 302.
 14977 See <http://austingroupbugs.net/view.php?id=302>
 14978 These changes were overlooked during the revision when
 14979 per-thread locales were added.
 14980
 14981
 14982 **Change Number: XSH/TC1/D5/0676 [283]**
 14983
 14984
 14985 On Page: 2129 Line: 67395 Section: towctrans()
 14986
 14987 In the DESCRIPTION section, add a new paragraph to the end of the section:
 14988
 14989 [CX]The behavior is undefined if the locale argument to towctrans_l()
 14990 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
 14991 object handle.[CX]
 14992
 14993 *Rationale:* Austin Group Defect Report(s) applied: 283.
 14994 See <http://austingroupbugs.net/view.php?id=283>
 14995
 14996
 14997 **Change Number: XSH/TC1/D5/0677 [283]**
 14998
 14999
 15000 On Page: 2129 Line: 67402 Section: towctrans()
 15001
 15002 In the ERRORS section, delete:
 15003
 15004 The towctrans_l() function may fail if:
 15005
 15006 [EINVAL] locale is not a valid locale object handle.
 15007
 15008 *Rationale:* Austin Group Defect Report(s) applied: 283.
 15009 See <http://austingroupbugs.net/view.php?id=283>

15010
15011
15012
15013
15014
15015
15016
15017
15018
15019
15020
15021
15022
15023
15024
15025
15026
15027
15028
15029
15030
15031
15032
15033
15034
15035
15036
15037
15038
15039
15040
15041
15042
15043
15044
15045
15046
15047
15048
15049
15050
15051
15052
15053
15054
15055
15056
15057
15058
15059
15060
15061
15062
15063
15064
15065
15066
15067
15068
15069
15070
15071
15072

Change Number: XSH/TC1/D5/0678 [302]

On Page: 2131 Line: 67442 Section: towlower()

In the DESCRIPTION section, change from:

... the locale of the process ...

to:

... the current locale ...

Rationale: Austin Group Defect Report(s) applied: 302.

See <http://austingroupbugs.net/view.php?id=302>

These changes were overlooked during the revision when per-thread locales were added.

Change Number: XSH/TC1/D5/0679 [283]

On Page: 2131 Line: 67445 Section: towlower()

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to towlower_l()
is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
object handle.[/CX]

Rationale: Austin Group Defect Report(s) applied: 283.

See <http://austingroupbugs.net/view.php?id=283>

Change Number: XSH/TC1/D5/0680 [283]

On Page: 2131 Line: 67451 Section: towlower()

In the ERRORS section, change from:

The towlower_l() function may fail if:

[EINVAL] locale is not a valid locale object handle.

to:

No errors are defined.

Rationale: Austin Group Defect Report(s) applied: 283.

See <http://austingroupbugs.net/view.php?id=283>

Change Number: XSH/TC1/D5/0681 [302]

On Page: 2133 Line: 67491 Section: towupper()

change from:

... the locale of the process ...

15073

15074 to:

15075

15076 ... the current locale ...

15077

15078 *Rationale:* Austin Group Defect Report(s) applied: 302.15079 See <http://austingroupbugs.net/view.php?id=302>

15080 These changes were overlooked during the revision when

15081 per-thread locales were added.

15082

15083

15084 **Change Number: XSH/TC1/D5/0682** [283]

15085

15086

15087 On Page: 2133 Line: 67494 Section: towupper()

15088

15089 In the DESCRIPTION section, add a new paragraph to the end of the section:

15090

15091 [CX]The behavior is undefined if the locale argument to towupper_l()

15092 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale

15093 object handle.[/CX]

15094

15095 *Rationale:* Austin Group Defect Report(s) applied: 283.15096 See <http://austingroupbugs.net/view.php?id=283>

15097

15098

15099 **Change Number: XSH/TC1/D5/0683** [283]

15100

15101

15102 On Page: 2133 Line: 67500 Section: towupper_l()

15103

15104 In the ERRORS section, change from:

15105

15106 The towupper_l() function may fail if:

15107

15108 [EINVAL] locale is not a valid locale object handle.

15109

15110 to:

15111

15112 No errors are defined.

15113

15114 *Rationale:* Austin Group Defect Report(s) applied: 283.15115 See <http://austingroupbugs.net/view.php?id=283>

15116

15117

15118 **Change Number: XSH/TC1/D5/0684** [346]

15119

15120

15121 On Page: 2135 Line: 67539 Section: trunc()

15122

15123 For trunc(), at P2135 L67539 (RETURN VALUE), add:

15124

15125 [MX]The result shall have the same sign as x.[/MX]

15126

15127 At L67547 (APPLICATION USAGE), replace "None." with:

15128

15129 The integral value returned by these functions need not be expressible

15130 as an intmax_t. The return value should be tested before assigning it

15131 to an integer type to avoid the undefined results of an integer overflow.

15132

15133 These functions may raise the inexact floating-point exception if the

15134 result differs in value from the argument.

15135

15136 *Rationale:* Austin Group Defect Report(s) applied: 346.
15137 See <http://austingroupbugs.net/view.php?id=346>
15138
15139
15140 **Change Number: XSH/TC1/D5/0685 [324]**
15141
15142
15143 On Page: 2136 Line: 67591 Section: truncate()
15144
15145 In the ERRORS section, for the [ENOTDIR] error, change from:
15146
15147 A component of the path prefix is not a directory, ...
15148
15149 to:
15150
15151 A component of the path prefix names an existing file that is neither
15152 a directory nor a symbolic link to a directory, ...
15153
15154 *Rationale:* Austin Group Defect Report(s) applied: 324.
15155 See <http://austingroupbugs.net/view.php?id=324>
15156 This is an editorial issue clarifying the intent of the
15157 standard.
15158
15159
15160 **Change Number: XSH/TC1/D5/0686 [75]**
15161
15162
15163 On Page: 2140 Line: 67654 Section: ttyname()
15164
15165 In the DESCRIPTION section, change from:
15166
15167 The return value may point to static data whose content is overwritten
15168 by each call.
15169
15170 to:
15171
15172 The application shall not modify the string returned. The returned
15173 pointer might be invalidated, or the string content might be overwritten
15174 by a subsequent call to ttyname().
15175
15176 *Rationale:* Austin Group Defect Report(s) applied: 75.
15177 See <http://austingroupbugs.net/view.php?id=75>
15178
15179
15180 **Change Number: XSH/TC1/D5/0687 [87,93]**
15181
15182
15183 On Page: 2151 Line: 67920 Section: ungetc()
15184
15185 In the DESCRIPTION section, change from:
15186
15187 A successful intervening call (with the stream pointed to by stream)
15188 to a file-positioning function (fseek(), fsetpos(), or rewind()) shall
15189 discard any pushed-back bytes for the stream.
15190
15191 to:
15192
15193 A successful intervening call (with the stream pointed to by stream)
15194 to a file-positioning function (fseek(), [CX]fseeko()[/CX] fsetpos(),
15195 or rewind()) [CX]or fflush()[/CX] shall discard any pushed-back bytes
15196 for the stream.
15197
15198 *Rationale:* Austin Group Defect Report(s) applied: 87,93.

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

15199 See <http://austingroupbugs.net/view.php?id=87>
 15200 See <http://austingroupbugs.net/view.php?id=93>
 15201 The `fseeko()` function is added to the list of file positioning functions.
 15202 The `fflush()` function is also added as a function that causes the discard
 15203 of any pushed-back bytes for the stream.
 15204
 15205
 15206 **Change Number: XSH/TC1/D5/0688 [87]**
 15207
 15208
 15209 On Page: 2151 Line: 67927 Section: `ungetc()`
 15210
 15211 In the DESCRIPTION section, change from:
 15212
 15213 The value of the file-position indicator for the stream after reading
 15214 or discarding all pushed-back bytes shall be the same as it was before
 15215 the bytes were pushed back.
 15216
 15217 to:
 15218
 15219 The value of the file-position indicator for the stream after all
 15220 pushed-back bytes have been read, or discarded by calling `fseek()`,
 15221 `[CX]fseeko()`, `[/CX] fsetpos()`, or `rewind()` `[CX]` (but not `fflush()`) `[/CX]`,
 15222 shall be the same as it was before the bytes were pushed back.
 15223
 15224 *Rationale:* Austin Group Defect Report(s) applied: 87.
 15225 See <http://austingroupbugs.net/view.php?id=87>
 15226
 15227
 15228 **Change Number: XSH/TC1/D5/0689 [14]**
 15229
 15230
 15231 On Page: 2151 Line: 67946 Section: `ungetc()`
 15232
 15233 In the SEE ALSO section, add a reference to XSH Section 2.5.
 15234
 15235 *Rationale:* Austin Group Defect Report(s) applied: 14.
 15236 See <http://austingroupbugs.net/view.php?id=14>
 15237 This is an editorial improvement
 15238
 15239
 15240 **Change Number: XSH/TC1/D5/0690 [87,93]**
 15241
 15242
 15243 On Page: 2152 Line: 67962 Section: `ungetwc()`
 15244
 15245 In the DESCRIPTION section, change from:
 15246
 15247 A successful intervening call (with the stream pointed to by `stream`)
 15248 to a file-positioning function (`fseek()`, `fsetpos()`, or `rewind()`) shall
 15249 discard any pushed-back characters for the stream.
 15250
 15251 to:
 15252
 15253 A successful intervening call (with the stream pointed to by `stream`)
 15254 to a file-positioning function (`fseek()`, `[CX]fseeko()` `[/CX]` `fsetpos()`,
 15255 or `rewind()`) `[CX]` or `fflush()` `[/CX]` shall discard any pushed-back characters
 15256 for the stream.
 15257
 15258 *Rationale:* Austin Group Defect Report(s) applied: 87,93.
 15259 See <http://austingroupbugs.net/view.php?id=87>
 15260 See <http://austingroupbugs.net/view.php?id=93>
 15261 The `fseeko()` function is added to the list of file positioning functions.

15262 The fflush() function is also added as a function that causes the discard of
15263 any pushed-back bytes for the stream.

15264

15265

15266 **Change Number: XSH/TC1/D5/0691 [87]**

15267

15268

15269 On Page: 2152 Line: 67971 Section: ungetwc()

15270

15271 In the DESCRIPTION section, change from:

15272

15273 The value of the file-position indicator for the stream after reading
15274 or discarding all pushed-back characters shall be the same as it was
15275 before the characters were pushed back.

15276

15277 to:

15278

15279 The value of the file-position indicator for the stream after all
15280 pushed-back characters have been read, or discarded by calling fseek(),
15281 [CX]fseeko(),[/CX] fsetpos(), or rewind() [CX](but not fflush())[/CX],
15282 shall be the same as it was before the characters were pushed back.

15283

15284 *Rationale:* Austin Group Defect Report(s) applied: 87.

15285 See <http://austingroupbugs.net/view.php?id=87>

15286

15287

15288 **Change Number: XSH/TC1/D5/0692 [14]**

15289

15290

15291 On Page: 2153 Line: 67992 Section: ungetwc()

15292

15293 In the SEE ALSO section, add a reference to XSH Section 2.5.

15294

15295 *Rationale:* Austin Group Defect Report(s) applied: 14.

15296 See <http://austingroupbugs.net/view.php?id=14>

15297 This is an editorial improvement

15298

15299

15300 **Change Number: XSH/TC1/D5/0693 [461]**

15301

15302

15303 On Page: 2154 Line: 68031 Section: unlink()

15304

15305 In the DESCRIPTION section, change from:

15306

15307 ... the current working directory is used ...

15308

15309 to:

15310

15311 ... the current working directory shall be used ...

15312

15313 *Rationale:* Austin Group Defect Report(s) applied: 461.

15314 See <http://austingroupbugs.net/view.php?id=461>

15315

15316

15317 **Change Number: XSH/TC1/D5/0694 [324]**

15318

15319

15320 On Page: 2155 Line: 68050 Section: unlink()

15321

15322 In the ERRORS section, for the [ENOTDIR] error, change from:

15323

15324 A component of the path prefix is not a directory, ...

15325
 15326 to:
 15327
 15328 A component of the path prefix names an existing file that is neither
 15329 a directory nor a symbolic link to a directory, ...
 15330
 15331 *Rationale:* Austin Group Defect Report(s) applied: 324.
 15332 See <http://austingroupbugs.net/view.php?id=324>
 15333 This is an editorial issue clarifying the intent of the
 15334 standard.
 15335
 15336
 15337 **Change Number: XSH/TC1/D5/0695 [278]**
 15338
 15339
 15340 On Page: 2155 Line: 68066 `unlink()`
 15341
 15342 In the ERRORS section, add (after the [EBADF] error):
 15343
 15344 [ENOTDIR] The path argument is not an absolute path and `fd` is a file
 15345 descriptor associated with a non-directory file.
 15346
 15347 *Rationale:* Austin Group Defect Report(s) applied: 278.
 15348 See <http://austingroupbugs.net/view.php?id=278>
 15349
 15350
 15351 **Change Number: XSH/TC1/D5/0696 [278]**
 15352
 15353
 15354 On Page: 2156 Line: 68085 Section: `unlink()`
 15355
 15356 In the ERRORS section, delete:
 15357
 15358 [ENOTDIR] The path argument is not an absolute path and `fd` is neither
 15359 `AT_FDCWD` nor a file descriptor associated with a directory.
 15360
 15361 *Rationale:* Austin Group Defect Report(s) applied: 278.
 15362 See <http://austingroupbugs.net/view.php?id=278>
 15363
 15364
 15365 **Change Number: XSH/TC1/D5/0697 [96]**
 15366
 15367
 15368 On Page: 2160 Line: 68233 Section: `unlockpt()`
 15369
 15370 In the RATIONALE section, change from:
 15371
 15372 None.
 15373
 15374 to:
 15375
 15376 See RATIONALE for `posix_openpt()`.
 15377
 15378 On L68237, in the SEE ALSO section, add `posix_openpt()`.
 15379
 15380 *Rationale:* Austin Group Defect Report(s) applied: 96.
 15381 See <http://austingroupbugs.net/view.php?id=96>
 15382 Add reference to `posix_openpt()` to the RATIONALE and SEE
 15383 ALSO sections.
 15384
 15385
 15386 **Change Number: XSH/TC1/D5/0698 [167]**
 15387

15388
15389 On Page: 2161 Line: 68256 Section: unsetenv()
15390
15391 In the DESCRIPTION section, remove the text:
15392
15393 If the application modifies environ or the pointers to which it points,
15394 the behavior of unsetenv() is undefined.
15395
15396 *Rationale:* Austin Group Defect Report(s) applied: 167.
15397 See <http://austingroupbugs.net/view.php?id=167>
15398 The text relating to undefined behavior on modification of
15399 environ is removed as this is now covered by general text added in XBD
15400 Section 8.1.
15401
15402
15403 **Change Number: XSH/TC1/D5/0699 [185]**
15404
15405
15406 On Page: 2161 Line: 68264 Section: unsetenv()
15407
15408 In the ERRORS section, change from:
15409
15410 [EINVAL] The name argument is a null pointer, points to an empty string,
15411 or points to a string containing an '=' character.
15412
15413 to:
15414
15415 [EINVAL] The name argument points to an empty string or points to a
15416 string containing an '=' character.
15417
15418 *Rationale:* Austin Group Defect Report(s) applied: 185.
15419 See <http://austingroupbugs.net/view.php?id=185>
15420
15421
15422 **Change Number: XSH/TC1/D5/0700 [290]**
15423
15424
15425 On Page: 2162 Line: 68296 Section: uselocale()
15426
15427 In the DESCRIPTION section Change "a null pointer" to "(locale_t)0".
15428
15429 *Rationale:* Austin Group Defect Report(s) applied: 290.
15430 See <http://austingroupbugs.net/view.php?id=290>
15431
15432
15433 **Change Number: XSH/TC1/D5/0701 [334]**
15434
15435
15436 On Page: 2162 Line: 68301 Section: uselocale()
15437
15438 In the RETURN VALUE section, change from:
15439
15440 The uselocale() function returns the locale handle from the previous
15441 call for the current thread. If there was no such previous call, the
15442 function shall return the value LC_GLOBAL_LOCALE.
15443
15444 to:
15445
15446 Upon successful completion, the uselocale() function shall return
15447 the locale handle from the previous call for the current thread,
15448 or LC_GLOBAL_LOCALE if there was no such previous call. Otherwise,
15449 uselocale() shall return (locale_t)0 and set errno to indicate the error.
15450

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

15451 *Rationale:* Austin Group Defect Report(s) applied: 334.
 15452 See <http://austingroupbugs.net/view.php?id=334>
 15453
 15454
 15455 **Change Number: XSH/TC1/D5/0702 [324]**
 15456
 15457
 15458 On Page: 2164 Line: 68355 Section: `utime()`
 15459
 15460 In the ERRORS section, for the [ENOTDIR] error, change from:
 15461
 15462 A component of the path prefix is not a directory, ...
 15463
 15464 to:
 15465
 15466 A component of the path prefix names an existing file that is neither
 15467 a directory nor a symbolic link to a directory, ...
 15468
 15469 *Rationale:* Austin Group Defect Report(s) applied: 324.
 15470 See <http://austingroupbugs.net/view.php?id=324>
 15471 This is an editorial issue clarifying the intent of the
 15472 standard.
 15473
 15474
 15475 **Change Number: XSH/TC1/D5/0703 [14]**
 15476
 15477
 15478 On Page: 2168 Line: 68459 Section: `vfprintf()`
 15479
 15480 In the SEE ALSO section, add a reference to XSH Section 2.5.
 15481
 15482 *Rationale:* Austin Group Defect Report(s) applied: 14.
 15483 See <http://austingroupbugs.net/view.php?id=14>
 15484 This is an editorial improvement
 15485
 15486
 15487 **Change Number: XSH/TC1/D5/0704 [14]**
 15488
 15489
 15490 On Page: 2170 Line: 68503 Section: `vfscanf()`
 15491
 15492 In the SEE ALSO section, add a reference to XSH Section 2.5.
 15493
 15494 *Rationale:* Austin Group Defect Report(s) applied: 14.
 15495 See <http://austingroupbugs.net/view.php?id=14>
 15496 This is an editorial improvement
 15497
 15498
 15499 **Change Number: XSH/TC1/D5/0705 [14]**
 15500
 15501
 15502 On Page: 2171 Line: 68540 Section: `vwprintf()`
 15503
 15504 In the SEE ALSO section, add a reference to XSH Section 2.5.
 15505
 15506 *Rationale:* Austin Group Defect Report(s) applied: 14.
 15507 See <http://austingroupbugs.net/view.php?id=14>
 15508 This is an editorial improvement
 15509
 15510
 15511 **Change Number: XSH/TC1/D5/0706 [14]**
 15512
 15513

15514 On Page: 2172 Line: 68581 Section: vfwscanf()
15515
15516 In the SEE ALSO section, add a reference to XSH Section 2.5.
15517
15518 *Rationale:* Austin Group Defect Report(s) applied: 14.
15519 See <http://austingroupbugs.net/view.php?id=14>
15520 This is an editorial improvement
15521
15522
15523 **Change Number: XSH/TC1/D5/0707 [421]**
15524
15525
15526 On Page: 2183 Line: 68781 Section: wait()
15527
15528 In the RETURN VALUE section, change from:
15529
15530 Otherwise, (pid_t)-1 shall be returned, ...
15531
15532 to:
15533
15534 Otherwise, -1 shall be returned, ...
15535
15536 *Rationale:* Austin Group Defect Report(s) applied: 421.
15537 See <http://austingroupbugs.net/view.php?id=421>
15538
15539
15540 **Change Number: XSH/TC1/D5/0708 [166]**
15541
15542
15543 On Page: 2185 Line: 68859 Section: wait()
15544
15545 In the EXAMPLES section, change from:
15546
15547 int status;
15548
15549 to:
15550
15551 int sav_errno = errno;
15552 int status;
15553
15554 *Rationale:* Austin Group Defect Report(s) applied: 166.
15555 See <http://austingroupbugs.net/view.php?id=166>
15556
15557
15558 **Change Number: XSH/TC1/D5/0709 [166]**
15559
15560
15561 On Page: 2186 Line: 68887 Section: wait()
15562
15563 In the EXAMPLES section, change from:
15564
15565
15566 }
15567
15568 to:
15569
15570 }
15571 errno = sav_errno;
15572 }
15573
15574 *Rationale:* Austin Group Defect Report(s) applied: 166.
15575 See <http://austingroupbugs.net/view.php?id=166>
15576

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

15577
 15578 **Change Number: XSH/TC1/D5/0710 [69]**
 15579
 15580
 15581 On Page: 2186 Line: 68919 Section: wait()
 15582
 15583 In the EXAMPLES section, insert before L68919 (before the final line of
 15584 the example code):
 15585
 15586 return 0; /* NOTREACHED */
 15587
 15588 *Rationale:* Austin Group Defect Report(s) applied: 69.
 15589 See <http://austingroupbugs.net/view.php?id=69>
 15590
 15591
 15592 **Change Number: XSH/TC1/D5/0711 [154]**
 15593
 15594
 15595 On Page: 2190 Line: 69071 Section: waitid()
 15596
 15597 In the DESCRIPTION section, change from:
 15598
 15599 WCONTINUED Status shall be returned for any child that was stopped and
 15600 has been continued.
 15601
 15602 to:
 15603
 15604 WCONTINUED Status shall be returned for any continued child process
 15605 whose status either has not been reported since it continued from a job
 15606 control stop or has been reported only by calls to waitid() with the
 15607 WNOWAIT flag set.
 15608
 15609 *Rationale:* Austin Group Defect Report(s) applied: 154.
 15610 See <http://austingroupbugs.net/view.php?id=154>
 15611
 15612
 15613 **Change Number: XSH/TC1/D5/0712 [154]**
 15614
 15615
 15616 On Page: 2190 Line: 69078 Section: waitid()
 15617
 15618 In the DESCRIPTION section, change from:
 15619
 15620 WSTOPPED Status shall be returned for any child that has stopped upon
 15621 receipt of a signal.
 15622
 15623 to:
 15624
 15625 WSTOPPED Status shall be returned for any child that has stopped upon
 15626 receipt of a signal, and whose status either has not been reported
 15627 since it stopped or has been reported only by calls to waitid() with
 15628 the WNOWAIT flag set.
 15629
 15630 *Rationale:* Austin Group Defect Report(s) applied: 154.
 15631 See <http://austingroupbugs.net/view.php?id=154>
 15632
 15633
 15634 **Change Number: XSH/TC1/D5/0713 [153]**
 15635
 15636
 15637 On Page: 2190 Line: 69084 Section: waitid()
 15638
 15639 In the DESCRIPTION section (final paragraph), change from:

15640
15641 ... with the status of the process. The si_signo member shall always be
15642 equal to SIGCHLD.
15643
15644 to:
15645
15646 ... with the status of the process; the si_signo member shall be set
15647 equal to SIGCHLD. If waitid() returns because WNOHANG was specified and
15648 status is not available for any process specified by idtype and id, then
15649 the si_signo and si_pid members of the structure pointed to by infop
15650 shall be set to zero and the values of other members of the structure
15651 are unspecified.
15652
15653 *Rationale:* Austin Group Defect Report(s) applied: 153.
15654 See <http://austingroupbugs.net/view.php?id=153>
15655
15656
15657 **Change Number: XSH/TC1/D5/0714 [88]**
15658
15659
15660 On Page: 2195 Line: 69147 Section: wctomb()
15661
15662 In the SYNOPSIS section, change from:
15663
15664 #include <stdio.h>
15665
15666 to:
15667
15668 #include <wchar.h>
15669
15670 *Rationale:* Austin Group Defect Report(s) applied: 88.
15671 See <http://austingroupbugs.net/view.php?id=88>
15672
15673
15674 **Change Number: XSH/TC1/D5/0715 [1050]**
15675
15676
15677 On Page: 2195 Line: 69168 Section: wctomb()
15678
15679 In the DESCRIPTION section, add a new paragraph at the end of the
15680 DESCRIPTION:
15681
15682 The wctomb() function shall not change the setting of errno if
15683 successful.
15684
15685 *Rationale:* Austin Group Defect Report(s) applied: 105.
15686 See <http://austingroupbugs.net/view.php?id=105>
15687
15688
15689 **Change Number: XSH/TC1/D5/0716 [294]**
15690
15691
15692 On Page: 2197 Line: 69221 Section: wcscasecmp()
15693
15694 In the DESCRIPTION section, change from:
15695
15696 When the LC_CTIME category of the current locale is from the POSIX locale,
15697 these functions shall behave as if the strings had been converted to
15698 lowercase and then a byte comparison performed. Otherwise, the results
15699 are unspecified.
15700
15701 The information for wcscasecmp_l() and wcsncasecmp_l() about the case
15702 of the characters comes from the locale represented by locale.

15703

15704 to:

15705

15706 The `wscasecmp()` and `wcsncasecmp()` functions use the current locale to
 15707 determine the case of the wide characters.

15708

15709 The `wscasecmp_l()` and `wcsncasecmp_l()` functions use the locale
 15710 represented by locale to determine the case of the wide characters.

15711

15712 When the `LC_CTYPE` category of the locale being used is from the POSIX
 15713 locale, these functions shall behave as if the wide-character strings
 15714 had been converted to lowercase and then a comparison of wide-character
 15715 codes performed. Otherwise, the results are unspecified.

15716

15717 *Rationale:* Austin Group Defect Report(s) applied: 294.

15718 See <http://austingroupbugs.net/view.php?id=294>

15719

15720

15721 **Change Number: XSH/TC1/D5/0717 [283]**

15722

15723

15724 On Page: 2197 Line: 69225 Section: `wscasecmp()`

15725

15726 In the DESCRIPTION section, add a new paragraph to the end of the section:

15727

15728 The behavior is undefined if the locale argument to `wscasecmp_l()` or
 15729 `wcsncasecmp_l()` is the special locale object `LC_GLOBAL_LOCALE` or is not
 15730 a valid locale object handle.

15731

15732 *Rationale:* Austin Group Defect Report(s) applied: 283.

15733 See <http://austingroupbugs.net/view.php?id=283>

15734

15735

15736 **Change Number: XSH/TC1/D5/0718 [283]**

15737

15738

15739 On Page: 2197 Line: 69236 Section: `wscasecmp()`

15740

15741 In the ERRORS section, change from:

15742

15743 The `wscasecmp_l()` and `wcsncasecmp_l()` functions may fail if:

15744

15745 [EINVAL] locale is not a valid locale object handle.

15746

15747 to:

15748

15749 No errors are defined.

15750

15751 *Rationale:* Austin Group Defect Report(s) applied: 283.

15752 See <http://austingroupbugs.net/view.php?id=283>

15753

15754

15755 **Change Number: XSH/TC1/D5/0719 [302]**

15756

15757

15758 On Page: 2202 Line: 69364 Section: `wscoll()`

15759

15760 In the DESCRIPTION section, change from:

15761

15762 ... current locale of the process ...

15763

15764 to:

15765

15766 ... current locale ...
15767
15768 *Rationale:* Austin Group Defect Report(s) applied: 302.
15769 See <http://austingroupbugs.net/view.php?id=302>
15770 These changes were overlooked during the revision when
15771 per-thread locales were added.
15772
15773
15774 **Change Number: XSH/TC1/D5/0720** [283]
15775
15776
15777 On Page: 2202 Line: 69368 Section: wscoll()
15778
15779 In the DESCRIPTION section, add a new paragraph to the end of the section:
15780
15781 [CX]The behavior is undefined if the locale argument to wscoll_l()
15782 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
15783 object handle.[/CX]
15784
15785 *Rationale:* Austin Group Defect Report(s) applied: 283.
15786 See <http://austingroupbugs.net/view.php?id=283>
15787
15788
15789 **Change Number: XSH/TC1/D5/0721** [283]
15790
15791
15792 On Page: 2202 Line: 69380 Section: wscoll()
15793
15794 In the ERRORS section, delete:
15795
15796 The wscoll_l() function may fail if:
15797
15798 [EINVAL] locale is not a valid locale object handle.
15799
15800 *Rationale:* Austin Group Defect Report(s) applied: 283.
15801 See <http://austingroupbugs.net/view.php?id=283>
15802
15803
15804 **Change Number: XSH/TC1/D5/0722** [109,105]
15805
15806
15807 On Page: 2219 Line: 69818 Section: wcsnrtoombs()
15808
15809 In the DESCRIPTION section, change from:
15810
15811 [CX]The wcsrtoombs() function need not be thread-safe ... [/CX]
15812
15813 to:
15814
15815 [CX]The wcsnrtoombs() and wcsrtoombs() functions need not be thread-safe
15816 ... [/CX]
15817
15818 The wcsrtoombs() function shall not change the setting of errno if
15819 successful.
15820
15821 *Rationale:* Austin Group Defect Report(s) applied: 109,105.
15822 See <http://austingroupbugs.net/view.php?id=109>
15823 See <http://austingroupbugs.net/view.php?id=105>
15824
15825
15826 **Change Number: XSH/TC1/D5/0723** [302]
15827
15828

IEEE Std 1003.1™-2008/Cor 1-2013
 IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
 Base Specifications, Issue 7—Technical Corrigendum 1

15829 On Page: 2224 Line: 69988 Section: wctod()
 15830
 15831 In the DESCRIPTION section, change from:
 15832
 15833 ... the locale of the process ...
 15834
 15835 to:
 15836
 15837 ... the current locale ...
 15838
 15839 *Rationale:* Austin Group Defect Report(s) applied: 302.
 15840 See <http://austingroupbugs.net/view.php?id=302>
 15841 These changes were overlooked during the revision when
 15842 per-thread locales were added.
 15843
 15844
 15845 **Change Number: XSH/TC1/D5/0724 [105]**
 15846
 15847
 15848 On Page: 2224 Line: 69996 Section: wctod()
 15849
 15850 In the DESCRIPTION section, remove the CX shading from the paragraphs:
 15851
 15852 The wctod() function shall not change the setting of errno if successful.
 15853
 15854 Since 0 is returned on error and is also a valid return on success, an
 15855 application wishing to check for error situations should set errno to 0,
 15856 then call wctod(), wctof(), or wctold(), then check errno.
 15857
 15858 and then change the first paragraph:
 15859
 15860 The wctod() function shall not change the setting of errno if successful.
 15861
 15862 to:
 15863
 15864 These functions shall not change the setting of errno if successful.
 15865
 15866 *Rationale:* Austin Group Defect Report(s) applied: 105.
 15867 See <http://austingroupbugs.net/view.php?id=105>
 15868
 15869
 15870 **Change Number: XSH/TC1/D5/0725 [105]**
 15871
 15872
 15873 On Page: 2231 Line: 70204 Section: wcstol()
 15874
 15875 In the DESCRIPTION section, remove the CX shading from the paragraphs:
 15876
 15877 These functions shall not change the setting of errno if successful.
 15878
 15879 Since 0, {LONG_MIN} or {LLONG_MIN} and {LONG_MAX} or {LLONG_MAX} are
 15880 returned on error and are also valid returns on success, an application
 15881 wishing to check for error situations should set errno to 0, then call
 15882 wcstol() or wcstoll(), then check errno.
 15883
 15884 *Rationale:* Austin Group Defect Report(s) applied: 105.
 15885 See <http://austingroupbugs.net/view.php?id=105>
 15886
 15887
 15888 **Change Number: XSH/TC1/D5/0726 [109]**
 15889
 15890
 15891 On Page: 2235 Line: 70283 Section: wcstombs()

15892
15893 In the DESCRIPTION section, delete:
15894
15895 The `wcstombs()` function need not be thread-safe.
15896
15897 *Rationale:* Austin Group Defect Report(s) applied: 109.
15898 See <http://austingroupbugs.net/view.php?id=109>
15899
15900
15901 **Change Number: XSH/TC1/D5/0727 [105]**
15902
15903
15904 On Page: 2238 Line: 70365 Section: `wcstoul()`
15905
15906 In the DESCRIPTION section, remove the CX shading from the paragraphs:
15907
15908 The `wcstoul()` function shall not change the setting of `errno` if successful.
15909
15910 Since 0, `{ULONG_MAX}`, and `{ULLONG_MAX}` are returned on error and 0 is
15911 also a valid return on success, an application wishing to check for error
15912 situations should set `errno` to 0, then call `wcstoul()` or `wcstoul()`,
15913 then check `errno`.
15914
15915 and then change the first paragraph from:
15916
15917 The `wcstoul()` function shall not change the setting of `errno` if successful.
15918
15919 to:
15920
15921 These functions shall not change the setting of `errno` if successful.
15922
15923 *Rationale:* Austin Group Defect Report(s) applied: 105.
15924 See <http://austingroupbugs.net/view.php?id=105>
15925
15926
15927 **Change Number: XSH/TC1/D5/0728 [302]**
15928
15929
15930 On Page: 2242 Line: 70463 Section: `wcsxfrm()`
15931
15932 Change from:
15933
15934 ... the locale of the process ...
15935
15936 to:
15937
15938 ... the current locale ...
15939
15940 *Rationale:* Austin Group Defect Report(s) applied: 302.
15941 See <http://austingroupbugs.net/view.php?id=302>
15942 These changes were overlooked during the revision when
15943 per-thread locales were added.
15944
15945
15946 **Change Number: XSH/TC1/D5/0729 [283]**
15947
15948
15949 On Page: 2242 Line: 70469 Section: `wcsxfrm()`
15950
15951 In the DESCRIPTION section, add a new paragraph to the end of the section:
15952
15953 [CX]The behavior is undefined if the locale argument to `wcsxfrm_l()`
15954 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale