

Technical Corrigendum

Base Specifications, Issue 7 – Technical Corrigendum 1

IEEE Computer Society

and

The Open Group



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

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.

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.

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 Lefèvre
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 Lefèvre
Wojtek Lerch
Scott Lurndal
Roger Marquis
Matt McCutchen
Esmaeil 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*

Introduction

This introduction is not part of IEEE Std 1003.1 TM -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.

Contents

1. Changes to Base Definitions.....	2
2. Changes to System Interfaces.....	27
3. Changes to Shell and Utilities	260
4. Changes to Rationale.....	311

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/TPR/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.

1. Changes to Base Definitions

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

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

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

On Page: ix Line: 0 Section: Typographical Conventions

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

Filename

to:

Filename, Pathname

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

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

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

On Page: ix Line: none Section: Typographical Conventions

Change:

'\'

in Note 2 to:

'\''

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

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

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

On Page: 9 Line: 247 Section: 1.7.1 Codes

Change from:

[MX]IEC 60559 Floating-Point

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

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

to:

[MX]IEC 60559 Floating-Point

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

[MXX] IEC 60559 Floating-Point Extension
The functionality described is part of the IEC 60559 Floating-Point
option, but is an extension to the ISO C standard. [/MXX]

Rationale: Austin Group Defect Report(s) applied: 68.
See <http://austingroupbugs.net/view.php?id=68>

Change Number: XBD/TC1/D5/0002 [291]

On Page: 21 Line: 730 Section: 2.1.5.1 Subprofiling Considerations

Change from:

filename argument

to:

pathname argument

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

Change Number: XBD/TC1/D5/0003 [82]

On Page: 24 Line: 830,839 Section: 2.1.5.2 XSI Option Groups

Add to the list of options (in alphabetical order):

_POSIX_THREAD_ROBUST_PRIO_INHERIT

_POSIX_THREAD_ROBUST_PRIO_PROTECT

Rationale: Austin Group Defect Report(s) applied: 82.
See <http://austingroupbugs.net/view.php?id=82>

Change Number: XBD/TC1/D5/0004 [291]

On Page: 54 Line: 1647 Section: 3.136 Dot

Change from:

In the context of naming files, the filename consisting of a single dot
character ('.').

to:

In the context of naming files, the filename consisting of a single
<period> character ('.').

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

Change Number: XBD/TC1/D5/0005 [291]

On Page: 55 Line: 1650 Section: 3.137 Dot-Dot

Change from:

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

to:

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

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

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

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

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

Change from:

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

to:

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

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

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

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

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

On Page: 60 Line: 1782 Section: 3.170 Filename

Change from:

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

to:

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

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

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

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

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

sections:

3.2xx Portable Filename

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

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

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

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

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

Add a sentence at the end of the section:

See also XBD Section 3.266 Pathname.

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

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

On Page: 97 Line: 2670 Section: 3.396 Thread

Change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 181.
See <http://austingroupbugs.net/view.php?id=181>

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

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

Change from:

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

to:

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

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

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

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

Delete the last paragraph in the section:

POSIX.1-2008 does not provide a way to open a directory for searching. It is unspecified whether directory search permission is granted based on the file access modes of the directory's file descriptor or on the mode of the directory at the time the directory is searched.

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

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

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

Change from:

Portable filenames shall not have the <hyphen> character as the first character since this may cause problems when filenames are passed as command line arguments.

to:

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

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

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

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

Change from:

A pathname that begins with two successive <slash> characters may be interpreted in an implementation-defined manner, although ...

to:

If a pathname begins with two successive <slash> characters, the first component following the leading <slash> characters may be interpreted in an implementation-defined manner, although ...

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

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

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

Change from:

Escape Sequences represent non-graphic characters.

to:

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

Rationale: Austin Group Defect Report(s) applied: 340.

See <http://austingroupbugs.net/view.php?id=340>

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

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

Change from:

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

to:

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

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

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

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

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

After L3589 insert a new bullet:

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

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

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

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

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

Change from:

The POSIX locale contains the characters in Table 6-1 (on page 125), ...

to:

The POSIX locale contains the characters in Table 6-1 (on page 125)
and Table 6-2 (on page 130), ...

Rationale: Austin Group Defect Report(s) applied: 216.
See <http://austingroupbugs.net/view.php?id=216>

Change Number: XBD/TC1/D5/0021 [291]

On Page: 128 Line: 3619 Section: 6.2 Character Encoding

Add a sentence:

Likewise, the byte values used to encode <period> and <slash> shall not
occur as part of any other character in any locale.

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

Change Number: XBD/TC1/D5/0022 [216]

On Page: 129 Line: 3661 Section: 6.4 Character Set Description File

Change from:

... the symbolic names and their corresponding encoding values shall be
included in the file.

to:

... the symbolic names and their corresponding encoding values shall be
included in the file. The encoding values shall each be represented in
a single byte.

Rationale: Austin Group Defect Report(s) applied: 216.
See <http://austingroupbugs.net/view.php?id=216>

Change Number: XBD/TC1/D5/0023 [302]

On Page: 135 Line: 3820-3822 Section: 7.1 General

Change from:

The behavior of some of the C-language functions defined in the System
Interfaces volume of POSIX.1-2008 shall also be modified based on the
current locale, as defined by the last call to `setlocale()`.

to:

The behavior of some of the C-language functions defined in the System
Interfaces volume of POSIX.1-2008 shall also be modified based on a locale
selection. The locale to be used by these functions can be selected in
the following ways:

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

2. For functions that do not take a locale object as an argument, the current locale for the thread can be set by calling `uselocale()` or the global locale for the process can be set by calling `setlocale()`. Such functions shall use the current locale of the calling thread if one has been set for that thread; otherwise, they shall use the global 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: XBD/TC1/D5/0024 [302]

On Page: 135 Line: 3839 Section: 7.1 General

Change from:

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

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

the value of the corresponding environment variable is used.

to:

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

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

or:

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

the value of the corresponding environment variable is used.

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

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

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

... the acceptable affirmative response ...

to:

... acceptable affirmative responses ...

On L5144-5145 in the description of the noexpr keyword, change from:

... the acceptable negative response ...

to:

... acceptable negative responses ...

Rationale: Austin Group Defect Report(s) applied: 22,427.

See <http://austingroupbugs.net/view.php?id=22>

See <http://austingroupbugs.net/view.php?id=427>

Change Number: XBD/TC1/D5/0026 [167]

On Page: 173 Line: 5476 Section: 8.1 Environment Variable Definition

Change from:

... manipulating the environ variable ...

to:

... assigning a new value to the environ variable ...

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

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

Change Number: XBD/TC1/D5/0027 [167]

On Page: 173 Line: 5478 Section: 8.1 Environment Variable Definition

Add a new paragraph after L5478:

If the application modifies the pointers to which environ points, the behavior of all interfaces described in the System Interfaces volume of POSIX.1-2008 is undefined.

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

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

Change Number: XBD/TC1/D5/0028 [291]

On Page: 175 Line: 5566 Section: 8.2 Internationalization Variables

In the description of NLSPATH, change from:

filename

to:

pathname

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

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

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

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

Change from:

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

to:

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

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

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

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

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

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

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

to:

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

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

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

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

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

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

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

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

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

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

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

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

At L5995, add a new paragraph:

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

the first and last list element are the same single-character element of <period>, <equals-sign>, or <colon>, then it is unspecified whether the bracket expression will be treated as a collating symbol, equivalence class, or character class, respectively; treated as a matching list expression; or rejected as an error.

Rationale: Austin Group Defect Report(s) applied: 305,427.
See <http://austingroupbugs.net/view.php?id=305>
See <http://austingroupbugs.net/view.php?id=427>

Change Number: XBD/TC1/D5/0033 [134]

On Page: 197 Line: 6402 Section: 10.1 Directory Structure and Files

Change from:

/dev/null An infinite data source and data sink. ...

to:

/dev/null An empty data source and infinite data sink. ...

Rationale: Austin Group Defect Report(s) applied: 134.
See <http://austingroupbugs.net/view.php?id=134>
Improve the description of /dev/null.

Change Number: XBD/TC1/D5/0034 [79]

On Page: 201 Line: 6531-6542 Section: 11.1.4 Terminal Access Control

On L6531 change from:

... if the reading process is ignoring or blocking the SIGTTIN signal, ...

to:

... if the reading process is ignoring the SIGTTIN signal or the reading thread is blocking the SIGTTIN signal, ...

On L6539 change from:

... and the process is ignoring or blocking the SIGTTOU signal, ...

to:

... and the process is ignoring the SIGTTOU signal or the writing thread is blocking the SIGTTOU signal, ...

On L6540 change from:

If TOSTOP is set, and the process group of the writing process is orphaned, and the writing process is not ignoring or blocking the SIGTTOU signal, ...

to:

If TOSTOP is set, the process group of the writing process is orphaned, the writing process is not ignoring the SIGTTOU signal, and the writing thread is not blocking the SIGTTOU signal, ...

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

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

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

Change from:

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

to:

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

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

The text can mean two things:

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

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

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

Change from:

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

to:

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

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

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

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

In Guideline 5 change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 171.
See <http://austingroupbugs.net/view.php?id=171>

Change Number: XBD/TC1/D5/0038 [98]

On Page: 220 Line: 7234 Section: <aio.h>

In the DESCRIPTION section, add FSC|SIO shading to the declaration
of aio_fsync().

Rationale: Austin Group Defect Report(s) applied: 98.
See <http://austingroupbugs.net/view.php?id=98>

Change Number: XBD/TC1/D5/0039 [291]

On Page: 231 Line: 7575 Section: <dirent.h>

In the DESCRIPTION section, change from:

d_name[] Name of entry.

to:

d_name[] Filename string of entry.

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

Change Number: XBD/TC1/D5/0040 [291]

On Page: 231 Line: 7577 Section: <dirent.h>

In the DESCRIPTION section, change from:

The character array d_name is of unspecified size, but the number of
bytes preceding the terminating null byte shall not exceed {NAME_MAX}.

to:

The array d_name is of unspecified size, but shall contain a filename
of at most {NAME_MAX} bytes followed by a terminating null byte.

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

Change Number: XBD/TC1/D5/0041 [291]

On Page: 232 Line: 7612 Section: <dirent.h>

In the RATIONALE section, change from:

... number of characters provided matches (or only slightly exceeds)
the length of the filename.

to:

... number of bytes provided matches (or only slightly exceeds) the length of the filename string.

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

Change Number: XBD/TC1/D5/0042 [206]

On Page: 232 Line: 7629 Section: <dirent.h>

In the CHANGE HISTORY section, change from:

... `fopendir()` ...

to:

... `fdopendir()` ...

Rationale: Austin Group Defect Report(s) applied: 206.
See <http://austingroupbugs.net/view.php?id=206>

Change Number: XBD/TC1/D5/0043 [324]

On Page: 236 Line: 7739 Section: <errno.h>

In the DESCRIPTION section, change from:

[ENOTDIR] Not a directory.

to:

[ENOTDIR] Not a directory or a symbolic link to a directory.

Rationale: Austin Group Defect Report(s) applied: 324.
See <http://austingroupbugs.net/view.php?id=324>

Change Number: XBD/TC1/D5/0044 [274]

On Page: 239 Line: 7848 Section: <fcntl.h>

In the DESCRIPTION section, after:

The <fcntl.h> header shall define the following symbolic constants for use as the file access modes for `open()`, `openat()`, and `fcntl()`.

add:

The values shall be unique, except that `O_EXEC` and `O_SEARCH` may have equal values.

Rationale: Austin Group Defect Report(s) applied: 274.
See <http://austingroupbugs.net/view.php?id=274>

Change Number: XBD/TC1/D5/0045 [78,432]

On Page: 240 Line: 7874-7879 Section: <fcntl.h>

In the DESCRIPTION section, delete L7874 to L7875.

The <fcntl.h> header shall define the following symbolic constants as values for the flag used by open() and openat():

In the DESCRIPTION section, move L7876 to L7879:

O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall be set to close the file descriptor upon execution of an exec family function.

O_DIRECTORY Fail if not a directory.

O_NOFOLLOW Do not follow symbolic links.

into the section at L7829 to L7834, so that it reads:

The <fcntl.h> header shall define the following symbolic constants as file creation flags for use in the oflag value to open() and openat(). The values shall be bitwise-distinct and shall be suitable for use in #if preprocessing directives.

O_CLOEXEC The FD_CLOEXEC flag associated with the new descriptor shall be set to close the file descriptor upon execution of an exec family function.

O_CREAT Create file if it does not exist.

O_DIRECTORY Fail if not a directory.

O_EXCL Exclusive use flag.

O_NOCTTY Do not assign controlling terminal.

O_NOFOLLOW Do not follow symbolic links.

O_TRUNC Truncate flag.

Rationale: Austin Group Defect Report(s) applied: 78,432.

See <http://austingroupbugs.net/view.php?id=78>

See <http://austingroupbugs.net/view.php?id=432>

Change Number: XBD/TC1/D5/0046 [346]

On Page: 249 Line: 8198 Section: <float.h>

In the DESCRIPTION section, add a paragraph within the *_MAX_EXP bullet:

[CX]Additionally, FLT_MAX_EXP shall be at least as large as FLT_MANT_DIG, DBL_MAX_EXP shall be at least as large as DBL_MANT_DIG, and LDBL_MAX_EXP shall be at least as large as LDBL_MANT_DIG; which has the effect that FLT_MAX, DBL_MAX, and LDBL_MAX are integral.[/CX]

Rationale: Austin Group Defect Report(s) applied: 346.

See <http://austingroupbugs.net/view.php?id=346>

Change Number: XBD/TC1/D5/0047 [346]

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
1054 **Change Number: XBD/TC1/D5/0048 [403]**
1055
1056
1057 On Page: 254 Line: 8333 Section: <ftw.h>
1058
1059 In the DESCRIPTION section, change from:
1060
1061 FTW_F File.
1062
1063 to:
1064
1065 FTW_F Non-directory file.
1066
1067 *Rationale:* Austin Group Defect Report(s) applied: 403.
1068 See <http://austingroupbugs.net/view.php?id=403>
1069
1070
1071 **Change Number: XBD/TC1/D5/0049 [24]**
1072
1073
1074 On Page: 258 Line: 8441-8442 Section: <grp.h>
1075
1076 In the DESCRIPTION, add XSI shading to endgrent() and getgrent().
1077
1078 *Rationale:* Austin Group Defect Report(s) applied: 24.
1079 See <http://austingroupbugs.net/view.php?id=24>
1080
1081
1082 **Change Number: XBD/TC1/D5/0050 [211]**
1083
1084
1085 On Page: 261 Line: 8510- Section: <inttypes.h>
1086
1087 In the DESCRIPTION section change from:
1088
1089 The <inttypes.h> header shall define at least the following type:
1090
1091 imaxdiv_t Structure type that is the type of the value returned by the
1092 imaxdiv() function.

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

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]**

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

Change Number: XBD/TC1/D5/0073 [327]

On Page: 399 Line: 13420 Section: <sys/types.h>

In the DESCRIPTION section, change from:

time_t and clock_t shall be integer or real-floating types.

to:

clock_t shall be an integer or real-floating type. [CX]time_t shall be an integer type.[/CX]

Rationale: Austin Group Defect Report(s) applied: 327.

See <http://austingroupbugs.net/view.php?id=327>

Change Number: XBD/TC1/D5/0074 [355]

On Page: 403 Line: 13517 Section: <sys/un.h>

In the DESCRIPTION section, change from:

Values of this type shall be cast by applications to struct sockaddr for use with socket functions.

to:

Pointers to this type shall be cast by applications to struct sockaddr * for use with socket functions.

Rationale: Austin Group Defect Report(s) applied: 355.

See <http://austingroupbugs.net/view.php?id=355>

Change Number: XBD/TC1/D5/0075 [357,427]

On Page: 418 Line: 13996 Section: <tgmath.h>

In the DESCRIPTION section, in the table at L13996-13997, scalbln() should sort before scalbn().

Rationale: Austin Group Defect Report(s) applied: 357,427.

See <http://austingroupbugs.net/view.php?id=357>

See <http://austingroupbugs.net/view.php?id=427>

Change Number: XBD/TC1/D5/0076 [212]

On Page: 422 Line: 14153 Section: <time.h>

In the DESCRIPTION section, add the CX margin marker notation and shading to L14153-14155. (The <time.h> header.... TIMER_ABSTIME ...)

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

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

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

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

In the DESCRIPTION section, change from:

... a progrm defines ...

to:

... a program defines ...

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

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

This is an editorial correction.

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

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

In the DESCRIPTION section, after:

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

add (to the same paragraph):

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

At L14455 after:

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

add:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Rationale: Austin Group Defect Report(s) applied: 360.

See <http://austingroupbugs.net/view.php?id=360>

Some implementations have had a declaration for ctermid() and this should still be allowed in Issue 7. Having ctermid() and pthread_atfork() prototypes in <unistd.h>, however, are namespace pollution issues that should be corrected in the next revision.

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

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

In the DESCRIPTION section, in the wcpcpy() and wcpncpy() prototype declarations change from:

restrict*

to:

*restrict

Rationale: Austin Group Defect Report(s) applied: 380.

See <http://austingroupbugs.net/view.php?id=380>

2. Changes to System Interfaces

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

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

Change Number: XSH/TC1/D5/0001 [316]

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

Change from:

The ERRORS section on each reference page specifies which error conditions shall be detected by all implementations ("shall fail") and which may be optionally detected by an implementation ("may fail"). If no error condition is detected, the action requested shall be successful.

to:

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

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

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

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

For the [ENOTDIR] description, change from:

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

to:

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

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

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

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

Change from:

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

to:

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

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

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

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

Change from:

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

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

to:

When any stop signal (SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU) is generated for a process or thread, all pending SIGCONT signals for that process or any of the threads within that process shall be discarded. Conversely, when SIGCONT is generated for a process or thread, all pending stop signals for that process or any of the threads within that process shall be discarded. When SIGCONT is generated for a process that is stopped, the process shall be continued, even if the SIGCONT signal is ignored by the process or is blocked by all threads within the process and there are no threads in a call to a `sigwait()` function selecting SIGCONT. If SIGCONT is blocked by all threads within the process, there are no threads in a call to a `sigwait()` function selecting SIGCONT, and SIGCONT is not ignored by the process, the SIGCONT signal shall remain pending on the process until it is either unblocked by a thread or a thread calls a `sigwait()` function selecting SIGCONT, or a stop signal is generated for the process or any of the threads within the process.

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

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

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

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

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

to:

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

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

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

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

Rationale: Austin Group Defect Report(s) applied: 66.
See <http://austingroupbugs.net/view.php?id=66>

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

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

Change Number: XSH/TC1/D5/0011 [345,428]

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

Add a new paragraph:

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

Rationale: Austin Group Defect Report(s) applied: 345,428.

See <http://austingroupbugs.net/view.php?id=345>

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

Change Number: XSH/TC1/D5/0012 [109,429]

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

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

On L17511, change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 109,429.

See <http://austingroupbugs.net/view.php?id=109>

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

Change Number: XSH/TC1/D5/0013 [89]

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

On L17536-L17538, change from:

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

to:

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

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

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

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 `fdatsync()`; 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 `fdatsync()`, 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 `aio_cb` 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

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 [EOVERFLOW] 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

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

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

In the RETURN VALUE section, change from:

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

to:

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

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

Rationale: Austin Group Defect Report(s) applied: 68.
See <http://austingroupbugs.net/view.php?id=68>

Change Number: XSH/TC1/D5/0041 [75]

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

In the DESCRIPTION section, change from:

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

to:

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

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

Change Number: XSH/TC1/D5/0042 [146]

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

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

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

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

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

Change Number: XSH/TC1/D5/0046 [346]

On Page: 645 Line: 21865 Section: `ceil()`

For `ceil()`, delete P645 L21865-21868 (DESCRIPTION), L21874-21875 (RETURN VALUE), and replace L21877-21882 (ERRORS) with:

No errors are defined.

At L21870 (RETURN VALUE), add:

[MX]The result shall have the same sign as `x`. [/MX]

At L21886 (APPLICATION USAGE), change "an int or long" to "an `intmax_t`", and replace L21889-21892 (APPLICATION USAGE) with:

These functions may raise the inexact floating-point exception if the result differs in value from the argument.

Rationale: Austin Group Defect Report(s) applied: 346.

See <http://austingroupbugs.net/view.php?id=346>

Change Number: XSH/TC1/D5/0047 [324]

On Page: 653 Line: 22116 Section: `chdir()`

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

A component of the pathname is not a directory; ...

to:

A component of the pathname names an existing file that is neither a directory nor a symbolic link to a directory; ...

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

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

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

Change Number: XSH/TC1/D5/0048 [300]

On Page: 655 Line: 22172-22173 Section: `chmod()`

Delete the paragraph at L22172-22173 from the DESCRIPTION section of `chmod()`.

Rationale: Austin Group Defect Report(s) applied: 300.

See <http://austingroupbugs.net/view.php?id=300>

Change Number: XSH/TC1/D5/0049 [461]

On Page: 655 Line: 22186 Section: `chmod()`

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

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()

In the DESCRIPTION section, add to the end of the section:

[CX]The clearerr() function shall not change the setting of errno if stream is valid.[/CX]

Rationale: Austin Group Defect Report(s) applied: 401.
See <http://austingroupbugs.net/view.php?id=401>

Change Number: XSH/TC1/D5/0058 [106]

On Page: 668 Line: 22669 Section: clock_gettime()

In the ERRORS section, add after the existing [EINVAL] error:

The clock_gettime() function shall fail if:

[EOVERFLOW] The number of seconds will not fit in an object of type time_t.

Rationale: Austin Group Defect Report(s) applied: 106.
See <http://austingroupbugs.net/view.php?id=106>

Change Number: XSH/TC1/D5/0059 [419]

On Page: 677 Line: 22921 Section: close()

In the ERRORS section, change from:

[EBADF] The fildes argument is not a valid file descriptor.

to:

[EBADF] The fildes argument is not an open file descriptor.

Rationale: Austin Group Defect Report(s) applied: 419.
See <http://austingroupbugs.net/view.php?id=419>

Change Number: XSH/TC1/D5/0060 [149]

On Page: 678 Line: 22962 Section: close()

In the APPLICATION USAGE section, add a new paragraph at the end of the section:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls close() on an arbitrary integer risks non-conforming behavior, and close() can only portably be used on file descriptor values that the application has obtained through explicit actions, as well as the three file descriptors corresponding to the standard file streams. In multi-threaded parent applications, the practice of calling close() in a loop after fork() and before an exec call in order to avoid a race condition of leaking an unintended file descriptor into a child process, is therefore unsafe, and the race should instead be combatted by opening all file descriptors with the FD_CLOEXEC bit set unless the file descriptor is intended to

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

Change Number: XSH/TC1/D5/0064 [291]

On Page: 702 Line: 23739-23741 Section: creat()

In the EXAMPLES section, change both instances of "filename" to "pathname".

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

Change Number: XSH/TC1/D5/0065 [75,428]

On Page: 713 Line: 24015 Section: ctermid()

In the RETURN VALUE section, change from:

If s is a null pointer, the string shall be generated in an area that may be static (and therefore may be overwritten by each call), the address of which shall be returned. Otherwise, ...

to:

If s is a null pointer, the string shall be generated in a area that may be static, the address of which shall be returned. The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten by a subsequent call to ctermid(). If s is not a null pointer, ...

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

Change Number: XSH/TC1/D5/0066 [321,428]

On Page: 715 Line: 24085 Section: ctime()

In the DESCRIPTION section, change from:

Unlike ctime(), the thread-safe version of ctime_r() is not required to set tzname.

to:

Unlike ctime(), the ctime_r() function is not required to set tzname. If ctime_r() does not set tzname, it shall not set daylight and shall not set timezone.

Rationale: Austin Group Defect Report(s) applied: 321,428.
See <http://austingroupbugs.net/view.php?id=321>
See <http://austingroupbugs.net/view.php?id=428>

Change Number: XSH/TC1/D5/0067 [422]

On Page: 723 Line: 24328 Section: dirfd()

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 dlclose(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

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

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

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

RETURN VALUE

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

ERRORS

No errors are defined.

EXAMPLES

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

```
#include <dlfcn.h>
int eret;
void *mylib;
...
/* Open a dynamic library and then close it ... */
mylib = dlopen("mylib.so", RTLD_LOCAL | RTLD_LAZY);
...
eret = dlclose(mylib);
...
```

APPLICATION USAGE

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

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

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

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

accessed. The mode parameter governs when these relocations take place and may have the following values:

RTLD_LAZY Relocations shall be performed at an implementation-defined time, ranging from the time of the `dlopen()` call until the first reference to a given symbol occurs. Specifying **RTLD_LAZY** should improve performance on implementations supporting dynamic symbol binding since a process might not reference all of the symbols in an executable object file. And, for systems supporting dynamic symbol resolution for normal process execution, this behavior mimics the normal handling of process execution.

RTLD_NOW All necessary relocations shall be performed when the executable object file is first loaded. This may waste some processing if relocations are performed for symbols that are never referenced. This behavior may be useful for applications that need to know that all symbols referenced during execution will be available before `dlopen()` returns.

Any executable object file loaded by `dlopen()` that requires relocations against global symbols can reference the symbols in the original process image file, any executable object files loaded at program start-up, from the initial process image itself, from any other executable object file included in the same `dlopen()` invocation, and any executable object files that were loaded in any `dlopen()` invocation and which specified the **RTLD_GLOBAL** flag. To determine the scope of visibility for the symbols loaded with a `dlopen()` invocation, the mode parameter should be a bitwise-inclusive OR with one of the following values:

RTLD_GLOBAL The executable object file's symbols shall be made available for relocation processing of any other executable object file. In addition, symbol lookup using `dlopen(NULL, mode)` and an associated `dlsym()` allows executable object files loaded with this mode to be searched.

RTLD_LOCAL The executable object file's symbols shall not be made available for relocation processing of any other executable object file.

If neither **RTLD_GLOBAL** nor **RTLD_LOCAL** are specified, the default behavior is unspecified.

If an executable object file is specified in multiple `dlopen()` invocations, mode is interpreted at each invocation.

If **RTLD_NOW** has been specified, all relocations shall have been completed rendering further **RTLD_NOW** operations redundant and any further **RTLD_LAZY** operations irrelevant.

If **RTLD_GLOBAL** has been specified, the executable object file shall maintain the **RTLD_GLOBAL** status regardless of any previous or future specification of **RTLD_LOCAL**, as long as the executable object file remains in the address space (see `dlclose()`).

Symbols introduced into the process image through calls to `dlopen()` may be used in relocation activities. Symbols so introduced may duplicate symbols already defined by the program or previous `dlopen()` operations. To resolve the ambiguities such a situation might present, the resolution of a symbol reference to symbol definition is based on a symbol resolution order. Two such resolution orders are defined: load order and dependency order. Load order establishes an ordering among symbol definitions, such that the first definition loaded (including definitions from the process image file and any dependent executable object files loaded with it) has priority over executable object files added later (by `dlopen()`). Load ordering is used in relocation processing. Dependency ordering uses a breadth-first order starting with a given executable object file, then all of its dependencies, then any dependents of those,

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

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

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

RETURN VALUE

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

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

Change Number: XSH/TC1/D5/0074 [74]

On Page: 735 Line: 24689 Section: `dlsym()`

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

NAME

`dlsym` -- get the address of a symbol from a symbol table handle

SYNOPSIS

```
#include <dlfcn.h>
void *dlsym(void *restrict handle, const char *restrict name);
```

DESCRIPTION

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

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

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

RETURN VALUE

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

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

ERRORS

No errors are defined.

EXAMPLES

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

```
void *handle;
int (*fptr)(int), *iptr, result;
/* open the needed symbol table */
handle = dlopen("/usr/home/me/libfoo.so", RTLD_LOCAL | RTLD_LAZY);
/* find the address of the function my_function */
fptr = (int (*)(int))dlsym(handle, "my_function");
/* find the address of the data object my_object */
iptr = (int *)dlsym(handle, "my_OBJ");
/* invoke my_function, passing the value of my_OBJ as the parameter */
result = (*fptr)(*iptr);
```

APPLICATION USAGE

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

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

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

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

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

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

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

RATIONALE

None.

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

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

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

In the APPLICATION USAGE section, change from:

None.

to:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls `dup2()` with an arbitrary integer for `fdes2` risks non-conforming behavior, and `dup2()` can only portably be used to overwrite file descriptor values that the application has obtained through explicit actions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race condition of leaking an unintended file descriptor into a child process, an application should consider opening all file descriptors with the `FD_CLOEXEC` bit set unless the file descriptor is intended to be inherited across `exec`.

Rationale: Austin Group Defect Report(s) applied: 149,428.
See <http://austingroupbugs.net/view.php?id=149>
See <http://austingroupbugs.net/view.php?id=428>

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

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

In the RATIONALE section, change from:

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

to:

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

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

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

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

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

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

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

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

Change Number: XSH/TC1/D5/0078 [283]

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

In the ERRORS section, delete:

The duplocale() function may fail if:

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

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

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

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

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

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

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

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

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

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

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

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

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

In the APPLICATION USAGE section, change from:

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

to:

None.

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

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

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

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

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

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

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

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

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/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, $2 * x/\sqrt{\pi}$
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 `erfc()`, `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:

extern char **environ;

Rationale: Austin Group Defect Report(s) applied: 386.
See <http://austingroupbugs.net/view.php?id=386>

Change Number: XSH/TC1/D5/0096 [167]

On Page: 772 Line: 25712 Section: exec

In the DESCRIPTION section, after L25712, add two new paragraphs:

Applications can change the entire environment in a single operation by assigning the environ variable to point to an array of character pointers to the new environment strings. After assigning a new value to environ, applications should not rely on the new environment strings remaining part of the environment, as a call to getenv(), [XSI]putenv(), [XSI]setenv(), unsetenv() or any function that is dependent on an environment variable may, on noticing that environ has changed, copy the environment strings to a new array and assign environ to point to it.

Any application that directly modifies the pointers to which the environ variable points has undefined behavior.

Rationale: Austin Group Defect Report(s) applied: 167.
See <http://austingroupbugs.net/view.php?id=167>

Change Number: XSH/TC1/D5/0097 [291]

On Page: 773 Line: 25737,25741 Section: exec

In the DESCRIPTION section, change both instances of "filename" to "filename string".

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

Change Number: XSH/TC1/D5/0098 [173]

On Page: 773 Line: 25757 Section: exec

In the DESCRIPTION section, change from:

If file descriptors 0, 1, and 2 would otherwise be closed after a successful call to one of the exec family of functions, and the new process image file has the set-user-ID or set-group-ID file mode bits set, and the ST_NOSUID bit is not set for the file system containing the new process image file, implementations may open an unspecified file for each of these file descriptors in the new process image.

to:

If file descriptor 0, 1, or 2 would otherwise be closed after a successful call to one of the exec family of functions, implementations may open an unspecified file for the file descriptor in the new process image. If a standard utility or a conforming application is executed with file descriptor 0 not open for reading or with file descriptor 1 or 2 not

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

Rationale: Austin Group Defect Report(s) applied: 173.
See <http://austingroupbugs.net/view.php?id=173>

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

On Page: 777 Line: 25899 Section: exec

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

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

to:

The ...

Rationale: Austin Group Defect Report(s) applied: 296.
See <http://austingroupbugs.net/view.php?id=296>

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

On Page: 777 Line: 25912 Section: exec

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

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 324.
See <http://austingroupbugs.net/view.php?id=324>
This is an editorial issue clarifying the intent of the standard.

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

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

In the ERRORS section, at L25917 before:

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

insert:

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

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

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

to:

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

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

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

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

Rationale: Austin Group Defect Report(s) applied: 167.
See <http://austingroupbugs.net/view.php?id=167>

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

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

In the `APPLICATION USAGE` section, change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 173.
See <http://austingroupbugs.net/view.php?id=173>

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

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

In the `RATIONALE` section, change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 291,429.

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

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

Change Number: XSH/TC1/D5/0106 [68]

On Page: 786 Line: 26265 Section: `exp()`

In the RETURN VALUE section, change from:

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

to:

If the correct value would cause underflow, `[MXX]` and is not representable `[MXX]`, a range error may occur, and `exp()`, `expf()`, and `expl()` shall return `[MXX]0.0`, or `[MXX]` (if IEC 60559 Floating-Point is not supported) an implementation-defined value no greater in magnitude than `DBL_MIN`, `FLT_MIN`, and `LDBL_MIN`, respectively.

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

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

Change Number: XSH/TC1/D5/0107 [68]

On Page: 786 Line: 26271 Section: `exp()`

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

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

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

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

Change Number: XSH/TC1/D5/0108 [68]

On Page: 788 Line: 26342 Section: `exp2()`

In the RETURN VALUE section, change from:

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

names an existing file that is neither a directory nor a symbolic link to a directory.

to:

[ENOTDIR] A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the path argument contains at least one non-`<slash>` character and ends with one or more trailing `<slash>` characters.

Rationale: Austin Group Defect Report(s) applied: 146,324.
See <http://austingroupbugs.net/view.php?id=146>
See <http://austingroupbugs.net/view.php?id=324>
The [ENOTDIR] error is updated, including an editorial clarification for the component of the path prefix.

Change Number: XSH/TC1/D5/0112 [291]

On Page: 796 Line: 26577-26579 Section: `fattach()`

In the EXAMPLES section, change both instances of "filename" to "pathname".

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

Change Number: XSH/TC1/D5/0113 [87]

On Page: 805 Line: 26801 Section: `fclose()`

In the DESCRIPTION section, change from:

... the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from or written to the stream being closed.

to:

... the file offset of the underlying open file description shall be set to the file position of the stream if the stream is the active handle to the underlying file description.

Rationale: Austin Group Defect Report(s) applied: 87.
See <http://austingroupbugs.net/view.php?id=87>

Change Number: XSH/TC1/D5/0114 [79]

On Page: 805 Line: 26824 Section: `fclose()`

In the ERRORS section [EIO], change from:

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

to:

... 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

Rationale: Austin Group Defect Report(s) applied: 401.
See <http://austingroupbugs.net/view.php?id=401>

Change Number: XSH/TC1/D5/0126 [87]

On Page: 844 Line: 28024 Section: `fflush()`

In the DESCRIPTION section, change from:

... the file offset of the underlying open file description shall be adjusted so that the next operation on the open file description deals with the byte after the last one read from or written to the stream being flushed.

to:

... the file offset of the underlying open file description shall be set to the file position of the stream, and any characters pushed back onto the stream by `ungetc()` or `ungetwc()` that have not subsequently been read from the stream shall be discarded (without further changing the file offset).

Rationale: Austin Group Defect Report(s) applied: 87.
See <http://austingroupbugs.net/view.php?id=87>

Change Number: XSH/TC1/D5/0127 [79]

On Page: 844 Line: 28042 Section: `fflush()`

In the ERRORS section [EIO], change from:

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

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

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

Change Number: XSH/TC1/D5/0128 [14]

On Page: 846 Line: 28101 Section: `fflush()`

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

Rationale: Austin Group Defect Report(s) applied: 14.
See <http://austingroupbugs.net/view.php?id=14>
This is an editorial improvement

Change Number: XSH/TC1/D5/0129 [79]

On Page: 848 Line: 28184 Section: `fgetc()`

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 ...
```

4708 *Rationale:* Austin Group Defect Report(s) applied: 182.
4709 See <http://austingroupbugs.net/view.php?id=182>

4712 **Change Number: XSH/TC1/D5/0135 [14]**

4715 On Page: 853 Line: 28316 Section: fgets()

4717 In the SEE ALSO section, add a reference to XSH Section 2.5.

4719 *Rationale:* Austin Group Defect Report(s) applied: 14.
4720 See <http://austingroupbugs.net/view.php?id=14>
4721 This is an editorial improvement

4724 **Change Number: XSH/TC1/D5/0136 [105]**

4727 On Page: 854 Line: 28343 Section: fgetwc()

4729 In the DESCRIPTION section, add a new paragraph at the end of the
4730 DESCRIPTION:

4732 The fgetwc() function shall not change the setting of errno if
4733 successful.

4735 *Rationale:* Austin Group Defect Report(s) applied: 105.
4736 See <http://austingroupbugs.net/view.php?id=105>

4739 **Change Number: XSH/TC1/D5/0137 [79]**

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.

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

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

Rationale: Austin Group Defect Report(s) applied: 68.
See <http://austingroupbugs.net/view.php?id=68>

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

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

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

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

to:

... and a NaN shall be returned.

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

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

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

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

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

Rationale: Austin Group Defect Report(s) applied: 68.
See <http://austingroupbugs.net/view.php?id=68>

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

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

In the NAME section, change from:

filename

to:

filename string

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

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

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

In the APPLICATION USAGE section, change from:

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 fildes 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 fildes 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>

5312 **Change Number: XSH/TC1/D5/0166** [451,291]

5315 On Page: 902 Line: 30140 Section: `fprintf()`

5317 In the EXAMPLES section, change from:

5319 Creating a Filename

5321 The following example creates a filename using information from a
5322 previous `getpwnam()` function that returned the HOME directory of the user.

```
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:
```

5336 Creating a Pathname

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.

```
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     snprintf(pathname, len, "%s/%jd.out", pw->pw_dir, (intmax_t)getpid());
5359     ...
5360 }
```

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>

5367 **Change Number: XSH/TC1/D5/0167** [14]

5370 On Page: 904 Line: 30229 Section: `fprintf()`

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

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

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

In the RATIONALE section, change from:

None

to:

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

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

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

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

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

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

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

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

This is an editorial improvement

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

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

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

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

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

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

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

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

In the ERRORS section [EIO], change from:

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

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: `fputwc()`
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
5563 #include <stdio.h>
5564 ...
5565 size_t bytes_read;
5566 char buf[100];
5567 FILE *fp;
5568 ...
5569 bytes_read = fread(buf, sizeof(buf), 1, fp);
5570 ...
5571
5572 to:
5573
5574 The following example reads a single element from the fp stream into
5575 the array pointed to by buf.
5576
5577 #include <stdio.h>
5578 ...
5579 size_t elements_read;
5580 char buf[100];
5581 FILE *fp;
5582 ...
5583 elements_read = fread(buf, sizeof(buf), 1, fp);
5584 ...
5585
5586 If a read error occurs, elements_read will be zero but the number of
5587 bytes read from the stream could be anything from zero to sizeof(buf)-1.
5588
5589 The following example reads multiple single-byte elements from the fp
5590 stream into the array pointed to by buf.
5591
5592 #include <stdio.h>
5593 ...
5594 size_t bytes_read;
5595 char buf[100];
5596 FILE *fp;
5597 ...
5598 bytes_read = fread(buf, 1, sizeof(buf), fp);
5599 ...
5600
5601 If a read error occurs, bytes_read will contain the number of bytes
5602 read from the stream.
5603
5604 *Rationale:* Austin Group Defect Report(s) applied: 232.
5605 See <http://austingroupbugs.net/view.php?id=232>
5606
5607
5608 **Change Number: XSH/TC1/D5/0179 [14]**
5609
5610
5611 On Page: 914 Line: 30558 Section: fread()
5612
5613 In the SEE ALSO section, add a reference to XSH Section 2.5.
5614
5615 *Rationale:* Austin Group Defect Report(s) applied: 14.
5616 See <http://austingroupbugs.net/view.php?id=14>
5617 This is an editorial improvement
5618
5619
5620 **Change Number: XSH/TC1/D5/0180 [283]**
5621
5622

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

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:

The `fseek()` [CX] and `fseeko()` [/CX] functions may fail if:

and then move the existing [ENXIO] condition at L31442-31443 out of the shall fail into this new may fail section, still with CX margin marker and shading.

[ENXIO] A request was made of a nonexistent device, or the request was outside the capabilities of the device.

Rationale: Austin Group Defect Report(s) applied: 225.
See <http://austingroupbugs.net/view.php?id=225>

Change Number: XSH/TC1/D5/0191 [14]

On Page: 938 Line: 31460 Section: `fseek()`

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

Rationale: Austin Group Defect Report(s) applied: 14.
See <http://austingroupbugs.net/view.php?id=14>

This is an editorial improvement

Change Number: XSH/TC1/D5/0192 [105]

On Page: 940 Line: 31503 Section: `fsetpos()`

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

The `fsetpos()` function shall not change the setting of `errno` if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See <http://austingroupbugs.net/view.php?id=105>

Change Number: XSH/TC1/D5/0193 [79]

On Page: 940 Line: 31523 Section: `fsetpos()`

In the ERRORS section [EIO], change from:

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

to:

... the calling thread is not blocking SIGTTOU, the process is not ignoring SIGTTOU, and the process group of the process is orphaned.

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

Change Number: XSH/TC1/D5/0194 [225]

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

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 [105]**
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

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

Change Number: XSH/TC1/D5/0213 [366]

On Page: 959 Line: 32119-32128 Section: ftok()

In the EXAMPLES section delete L32119-32128 [EXAMPLES] (the body of the first example, and the title of the second), leaving just the second example under the title "Getting an IPC Key".

Rationale: Austin Group Defect Report(s) applied: 366.
See <http://austingroupbugs.net/view.php?id=366>

Change Number: XSH/TC1/D5/0214 [366]

On Page: 959 Line: 32129 Section: ftok()

In the EXAMPLES section, on L32129 delete the word "unique".

Rationale: Austin Group Defect Report(s) applied: 366.
See <http://austingroupbugs.net/view.php?id=366>

Change Number: XSH/TC1/D5/0215 [366]

On Page: 959 Line: 32139 Section: ftok()

In the APPLICATION USAGE section add:

Applications should not assume that the resulting key value is unique.

Rationale: Austin Group Defect Report(s) applied: 366.
See <http://austingroupbugs.net/view.php?id=366>

Change Number: XSH/TC1/D5/0216 [366]

On Page: 959 Line: 32143 Section: ftok()

In the FUTURE DIRECTIONS section, change from:

None.

to:

Future versions of this standard may add new interfaces to provide unique keys.

Rationale: Austin Group Defect Report(s) applied: 366.
See <http://austingroupbugs.net/view.php?id=366>
No current implementation is known to provide unique keys.

Change Number: XSH/TC1/D5/0217 [403]

On Page: 964 Line: 32261 Section: ftw()

In the DESCRIPTION section, change from:

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

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

6409 to:
6410

6411 ... the current locale ...
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

On Page: 1008 Line: 33853 Section: getenv()

In the DESCRIPTION section, change from:

The string pointed to may be overwritten by a subsequent call to
getenv(), [CX]setenv(), unsetenv(), [/CX] [XSI] or putenv() [/XSI] but
shall not be overwritten by a call to any other function in this volume
of POSIX.1-200x.

to:

[CX]The returned string pointer might be invalidated or [/CX] the
string content might be overwritten by a subsequent call to getenv(),
[CX]setenv(), unsetenv(), [/CX] [XSI] or putenv() [/XSI] but they shall
not be affected by a call to any other function in this volume of
POSIX.1-200x.

Rationale: Austin Group Defect Report(s) applied: 75,428.

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

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

Change Number: XSH/TC1/D5/0239 [167]

On Page: 1008 Line: 33856 Section: getenv()

In the DESCRIPTION section, remove the text:

If the application modifies environ or the pointers to which it points,
the behavior of getenv() is undefined.

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

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

The text relating to undefined behavior on modification of
environ is removed as this is now covered by general text added in XBD
Section 8.1.

Change Number: XSH/TC1/D5/0240 [167]

On Page: 1008 Line: 33880 Section: getenv()

In the RATIONALE section, change from:

Conforming applications are required not to modify environ directly,
but to use only the functions described here to manipulate the process
environment as an abstract object. Thus, the implementation of the
environment access functions has complete control over the data structure
used to represent the environment (subject to the requirement that environ
be maintained as a list of strings with embedded <equals-sign> characters
for applications that wish to scan the environment). This constraint
allows the implementation to properly manage the memory it allocates,
either by using allocated storage for all variables (copying them on
the first invocation of setenv() or unsetenv()), or keeping track of
which strings are currently in allocated space and which are not, via a
separate table or some other means. This enables the implementation to
free any allocated space used by strings (and perhaps the pointers to
them) stored in environ when unsetenv() is called. A C runtime start-up
procedure (that which invokes main() and perhaps initializes environ)
can also initialize a flag indicating that none of the environment has
yet been copied to allocated storage, or that the separate table has

not yet been initialized.

In fact, for higher performance of `getenv()`, the implementation could also maintain a separate copy of the environment in a data structure that could be searched much more quickly (such as an indexed hash table, or a binary tree), and update both it and the linear list at `environ` when `setenv()` or `unsetenv()` is invoked.

to:

Conforming applications are required not to directly modify the pointers to which `environ` points, but to use only the `setenv()`, `unsetenv()`, and `putenv()` functions, or assignment to `environ` itself, to manipulate the process environment. This constraint allows the implementation to properly manage the memory it allocates. This enables the implementation to free any space it has allocated to strings (and perhaps the pointers to them) stored in `environ` when `unsetenv()` is called. A C runtime start-up procedure (that which invokes `main()` and perhaps initializes `environ`) can also initialize a flag indicating that none of the environment has yet been copied to allocated storage, or that the separate table has not yet been initialized. If the application switches to a complete new environment by assigning a new value to `environ`, this can be detected by `getenv()`, `setenv()`, `unsetenv()`, or `putenv()` and the implementation can at that point reinitialize based on the new environment. (This may include copying the environment strings into a new array and assigning `environ` to point to it.)

In fact, for higher performance of `getenv()`, implementations that do not provide `putenv()` could also maintain a separate copy of the environment in a data structure that could be searched much more quickly (such as an indexed hash table, or a binary tree), and update both it and the linear list at `environ` when `setenv()` or `unsetenv()` is invoked. On implementations that do provide `putenv()`, such a copy might still be worthwhile but would need to allow for the fact that applications can directly modify the content of environment strings added with `putenv()`. For example, if an environment string found by searching the copy is one that was added using `putenv()`, the implementation would need to check that the string in `environ` still has the same name (and value, if the copy includes values), and whenever searching the copy produces no match the implementation would then need to search each environment string in `environ` that was added using `putenv()` in case any of them have changed their names and now match. Thus each use of `putenv()` to add to the environment would reduce the speed advantage of having the copy.

Rationale: Austin Group Defect Report(s) applied: 167.
See <http://austingroupbugs.net/view.php?id=167>

Change Number: XSH/TC1/D5/0241 [75]

On Page: 1014 Line: 34021 Section: `getgrgid()`

In the RETURN VALUE section, change from:

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

to:

The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure,

might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to `getgrent()`, `getgrgid()`, or `getgrnam()`.

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

Change Number: XSH/TC1/D5/0242 [75]

On Page: 1018 Line: 34149 Section: `getgrnam()`

In the RETURN VALUE section, change from:

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

to:

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

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

Change Number: XSH/TC1/D5/0243 [172]

On Page: 1029 Line: 34464 Section: `getlogin()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

The `getlogin()` and `getlogin_r()` functions may make use of file descriptors 0, 1, and 2 to find the controlling terminal of the current process, examining each in turn until the terminal is found. If in this case none of these three file descriptors is open to the controlling terminal, these functions may fail. The method used to find the terminal associated with a file descriptor may depend on the file descriptor being open to the actual terminal device, not `/dev/tty`.

Rationale: Austin Group Defect Report(s) applied: 172.
See <http://austingroupbugs.net/view.php?id=172>

Change Number: XSH/TC1/D5/0244 [75]

On Page: 1029 Line: 34469 Section: `getlogin()`

In the RETURN VALUE section, change from:

The return value from `getlogin()` may point to static data whose content is overwritten by each call.

to:

The application shall not modify the string returned. The returned pointer might be invalidated or the string content might be overwritten 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 After:
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

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.

See <http://austingroupbugs.net/view.php?id=460>

Change Number: XSH/TC1/D5/0254 [464]

On Page: 1045 Line: 35015 Section: `getpeername()`

In the DESCRIPTION section, insert the following sentence at the beginning of the paragraph:

The `address_len` argument points to a `socklen_t` object which on input specifies the length of the supplied `sockaddr` structure, and on output specifies the length of the stored address.

Rationale: Austin Group Defect Report(s) applied: 464.

See <http://austingroupbugs.net/view.php?id=464>

Change Number: XSH/TC1/D5/0255 [75,428]

On Page: 1057 Line: 35319 Section: `getpwnam()`

In the RETURN VALUE section, change from:

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

to:

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

Rationale: Austin Group Defect Report(s) applied: 75,428.

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

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

Change Number: XSH/TC1/D5/0256 [75]

On Page: 1061 Line: 35452 Section: `getpwuid()`

In the RETURN VALUE section, change from:

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

to:

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

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

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

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

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, rsize=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

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 These changes were overlooked during the revision when
7632 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()`

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

Change Number: XSH/TC1/D5/0307 [283]

On Page: 1167 Line: 38837 Section: `isupper()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to `isupper_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/D5/0308 [283]

On Page: 1167 Line: 38842 Section: `isupper()`

In the ERRORS section, change from:

The `isupper_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/D5/0309 [302]

On Page: 1169 Line: 38876 Section: `iswalnum()`

In the DESCRIPTION section, change from:

... current locale of the process ...

to:

... 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/D5/0310 [283]

On Page: 1169 Line: 38880 Section: `iswalnum()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to `iswalnum_l()` is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale 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 locale

8006 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

8017 In the ERRORS section, change from:
8018
8019 The iswalphal() 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

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 **Change Number: XSH/TC1/D5/0333** [283]
8349
8350
8351 On Page: 1184 Line: 39295 Section: `iswprint()`
8352
8353 In the DESCRIPTION section, add a new paragraph to the end of the section:
8354
8355 [CX]The behavior is undefined if the locale argument to `iswprint_l()`
8356 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
8357 object handle.[/CX]
8358
8359 *Rationale:* Austin Group Defect Report(s) applied: 283.
8360 See <http://austingroupbugs.net/view.php?id=283>
8361
8362 **Change Number: XSH/TC1/D5/0334** [283]
8363
8364 On Page: 1184 Line: 39300 Section: `iswprint()`
8365
8366 In the ERRORS section, change from:
8367
8368 The `iswprint_l()` function may fail if:
8369
8370 [EINVAL] locale is not a valid locale object handle.
8371
8372 to:
8373
8374 No errors are defined.
8375
8376 *Rationale:* Austin Group Defect Report(s) applied: 283.
8377 See <http://austingroupbugs.net/view.php?id=283>
8378
8379 **Change Number: XSH/TC1/D5/0335** [302]
8380
8381 On Page: 1186 Line: 39339 Section: `iswpunct()`
8382
8383 In the DESCRIPTION section, change from:
8384
8385 ... current locale of the process ...
8386
8387 to:
8388
8389 ... current locale ...
8390
8391
8392
8393
8394

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>

Change Number: XSH/TC1/D5/0364 [68]

On Page: 1249 Line: 41185 Section: `loglp()`

In the RETURN VALUE section, change from:

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

to:

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

[MX]If `x` is not returned, `loglp()`, `loglpf()`, and `loglpl()` shall return an implementation-defined value no greater in magnitude than `DBL_MIN`, `FLT_MIN`, and `LDBL_MIN`, respectively. [/MX]

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

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

Change Number: XSH/TC1/D5/0365 [394]

On Page: 1256 Line: 41363 Section: `longjmp()`

In the DESCRIPTION section, change from:

The `longjmp()` function shall restore the environment saved by the most recent invocation of `setjmp()` in the same thread, with the corresponding `jmp_buf` argument. If there is no such invocation, or if the function containing the invocation of `setjmp()` has terminated execution in the interim, or if the invocation of `setjmp()` was within the scope of an identifier with variably modified type and execution has left that scope in the interim, the behavior is undefined

to:

The `longjmp()` function shall restore the environment saved by the most recent invocation of `setjmp()` in the same process, with the corresponding `jmp_buf` argument. If the most recent invocation of `setjmp()` with the corresponding `jmp_buf` occurred in another thread, or if there is no such invocation, or if the function containing the invocation of `setjmp()` has terminated execution in the interim, or if the invocation of `setjmp()` was within the scope of an identifier with variably modified type and execution has left that scope in the interim, the behavior is undefined

Rationale: Austin Group Defect Report(s) applied: 394.

See <http://austingroupbugs.net/view.php?id=394>

Change Number: XSH/TC1/D5/0366 [421]

On Page: 1265 Line: 41639 Section: `lseek()`

In the RETURN VALUE section, change from:

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

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

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 ends
9399 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.

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** [104]
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

Change Number: XSH/TC1/D5/0394 [259]

On Page: 1328 Line: 43672 Section: `mq_open()`

In the DESCRIPTION section (`O_CREAT`), after:

... the message queue `mq_maxmsg` and `mq_msgsize` attributes shall be set to the values of the corresponding members in the `mq_attr` structure referred to by `attr`.

Add:

The values of the `mq_flags` and `mq_curmsgs` members of the `mq_attr` structure shall be ignored.

Rationale: Austin Group Defect Report(s) applied: 259.
See <http://austingroupbugs.net/view.php?id=259>

Change Number: XSH/TC1/D5/0395 [345]

On Page: 1342 Line: 44077 Section: `msgctl()`

In the DESCRIPTION section (for `IPC_SET`), add:

Also, the `msg_ctime` timestamp shall be set to the current time, as described in XSH Section 2.7.1.

Rationale: Austin Group Defect Report(s) applied: 345.
See <http://austingroupbugs.net/view.php?id=345>

Change Number: XSH/TC1/D5/0396 [345]

On Page: 1344 Line: 44145 Section: `msgget()`

In the DESCRIPTION section, change from:

... set equal to the current time

to:

... set to the current time, as described in XSH Section 2.7.1

Rationale: Austin Group Defect Report(s) applied: 345.
See <http://austingroupbugs.net/view.php?id=345>

Change Number: XSH/TC1/D5/0397 [344]

On Page: 1345 Line: 44174 Section: `msgget()`

In the SEE ALSO section, add `ftok()` prior to `mq_close()`.

Rationale: Austin Group Defect Report(s) applied: 344.
See <http://austingroupbugs.net/view.php?id=344>

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 nexttoward() 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" :

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 ... the locale of the process ...
9801

9802
9803 to:

9804 ... the current locale ...
9805

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

Change Number: XSH/TC1/D5/0415 [75,402]

On Page: 1375 Line: 45109 Section: `nl_langinfo()`

In the RETURN VALUE section, change from:

This pointer may point to static data that may be overwritten on the next call to either function.

to:

The application shall not modify the string returned. The pointer returned by `nl_langinfo()` might be invalidated or the string content might be overwritten by a subsequent call to `nl_langinfo()` in any thread or to `nl_langinfo_l()` in the same thread or the initial thread, by subsequent calls to `setlocale()` with a category corresponding to the category of item (see `<langinfo.h>`) or the category `LC_ALL`, or by subsequent calls to `uselocale()` which change the category corresponding to the category of item. The pointer returned by `nl_langinfo_l()` might be invalidated or the string content might be overwritten by a subsequent call to `nl_langinfo_l()` in the same thread or to `nl_langinfo()` in any thread, or by subsequent calls to `freelocale()` or `newlocale()` which free or modify the locale object that was passed to `nl_langinfo_l()`.

Rationale: Austin Group Defect Report(s) applied: 75,402.

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

See <http://austingroupbugs.net/view.php?id=402>

Change Number: XSH/TC1/D5/0416 [283]

On Page: 1375 Line: 45111 Section: `nl_langinfo()`

In the ERRORS section, change from:

The `nl_langinfo_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/D5/0417 [402]

On Page: 1376 Line: 45126 Section: `nl_langinfo()`

In the RATIONALE section, change from:

None.

to:

The possible interactions between internal data used by `nl_langinfo()` and `nl_langinfo_l()` are complicated by the fact that `nl_langinfo_l()`

must be thread-safe but `nl_langinfo()` need not be. The various implementation choices are:

1. `nl_langinfo_l()` and `nl_langinfo()` use separate buffers, or at least one of them does not use an internal string buffer. In this case there are no interactions.
2. `nl_langinfo_l()` and `nl_langinfo()` share an internal per-thread buffer. There can be interactions, but only in the same thread.
3. `nl_langinfo_l()` uses an internal per-thread buffer, and `nl_langinfo()` uses (in all threads) the same buffer that `nl_langinfo_l()` uses in the initial thread. There can be interactions, but only when `nl_langinfo_l()` is called in the initial thread.

Rationale: Austin Group Defect Report(s) applied: 402.
See <http://austingroupbugs.net/view.php?id=402>

Change Number: XSH/TC1/D5/0418 [292]

On Page: 1380 Line: 45209 Section: `open()`

In the DESCRIPTION section, change from:

`O_DIRECTORY` If path does not name a directory, fail and set `errno` to `[ENOTDIR]`.

to:

`O_DIRECTORY` If path resolves to a non-directory file, fail and set `errno` to `[ENOTDIR]`.

Rationale: Austin Group Defect Report(s) applied: 292.
See <http://austingroupbugs.net/view.php?id=292>

The standard is clear regarding symbolic links; however, there is an overlap between the `[ENOENT]` and `[ENOTDIR]` error conditions. The intention is that an `[ENOENT]` error should result when the pathname does not resolve.

Change Number: XSH/TC1/D5/0419 [141]

On Page: 1380 Line: 45239 Section: `open()`

In the DESCRIPTION section (for `O_NONBLOCK`), change from:

Otherwise, the behavior of `O_NONBLOCK` is unspecified.

to:

Otherwise, the `O_NONBLOCK` flag shall not cause an error, but it is unspecified whether the file status flags will include the `O_NONBLOCK` flag.

Rationale: Austin Group Defect Report(s) applied: 141.
See <http://austingroupbugs.net/view.php?id=141>

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 -1 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 ferror(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

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 `fildest` 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 `fildest` 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 `fildest` 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

child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls `posix_spawn_file_actions_addclose()` with an arbitrary integer risks non-conforming behavior, and this function can only portably be used to close file descriptor values that the application has obtained through explicit actions, or for the three file descriptors corresponding to the standard file streams. In order to avoid a race condition of leaking an unintended file descriptor into a child process, an application should consider opening all file descriptors with the `FD_CLOEXEC` bit set unless the file descriptor is intended to be inherited across `exec`.

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

Change Number: XSH/TC1/D5/0438 [291]

On Page: 1431 Line: 46915 Section: `posix_spawn_file_actions_addclose()`

In the RATIONALE section, change "filenames" to "pathnames".

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

Change Number: XSH/TC1/D5/0439 [418]

On Page: 1432 Line: 46954 Section: `posix_spawn_file_actions_addclose()`

In the RATIONALE section, add a new paragraph at the end of the section:

The `posix_spawn_file_actions_addclose()` function is not required to check whether the file descriptor is less than `{OPEN_MAX}` because on some implementations `{OPEN_MAX}` reflects the `RLIMIT_NOFILE` soft limit and therefore calling `setrlimit()` to reduce this limit can result in an `{OPEN_MAX}` value less than or equal to an already open file descriptor. Applications need to be able to close such file descriptors on `spawn`. On implementations where `{OPEN_MAX}` does not change, it is recommended that `posix_spawn_file_actions_addclose()` should return `EBADF` if `fd` is greater than or equal to `{OPEN_MAX}`.

Rationale: Austin Group Defect Report(s) applied: 418.
See <http://austingroupbugs.net/view.php?id=418>

Change Number: XSH/TC1/D5/0440 [149]

On Page: 1433 Line: 46997 Section: `posix_spawn_file_actions_adddup2()`

In the APPLICATION USAGE section, add a new paragraph at the end of the section:

Implementations may use file descriptors that must be inherited into child processes for the child process to remain conforming, such as for message catalog or tracing purposes. Therefore, an application that calls `posix_spawn_file_actions_adddup2()` with an arbitrary integer for `newfd` risks non-conforming behavior, and this function can only portably be used to overwrite file descriptor values that the application has obtained through explicit actions, or for the three file descriptors 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

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: psiginfo()

10461

10462 In the DESCRIPTION section, add three new paragraphs to the end of
10463 the section:

10464

10465 The psiginfo() and psignal() functions shall not change the setting of
10466 errno if successful.

10467

10468 On error, the psiginfo() 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 psiginfo() 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

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

not be provided. Attempting to relock the mutex causes deadlock. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, undefined behavior results.

If the mutex type is `PTHREAD_MUTEX_ERRORCHECK`, then error checking shall be provided. If a thread attempts to relock a mutex that it has already locked, an error shall be returned. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, an error shall be returned.

If the mutex type is `PTHREAD_MUTEX_RECURSIVE`, then the mutex shall maintain the concept of a lock count. When a thread successfully acquires a mutex for the first time, the lock count shall be set to one. Every time a thread relocks this mutex, the lock count shall be incremented by one. Each time the thread unlocks the mutex, the lock count shall be decremented by one. When the lock count reaches zero, the mutex shall become available for other threads to acquire. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, an error shall be returned.

If the mutex type is `PTHREAD_MUTEX_DEFAULT`, attempting to recursively lock the mutex results in undefined behavior. Attempting to unlock the mutex if it was not locked by the calling thread results in undefined behavior. Attempting to unlock the mutex if it is not locked results in undefined behavior.

to:

The mutex object referenced by `mutex` shall be locked by a call to `pthread_mutex_lock()` that returns zero or `EOWNERDEAD`. If the mutex is already locked by another thread, the calling thread shall block until the mutex becomes available. This operation shall return with the mutex object referenced by `mutex` in the locked state with the calling thread as its owner. If a thread attempts to relock a mutex that it has already locked, `pthread_mutex_lock()` shall behave as described in the Relock column of the following table. If a thread attempts to unlock a mutex that it has not locked or a mutex which is unlocked, `pthread_mutex_unlock()` shall behave as described in the Unlock When Not Owner column of the following table.

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

1. If the mutex type is `PTHREAD_MUTEX_DEFAULT`, the behavior of `pthread_mutex_lock()` may correspond to one of the three other standard mutex types as described in the table above. If it does not correspond 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

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

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 **Change Number: XSH/TC1/D5/0477 [14]**
11184
11185
11186 On Page: 1724 Line: 55083 Section: puts()
11187
11188 In the SEE ALSO section, add a reference to XSH Section 2.5.
11189
11190 *Rationale:* Austin Group Defect Report(s) applied: 14.
11191 See <http://austingroupbugs.net/view.php?id=14>
11192 This is an editorial improvement
11193
11194
11195 **Change Number: XSH/TC1/D5/0478 [14]**
11196
11197
11198 On Page: 1726 Line: 55126 Section: putwc()
11199
11200 In the SEE ALSO section, add a reference to XSH Section 2.5.
11201
11202 *Rationale:* Austin Group Defect Report(s) applied: 14.
11203 See <http://austingroupbugs.net/view.php?id=14>
11204 This is an editorial improvement
11205
11206
11207 **Change Number: XSH/TC1/D5/0479 [14]**
11208
11209
11210 On Page: 1727 Line: 55157 Section: putwchar()
11211
11212 In the SEE ALSO section, add a reference to XSH Section 2.5.
11213
11214 *Rationale:* Austin Group Defect Report(s) applied: 14.
11215 See <http://austingroupbugs.net/view.php?id=14>
11216 This is an editorial improvement
11217
11218
11219 **Change Number: XSH/TC1/D5/0480 [218]**
11220
11221
11222 On Page: 1739 Line: 55487 Section: read()
11223
11224 In the ERRORS section, change from:
11225
11226 [EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the
11227 thread would be delayed.
11228
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 [EWOULDBLOCK]
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

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** [143]
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

Change Number: XSH/TC1/D5/0494 [151,231]

On Page: 1750 Line: 55912 Section: readlink()

In the RATIONALE section, delete the first sentence:

Since POSIX.1-2008 does not require any association of file times with symbolic links, there is no requirement that file times be updated by readlink().

Rationale: Austin Group Defect Report(s) applied: 151,231.

See <http://austingroupbugs.net/view.php?id=151>

See <http://austingroupbugs.net/view.php?id=231>

Change Number: XSH/TC1/D5/0495 [400]

On Page: 1754 Line: 56030 Section: realloc()

In the DESCRIPTION section, change from:

The realloc() function shall change the size of the memory object pointed to by ptr to the size specified by size. The contents of the object shall remain unchanged up to the lesser of the new and old sizes. If the new size of the memory object would require movement of the object, the space for the previous instantiation of the object is freed. If the new size is larger, the contents of the newly allocated portion of the object are unspecified. If size is 0 and ptr is not a null pointer, the object pointed to is freed.

to:

The realloc() function shall deallocate the old object pointed to by ptr and return a pointer to a new object that has the size specified by size. The contents of the new object shall be the same as that of the old object prior to deallocation, up to the lesser of the new and old sizes. Any bytes in the new object beyond the size of the old object have indeterminate values. If the size of the space requested is zero, the behavior shall be implementation-defined: either a null pointer is returned, or the behavior shall be as if the size were some nonzero value, except that the returned pointer shall not be used to access an object.

Rationale: Austin Group Defect Report(s) applied: 400.

See <http://austingroupbugs.net/view.php?id=400>

Change Number: XSH/TC1/D5/0496 [400]

On Page: 1754 Line: 56046 Section: realloc()

In the RETURN VALUE section, change from:

Upon successful completion with a size not equal to 0, realloc() shall return a pointer to the (possibly moved) allocated space. If size is 0, either a null pointer or a unique pointer that can be successfully passed to free() shall be returned. If there is not enough available memory, realloc() shall return a null pointer [CX] and set errno to [ENOMEM].[/CX]

to:

Upon successful completion, `realloc()` shall return a pointer to the (possibly moved) allocated space. If size is 0, either:

- * A null pointer shall be returned [CX] and `errno` set to an implementation defined value [CX].
- * A unique pointer that can be successfully passed to `free()` shall be returned, and the memory object pointed to by `ptr` shall be freed. The application shall ensure that the pointer is not used to access an object.

If there is not enough available memory, `realloc()` shall return a null pointer [CX] and set `errno` to `[ENOMEM]` [CX]. If `realloc()` returns a null pointer [CX] and `errno` has been set to a `ENOMEM` [CX], the memory referenced by `ptr` shall not be changed.

Rationale: Austin Group Defect Report(s) applied: 400.
See <http://austingroupbugs.net/view.php?id=400>

Change Number: XSH/TC1/D5/0497 [400]

On Page: 1754 Line: 56056 Section: `realloc()`

In the APPLICATION USAGE section, change from:

None.

to

The description of `realloc()` has been modified from previous versions of this standard to align with C99. Previous versions explicitly permitted a call to `realloc(p, 0)` to free the space pointed to by `p` and return `NULL`. While this behavior could be interpreted as permitted by this version of the standard, the C language committee have indicated that this interpretation is incorrect. Applications should assume that if `realloc` returns a null pointer, the space pointed to by `p` has not been freed. Since this could lead to double-frees, implementations should also set `errno` if a null pointer actually indicates a failure, and applications should only free the space if `errno` was changed.

Rationale: Austin Group Defect Report(s) applied: 400.
See <http://austingroupbugs.net/view.php?id=400>

Change Number: XSH/TC1/D5/0498 [400]

On Page: 1754 Line: 56060 Section: `realloc()`

In the FUTURE DIRECTIONS section, change from:

None.

to:

This standard defers to the C standard. While that standard currently has language that might permit `realloc(p, 0)`, where `p` is not a null pointer, to free `p` while still returning a null pointer, the committee responsible for that standard is considering clarifying the language to explicitly prohibit that alternative.

Rationale: Austin Group Defect Report(s) applied: 400.
See <http://austingroupbugs.net/view.php?id=400>

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

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 (`math_errhandling & 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

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`


```
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()

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:

```
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
12459 On Page: 1842 Line: 58864 Section: semop()
12460
12461 In the EXAMPLES section, delete "char filename[PATH_MAX+1]".
12462
12463 Rationale: Austin Group Defect Report(s) applied: 291,429.
12464 See http://austingroupbugs.net/view.php?id=291
12465 See http://austingroupbugs.net/view.php?id=429
12466 This declaration is unused in this example.
12467
12468 Change Number: XSH/TC1/D5/0543 [463]
12469
12470
12471
12472 On Page: 1845 Line: 58990 Section: send()
12473
12474 In the APPLICATION USAGE section, change from:
12475
12476 The send() function is equivalent to sendto() with a null pointer dest_len
12477 argument, and to write() if no flags are used.
12478
12479 to:
12480
12481 If the socket argument refers to a connection-mode socket, the send()
12482 function is equivalent to sendto() (with any value for the dest_addr and
12483 dest_len arguments, as they are ignored in this case). If the socket
12484 argument refers to a socket and the flags argument is 0, the send()
12485 function is equivalent to write().
12486
12487 Rationale: Austin Group Defect Report(s) applied: 463.
12488 See http://austingroupbugs.net/view.php?id=463
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 This is an editorial improvement

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 The text relating to undefined behavior on modification

12596 of environ is removed as this is now covered by general text added in
12597 XBD 8.1.

12598

12599

12600 **Change Number: XSH/TC1/D5/0550** [185]

12601

12602

12603 On Page: 1857 Line: 59356 Section: setenv()

12604

12605 In the ERRORS section, change from:

12606

12607 [EINVAL] The name argument is a null pointer, points to an empty string,
12608 or points to a string containing an '=' character.

12609

12610 to:

12611

12612 [EINVAL] The envname argument points to an empty string or points to a
12613 string containing an '=' character.

12614

12615 *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

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
12769 ... the global locale ...
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: shmctl()
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: shmdt()
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

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

Change Number: XSH/TC1/D5/0579 [140]

On Page: 1919 Line: 61064 Section: sigaction()

In the APPLICATION USAGE section, add a new a paragraph at the end of the section:

See also the rationale for Realtime Signal Generation and Delivery in Section B.2.4.2.

Rationale: Austin Group Defect Report(s) applied: 140.

See <http://austingroupbugs.net/view.php?id=140>

Change Number: XSH/TC1/D5/0580 [275]

On Page: 1937 Line: 61633 Section: signal()

In the DESCRIPTION section, change from:

If the signal occurs as the result of calling the abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue() [/CX] function, the signal handler shall not call the raise() function.

to:

The C Standard places a restriction on applications relating to the use of raise() from signal handlers. [CX]This restriction does not apply to POSIX applications, as POSIX.1-2008 requires raise() to be async-signal-safe (see Section 2.4.3 Signal Actions).[/CX]

Rationale: Austin Group Defect Report(s) applied: 275.

See <http://austingroupbugs.net/view.php?id=275>

Change Number: XSH/TC1/D5/0581 [66]

On Page: 1937 Line: 61635 Section: signal()

In the DESCRIPTION section, change from:

If the signal occurs other than as the result of calling abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue() [/CX], the behavior is undefined if the signal handler refers to any object with static storage duration other than by assigning a value to an object declared as volatile sig_atomic_t, or if the signal handler calls any function in the standard library other than one of the functions listed in Section 2.4. Furthermore, if such a call fails, the value of errno is unspecified.

to:

[CX]If the process is multi-threaded[/CX], or if the process is single threaded and a signal handler is executed other than as the result of:

* The process calling abort(), raise(), [CX]kill(), pthread_kill(), or sigqueue() [/CX] to generate a signal that is not blocked

* [CX]A pending signal being unblocked and being delivered before the call that unblocked it returns[/CX]

the behavior is undefined if the signal handler refers to any object [CX]other than `errno`[CX] with static storage duration other than by assigning a value to an object declared as `volatile sig_atomic_t`, or if the signal handler calls any function defined in this standard other than [CX]one of the functions listed in Section 2.4[CX].

Rationale: Austin Group Defect Report(s) applied: 66.
See <http://austingroupbugs.net/view.php?id=66>

Change Number: XSH/TC1/D5/0582 [105]

On Page: 1937 Line: 61644 Section: `signal()`

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

The `signal()` function shall not change the setting of `errno` if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See <http://austingroupbugs.net/view.php?id=105>

Change Number: XSH/TC1/D5/0583 [392]

On Page: 1952 Line: 62027 Section: `sigtimedwait()`

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

Note that in order to ensure generated signals are queued and signal values passed to `sigqueue()` are available in `si_value`, applications which use `sigwaitinfo()` or `sigtimedwait()` need to set the `SA_SIGINFO` flag for each signal in the set (see 2.4). This means setting each signal to be handled by a three-argument signal catching function, even if the handler will never be called. It is not possible (portably) to set a signal handler to `SIG_DFL` while setting the `SA_SIGINFO` flag, because assigning to the `sa_handler` member of `struct sigaction` instead of the `sa_sigaction` member would result in undefined behavior, and `SIG_DFL` need not be assignment compatible with `sa_sigaction`. Even if an assignment of `SIG_DFL` to `sa_sigaction` is accepted by the compiler, the implementation need not treat this value as special - it could just be taken as the address of a signal catching function.

Rationale: Austin Group Defect Report(s) applied: 392.
See <http://austingroupbugs.net/view.php?id=392>

Change Number: XSH/TC1/D5/0584 [76]

On Page: 1955 Line: 62165 Section: `sigwait()`

In the RATIONALE section, change from:

After some consideration, threads were allowed to use semaphores and `sem_post()` was defined to be `async-signal` and `async-cancel-safe`.

to:

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

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
13337 On Page: 1985 Line: 62813 Section: strcasecmp()
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
13356 On Page: 1985 Line: 62817 Section: strcasecmp()
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

Change Number: XSH/TC1/D5/0591 [283]

On Page: 1985 Line: 62819 Section: `strcasecmp()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

The behavior is undefined if the locale argument to `strcasecmp_l()` or `strncasecmp_l()` is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale object handle.

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

Change Number: XSH/TC1/D5/0592 [283]

On Page: 1985 Line: 62829 Section: `strcasecmp()`

In the ERRORS section, change from:

The `strcasecmp_l()` and `strncasecmp_l()` functions 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/D5/0593 [283]

On Page: 1991 Line: 62993 Section: `strcoll()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to `strcoll_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/D5/0594 [283]

On Page: 1991 Line: 63007 Section: `strcoll()`

In the ERRORS section, delete:

The `strcoll_l()` function may fail if:

[EINVAL] Locale is not a valid locale object handle.

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

Change Number: XSH/TC1/D5/0595 [75]

On Page: 1999 Line: 63237 Section: `strerror()`

In the DESCRIPTION section, change from:

The string pointed to shall not be modified by the application.
The string may be overwritten by a subsequent call to `strerror()`.
[CX]The string may be overwritten by a subsequent call to `strerror_l()`
in the same thread. [/CX]

to:

The application shall not modify the string returned. [CX]The returned
string pointer might be invalidated or [/CX] the string content might be
overwritten by a subsequent call to `strerror()`, [CX]or by a subsequent
call to `strerror_l()` in the same thread [/CX].

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

Change Number: XSH/TC1/D5/0596 [447]

On Page: 1999 Line: 63244 Section: `strerror()`

In the DESCRIPTION section, change from:

The `strerror()` and `strerror_l()` functions shall not change the setting
of `errno` if successful.

Since no return value is reserved to indicate an error, an application
wishing to check for error situations should set `errno` to 0, then call
`strerror()`, then check `errno`.

to:

The `strerror()` and `strerror_l()` functions shall not change the setting
of `errno` if successful.

Since no return value is reserved to indicate an error of `strerror()`, an
application wishing to check for error situations should set `errno` to 0,
then call `strerror()`, then check `errno`. Similarly, since `strerror_l()`
is required to return a string for some errors, an application wishing
to check for all error situations should set `errno` to 0, then call
`strerror_l()`, then check `errno`.

Rationale: Austin Group Defect Report(s) applied: 447.
See <http://austingroupbugs.net/view.php?id=447>

Change Number: XSH/TC1/D5/0597 [382,428]

On Page: 1999 Line: 63253 Section: `strerror()`

In the DESCRIPTION section, change from (retaining the CX shading):

If the value of `errnum` is a valid error number, the message string shall
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
13517 **Change Number: XSH/TC1/D5/0598** [283]
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:

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 errno 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"

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%y1+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(), strtodf(), 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: strtointmax()

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: strtointmax()

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: strtimax()
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

13957 **Change Number: XSH/TC1/D5/0618** [453]

13958

13959

13960 On Page: 2044 Line: 64785 Section: strtol()

13961

13962 In the RETURN VALUE section, change from:

13963

13964 If no conversion could be performed, 0 shall be returned [CX]and
13965 errno may be set to [EINVAL][CX].

13966

13967 to:

13968

13969 If no conversion could be performed, 0 shall be returned [CX]and
13970 errno may be set to [EINVAL][CX].

13971

13972 [CX]If the value of base is not supported, 0 shall be returned and
13973 errno shall be set to [EINVAL].[/CX]

13974

13975 *Rationale:* Austin Group Defect Report(s) applied: 453.

13976 See <http://austingroupbugs.net/view.php?id=453>

13977

13978

13979 **Change Number: XSH/TC1/D5/0619** [453]

13980

13981

13982 On Page: 2044 Line: 64791 Section: strtol()

13983

13984 In the ERRORS section, change from:

13985

13986 These functions shall fail if:

13987

13988 [ERANGE] The value to be returned is not representable.

13989

13990 These functions may fail if:

13991

13992 [EINVAL] [CX] The value of base is not supported. [/CX]

13993

13994 to:

13995

13996 These functions shall fail if:

13997

13998 [EINVAL] [CX] The value of base is not supported. [/CX]

13999

14000 [ERANGE] The value to be returned is not representable.

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].

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

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

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

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

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

Change Number: XSH/TC1/D5/0663 [106]

On Page: 2107 Line: 66673 Section: time()

In the ERRORS section, change from:

No errors are defined.

to:

The time() function may fail if:

[EOVERFLOW] The number of seconds since the Epoch will not fit in an object of type time_t.

with CX shading.

Rationale: Austin Group Defect Report(s) applied: 106.

See <http://austingroupbugs.net/view.php?id=106>

Change Number: XSH/TC1/D5/0664 [350]

On Page: 2108 Line: 66712 Section: time()

In the RATIONALE section, change from:

The time() function returns a value in seconds (type time_t) while times() returns a set of values in clock ticks (type clock_t). Some historical implementations, such as 4.3 BSD, have mechanisms capable of returning more precise times (see below). A generalized timing scheme to unify these various timing mechanisms has been proposed but not adopted.

to:

The time() function returns a value in seconds while clock_gettime() and gettimeofday() return a struct timespec (seconds and nanoseconds) and struct timeval (seconds and microseconds), respectively, and are therefore capable of returning more precise times. The times() function is also capable of more precision than time() as it returns a value in clock ticks, although it returns the elapsed time since an arbitrary point such as system boot time, not since the epoch.

Rationale: Austin Group Defect Report(s) applied: 350.

See <http://austingroupbugs.net/view.php?id=350>

Change Number: XSH/TC1/D5/0665 [106]

On Page: 2108 Line: 66718 Section: time()

In the RATIONALE section, add after the 2nd paragraph a new paragraph:

On some systems the time() function is implemented using a system call that does not return an error condition in addition to the return value. On these systems it is impossible to differentiate between valid and invalid return values and hence overflow conditions cannot be 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

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 **Change Number: XSH/TC1/D5/0675 [302]**
14963
14964
14965 On Page: 2129 Line: 67386 Section: towctrans()
14966
14967 In the DESCRIPTION section, change from:
14968
14969 ... current locale of the process ...
14970
14971 to:
14972
14973 ... current locale ...
14974
14975 *Rationale:* Austin Group Defect Report(s) applied: 302.
14976 See <http://austingroupbugs.net/view.php?id=302>
14977 These changes were overlooked during the revision when
14978 per-thread locales were added.
14979
14980 **Change Number: XSH/TC1/D5/0676 [283]**
14981
14982
14983 On Page: 2129 Line: 67395 Section: towctrans()
14984
14985 In the DESCRIPTION section, add a new paragraph to the end of the section:
14986
14987 [CX]The behavior is undefined if the locale argument to towctrans_l()
14988 is the special locale object LC_GLOBAL_LOCALE or is not a valid locale
14989 object handle.[/CX]
14990
14991 *Rationale:* Austin Group Defect Report(s) applied: 283.
14992 See <http://austingroupbugs.net/view.php?id=283>
14993
14994 **Change Number: XSH/TC1/D5/0677 [283]**
14995
14996
14997 On Page: 2129 Line: 67402 Section: towctrans()
15000
15001 In the ERRORS section, delete:
15002
15003 The towctrans_l() function may fail if:
15004
15005 [EINVAL] locale is not a valid locale object handle.
15006
15007 *Rationale:* Austin Group Defect Report(s) applied: 283.
15008 See <http://austingroupbugs.net/view.php?id=283>
15009

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()

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.

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

15274 The value of the file-position indicator for the stream after reading
15275 or discarding all pushed-back characters shall be the same as it was
15276 before the characters were pushed back.

15276
15277

to:

15278
15279

15280 The value of the file-position indicator for the stream after all
15281 pushed-back characters have been read, or discarded by calling fseek(),
15282 [CX]fseeko(), [CX] fsetpos(), or rewind() [CX] (but not fflush()) [CX],
15283 shall be the same as it was before the characters were pushed back.

15283
15284

Rationale: Austin Group Defect Report(s) applied: 87.

15285
15286

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
15297

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

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
15315

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

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

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 int status;
15547
15548 to:
15549
15550 int sav_errno = errno;
15551 int status;
15552

15553 *Rationale:* Austin Group Defect Report(s) applied: 166.
15554 See <http://austingroupbugs.net/view.php?id=166>
15555

15556
15557 **Change Number: XSH/TC1/D5/0709 [166]**
15558

15559
15560 On Page: 2186 Line: 68887 Section: wait()
15561
15562 In the EXAMPLES section, change from:

15563 }
15564 }
15565
15566 to:
15567
15568 }
15569 }
15570 }
15571 }
15572 }
15573

15574 *Rationale:* Austin Group Defect Report(s) applied: 166.
15575 See <http://austingroupbugs.net/view.php?id=166>
15576

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:

... with the status of the process. The `si_signo` member shall always be equal to `SIGCHLD`.

to:

... with the status of the process; the `si_signo` member shall be set equal to `SIGCHLD`. If `waitid()` returns because `WNOHANG` was specified and status is not available for any process specified by `idtype` and `id`, then the `si_signo` and `si_pid` members of the structure pointed to by `infop` shall be set to zero and the values of other members of the structure are unspecified.

Rationale: Austin Group Defect Report(s) applied: 153.
See <http://austingroupbugs.net/view.php?id=153>

Change Number: XSH/TC1/D5/0714 [88]

On Page: 2195 Line: 69147 Section: `wcrtomb()`

In the SYNOPSIS section, change from:

```
#include <stdio.h>
```

to:

```
#include <wchar.h>
```

Rationale: Austin Group Defect Report(s) applied: 88.
See <http://austingroupbugs.net/view.php?id=88>

Change Number: XSH/TC1/D5/0715 [105]

On Page: 2195 Line: 69168 Section: `wcrtomb()`

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

The `wcrtomb()` function shall not change the setting of `errno` if successful.

Rationale: Austin Group Defect Report(s) applied: 105.
See <http://austingroupbugs.net/view.php?id=105>

Change Number: XSH/TC1/D5/0716 [294]

On Page: 2197 Line: 69221 Section: `wscasecmp()`

In the DESCRIPTION section, change from:

When the `LC_CTIME` category of the current locale is from the POSIX locale, these functions shall behave as if the strings had been converted to lowercase and then a byte comparison performed. Otherwise, the results are unspecified.

The information for `wscasecmp_l()` and `wcsncasecmp_l()` about the case 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: `wcsnrtombs()`

15808

15809 In the DESCRIPTION section, change from:

15810

15811 [CX]The `wcsrtombs()` function need not be thread-safe ... [/CX]

15812

15813 to:

15814

15815 [CX]The `wcsnrtombs()` and `wcsrtombs()` functions need not be thread-safe

15816 ... [/CX]

15817

15818 The `wcsrtombs()` 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

15829 On Page: 2224 Line: 69988 Section: `wcstod()`
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: `wcstod()`
15849
15850 In the DESCRIPTION section, remove the CX shading from the paragraphs:
15851
15852 The `wcstod()` 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 `wcstod()`, `wcstof()`, or `wcstold()`, then check `errno`.
15857
15858 and then change the first paragraph:
15859
15860 The `wcstod()` 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 wcstoull(),
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

15955 object handle.[/CX]
15956
15957 *Rationale:* Austin Group Defect Report(s) applied: 283.
15958 See <http://austingroupbugs.net/view.php?id=283>
15959
15960
15961 **Change Number: XSH/TC1/D5/0730 [283]**
15962
15963
15964 On Page: 2242 Line: 70480 Section: wcsxfrm()
15965
15966 In the ERRORS section, delete:
15967
15968 The wcsxfrm_l() function may fail if:
15969
15970 [EINVAL] locale is not a valid locale object handle.
15971
15972 *Rationale:* Austin Group Defect Report(s) applied: 283.
15973 See <http://austingroupbugs.net/view.php?id=283>
15974
15975
15976 **Change Number: XSH/TC1/D5/0731 [302]**
15977
15978
15979 On Page: 2243 Line: 70486 Section: wcsxfrm()
15980
15981 change from:
15982
15983 ... the locale of the process ...
15984
15985 to:
15986
15987 ... the current locale ...
15988
15989 *Rationale:* Austin Group Defect Report(s) applied: 302.
15990 See <http://austingroupbugs.net/view.php?id=302>
15991 These changes were overlooked during the revision when
15992 per-thread locales were added.
15993
15994
15995 **Change Number: XSH/TC1/D5/0732 [302]**
15996
15997
15998 On Page: 2247 Line: 70616 Section: wctrans()
15999
16000 In the DESCRIPTION section, change from:
16001
16002 ... the locale of the process ...
16003
16004 to:
16005
16006 ... the current locale ...
16007
16008 *Rationale:* Austin Group Defect Report(s) applied: 302.
16009 See <http://austingroupbugs.net/view.php?id=302>
16010 These changes were overlooked during the revision when
16011 per-thread locales were added.
16012
16013
16014 **Change Number: XSH/TC1/D5/0733 [289]**
16015
16016
16017 On Page: 2247 Line: 70620 Section: wctrans()

16018

16019 In the DESCRIPTION section, change from:

16020

16021 ... only in calls to `wctrans_l()` ...

16022

16023 to:

16024

16025 ... only in calls to `towctrans_l()` ...

16026

16027 *Rationale:* Austin Group Defect Report(s) applied: 289.

16028 See <http://austingroupbugs.net/view.php?id=289>

16029

16030

16031 **Change Number: XSH/TC1/D5/0734** [283]

16032

16033

16034 On Page: 2247 Line: 70621 Section: `wctrans()`

16035

16036 In the DESCRIPTION section, add a new paragraph to the end of the section:

16037

16038 [CX]The behavior is undefined if the locale argument to `wctrans_l()`
16039 is the special locale object `LC_GLOBAL_LOCALE` or is not a valid locale
16040 object handle. [/CX]

16041

16042 *Rationale:* Austin Group Defect Report(s) applied: 283.

16043 See <http://austingroupbugs.net/view.php?id=283>

16044

16045

16046 **Change Number: XSH/TC1/D5/0735** [283]

16047

16048

16049 On Page: 2247 Line: 70631 Section: `wctrans()`

16050

16051 In the ERRORS section, delete:

16052

16053 The `wcstrans_l()` function may fail if:

16054

16055 [EINVAL] locale is not a valid locale object handle.

16056

16057 *Rationale:* Austin Group Defect Report(s) applied: 283.

16058 See <http://austingroupbugs.net/view.php?id=283>

16059

16060

16061 **Change Number: XSH/TC1/D5/0736** [302]

16062

16063

16064 On Page: 2249 Line: 70672 Section: `wctype()`

16065

16066 change from:

16067

16068 ... the locale of the process ...

16069

16070 to:

16071

16072 ... the current locale ...

16073

16074 *Rationale:* Austin Group Defect Report(s) applied: 302.

16075 See <http://austingroupbugs.net/view.php?id=302>

16076 These changes were overlooked during the revision when
16077 per-thread locales were added.

16078

16079

16080 **Change Number: XSH/TC1/D5/0737** [283]

16081
16082
16083
16084
16085
16086
16087
16088
16089
16090
16091
16092
16093
16094
16095
16096
16097
16098
16099
16100
16101
16102
16103
16104
16105
16106
16107
16108
16109
16110
16111
16112
16113
16114
16115
16116
16117
16118
16119
16120
16121
16122
16123
16124
16125
16126
16127
16128
16129
16130
16131
16132
16133
16134
16135
16136
16137
16138
16139
16140
16141
16142
16143

On Page: 2249 Line: 70677 Section: `wctype()`

In the DESCRIPTION section, add a new paragraph to the end of the section:

[CX]The behavior is undefined if the locale argument to `wctype_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/D5/0738 [283]

On Page: 2249 Line: 70683 Section: `wctype()`

In the ERRORS section, change from:

The `wctype_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/D5/0739 [460]

On Page: 2258 Line: 71005 Section: `wordexp()`

In the DESCRIPTION section, change from:

If `WRDE_SHOWERR` is set, `wordexp()` may write messages to `stderr` if syntax errors are detected while expanding words.

to:

If `WRDE_SHOWERR` is set, `wordexp()` may write messages to `stderr` if syntax errors are detected while expanding words; however, it is unspecified whether any write errors encountered while outputting such messages will affect the `stderr` error indicator or the value of `errno`.

Rationale: Austin Group Defect Report(s) applied: 460.
See <http://austingroupbugs.net/view.php?id=460>

Change Number: XSH/TC1/D5/0740 [291]

On Page: 2259 Line: 71031 Section: `wordexp()`

In the APPLICATION USAGE section, change from:

filename (or a list of filenames)

16144 to:
16145
16146 pathname (or a list of pathnames).
16147
16148 *Rationale:* Austin Group Defect Report(s) applied: 291.
16149 See <http://austingroupbugs.net/view.php?id=291>
16150
16151
16152 **Change Number: XSH/TC1/D5/0741** [460]
16153
16154
16155 On Page: 2259 Line: 71038 Section: wordexp()
16156
16157 In the APPLICATION USAGE section, add a new paragraph to the end of
16158 the section:
16159
16160 Even though the WRDE_SHOWERR flag allows the implementation to write
16161 messages to stderr during command substitution or syntax errors, the
16162 standard does not provide any way to detect write failures during
16163 the output of such messages.
16164
16165 *Rationale:* Austin Group Defect Report(s) applied: 460.
16166 See <http://austingroupbugs.net/view.php?id=460>
16167
16168
16169 **Change Number: XSH/TC1/D5/0742** [219]
16170
16171
16172 On Page: 2265 Line: 71222 Section: write()
16173
16174 In the DESCRIPTION section, append to the last paragraph in the section:
16175
16176 An attempt to perform a pwrite() on a file that is incapable of seeking
16177 shall result in an error.
16178
16179 *Rationale:* Austin Group Defect Report(s) applied: 219.
16180 See <http://austingroupbugs.net/view.php?id=219>
16181
16182
16183 **Change Number: XSH/TC1/D5/0743** [215]
16184
16185
16186 On Page: 2265-2267 Line: 71229,71245,71251,71255,71277 Section: write()
16187
16188 In the ERRORS section, at L71229 change from:
16189
16190 [EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the
16191 thread would be delayed in the write() operation.
16192
16193 to:
16194
16195 [EAGAIN] The file is neither a pipe, nor a FIFO, nor a socket, the
16196 O_NONBLOCK flag is set for the file descriptor, and the thread would be
16197 delayed in the write operation.
16198
16199 Move L71245-71247 ([EPIPE] on pipe/FIFO) to after L71254 ([ECONNRESET]).
16200
16201 Before L71250 insert:
16202
16203 The pwrite() function shall fail if:
16204
16205 [EINVAL] The file is a regular file or block special file, and the offset
16206 argument is negative. The file pointer shall remain unchanged.

16207
16208 [ESPIPE] The file is a pipe, FIFO, or socket.
16209
16210 At L71251 change from:
16211
16212 [EAGAIN] or [EWOULDBLOCK]
16213 The file descriptor is for a socket, is marked O_NONBLOCK, and write
16214 would block.
16215
16216 to:
16217
16218 [EAGAIN] The file is a pipe or FIFO, the O_NONBLOCK flag is set for the
16219 file descriptor, and the thread would be delayed in the write operation.
16220
16221 [EAGAIN] or [EWOULDBLOCK]
16222 The file is a socket, the O_NONBLOCK flag is set for the file descriptor,
16223 and the thread would be delayed in the write operation.
16224
16225 Delete L71275-71277 (the pwrite() "shall fail" errors).
16226
16227 *Rationale:* Austin Group Defect Report(s) applied: 215.
16228 See <http://austingroupbugs.net/view.php?id=215>
16229
16230
16231 **Change Number: XSH/TC1/D5/0744** [79]
16232
16233
16234 On Page: 2266 Line: 71241 Section: write()
16235
16236 In the ERRORS section [EIO], change from:
16237
16238 ... the process is neither ignoring nor blocking SIGTTOU, and the process
16239 group of the process is orphaned.
16240
16241 to:
16242
16243 ... the calling thread is not blocking SIGTTOU, the process is not
16244 ignoring SIGTTOU, and the process group of the process is orphaned.
16245
16246 *Rationale:* Austin Group Defect Report(s) applied: 79.
16247 See <http://austingroupbugs.net/view.php?id=79>
16248
16249
16250 **Change Number: XSH/TC1/D5/0745** [215]
16251
16252
16253 On Page: 2269 Line: 71388 Section: write()
16254
16255 In the RATIONALE section, add a new paragraph to the end of the section:
16256
16257 This volume of POSIX.1-2008 intentionally does not specify any pwrite()
16258 errors related to pipes, FIFOs, and sockets other than [ESPIPE].
16259
16260 *Rationale:* Austin Group Defect Report(s) applied: 215.
16261 See <http://austingroupbugs.net/view.php?id=215>
16262
16263
16264 **Change Number: XSH/TC1/D5/0746** [68]
16265
16266
16267 On Page: 2274 Line: 71531 Section: y0()
16268
16269 In the RETURN VALUE section, add MXX shading to:

If *x* is NaN, a NaN shall be returned.

Rationale: Austin Group Defect Report(s) applied: 68.
See <http://austingroupbugs.net/view.php?id=68>

3. Changes to Shell and Utilities

This section contains the set of changes to the text of the Shell and Utilities.

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

Change Number: XCU/TC1/D5/0001 [150]

On Page: 2294 Line: 72226 Section: 1.4 Consequences of Errors

Change from:

Default Behavior: When this section is listed as "Default.", it means that any changes to the environment are unspecified.

to:

Default Behavior: When this section is listed as "Default.", it means that any changes to the environment, file systems, process state, and so on are unspecified.

Rationale: Austin Group Defect Report(s) applied: 150.
See <http://austingroupbugs.net/view.php?id=150>

Change Number: XCU/TC1/D5/0002 [152]

On Page: 2303 Line: 72550 Section: 2.5.3 Shell Variables (ENV)

Change from:

ENV The processing of the ENV shell variable shall be supported on all XSI-conformant systems or if the system supports the User Portability Utilities option.

to:

ENV The processing of the ENV shell variable shall be supported if the system supports the User Portability Utilities option.

and change the shading from UP XSI to just UP.

Rationale: Austin Group Defect Report(s) applied: 152.
See <http://austingroupbugs.net/view.php?id=152>

Change Number: XCU/TC1/D5/0003 [49,430]

On Page: 2305 Line: 72669 Section: 2.6 Word Expansions

Change from:

16327 If an unquoted '\$' is followed by a character that is either not numeric,
16328 the name of one of the special parameters (see Section 2.5.2, on page
16329 2302), a valid first character of a variable name, a <left-curly-bracket>
16330 ('{') or a <left-parenthesis>, the result is unspecified.

16331
16332 to:

16333
16334 If an unquoted '\$' is followed by a character that is not one of the
16335 following:

- 16336
16337 * A numeric character
16338 * The name of one of the special parameters (see Section 2.5.2)
16339 * A valid first character of a variable name
16340 * A <left-curly-bracket> ('{')
16341 * A <left-parenthesis>

16342
16343 the result is unspecified.

16344
16345 *Rationale:* Austin Group Defect Report(s) applied: 49,430.
16346 See <http://austingroupbugs.net/view.php?id=49>
16347 See <http://austingroupbugs.net/view.php?id=430>
16348

16349
16350 **Change Number: XCU/TC1/D5/0004** [458]
16351

16352
16353 On Page: 2306 Line: 72701 Section: 2.6.2 Parameter Expansion
16354

16355 Change from:

16356
16357 ... or when parameter is followed by a character that could be
16358 interpreted as part of the name.

16359
16360 to:
16361
16362 ... or when parameter is a name and is followed by a character that
16363 could be interpreted as part of the name.

16364
16365 *Rationale:* Austin Group Defect Report(s) applied: 458.
16366 See <http://austingroupbugs.net/view.php?id=458>
16367

16368
16369 **Change Number: XCU/TC1/D5/0005** [458]
16370

16371
16372 On Page: 2306 Line: 72704 Section: 2.6.2 Parameter Expansion
16373

16374 Change from:

16375
16376 If the parameter name or symbol is not enclosed in braces, the expansion
16377 shall use the longest valid name (see XBD Section 3.230, on page 70),
16378 whether or not the symbol represented by that name exists.

16379
16380 to:
16381
16382 If the parameter is not enclosed in braces, and is a name, the expansion
16383 shall use the longest valid name (see XBD Section 3.230, on page 70),
16384 whether or not the variable represented by that name exists. Otherwise,
16385 the parameter is a single-character symbol, and behavior is unspecified
16386 if that character is neither a digit nor one of the special parameters
16387 (see XCU Section 2.5.2).
16388

16389 *Rationale:* Austin Group Defect Report(s) applied: 458.

See <http://austingroupbugs.net/view.php?id=458>

Change Number: XCU/TC1/D5/0006 [457]

On Page: 2306 Line: 72719 Section: 2.6.2 Parameter Expansion

Change from:

`${parameter:-word}` Use Default Values. If parameter is unset or null, the expansion of word shall be substituted; otherwise, the value of parameter shall be substituted.

`${parameter:=word}` Assign Default Values. If parameter is unset or null, the expansion of word shall be assigned to parameter.

to:

`${parameter:-[word]}` Use Default Values. If parameter is unset or null, the expansion of word (or an empty string if word is omitted) shall be substituted; otherwise, the value of parameter shall be substituted.

`${parameter:= [word]}` Assign Default Values. If parameter is unset or null, the expansion of word (or an empty string if word is omitted) shall be assigned to parameter.

Rationale: Austin Group Defect Report(s) applied: 457.

See <http://austingroupbugs.net/view.php?id=457>

Change Number: XCU/TC1/D5/0007 [457]

On Page: 2307 Line: 72730 Section: 2.6.2 Parameter Expansion

Change from:

`${parameter:+word}` Use Alternative Value. If parameter is unset or null, null shall be substituted; otherwise, the expansion of word shall be substituted.

to:

`${parameter:+[word]}` Use Alternative Value. If parameter is unset or null, null shall be substituted; otherwise, the expansion of word (or an empty string if word is omitted) shall be substituted.

Rationale: Austin Group Defect Report(s) applied: 457.

See <http://austingroupbugs.net/view.php?id=457>

Change Number: XCU/TC1/D5/0008 [417]

On Page: 2307 Line: 72734 Section: 2.6.2 Parameter Expansion

Insert a sentence before "The following table summarizes ...":

If parameter is '#' and the colon is omitted, the application shall ensure that word is specified (this is necessary to avoid ambiguity with the string length expansion).

Rationale: Austin Group Defect Report(s) applied: 417.

See <http://austingroupbugs.net/view.php?id=417>

16453
16454
16455 **Change Number: XCU/TC1/D5/0009** [457]
16456
16457
16458 On Page: 2307 Line: 72749 Section: 2.6.2 Parameter Expansion
16459
16460 Add a sentence to `${#parameter}`:
16461
16462 If parameter is unset and set `-u` is in effect, the expansion shall fail.
16463
16464 *Rationale:* Austin Group Defect Report(s) applied: 457.
16465 See <http://austingroupbugs.net/view.php?id=457>
16466
16467
16468 **Change Number: XCU/TC1/D5/0010** [457]
16469
16470
16471 On Page: 2307 Line: 72752 Section: 2.6.2 Parameter Expansion
16472
16473 Change from:
16474
16475 If parameter is `'*` or `'@'`, the result of the expansion is unspecified.
16476
16477 to:
16478
16479 If parameter is `'#'`, `'*'`, or `'@'`, the result of the expansion is
16480 unspecified. If parameter is unset and set `-u` is in effect, the
16481 expansion shall fail.
16482
16483 *Rationale:* Austin Group Defect Report(s) applied: 457.
16484 See <http://austingroupbugs.net/view.php?id=457>
16485
16486
16487 **Change Number: XCU/TC1/D5/0011** [457]
16488
16489
16490 On Page: 2307 Line: 72755 Section: 2.6.2 Parameter Expansion
16491
16492 Add a sentence:
16493
16494 In each variety, if word is omitted, the empty pattern shall be used.
16495
16496 *Rationale:* Austin Group Defect Report(s) applied: 457.
16497 See <http://austingroupbugs.net/view.php?id=457>
16498
16499
16500 **Change Number: XCU/TC1/D5/0012** [457]
16501
16502
16503 On Page: 2307 Line: 72756 Section: 2.6.2 Parameter Expansion
16504
16505 Change from:
16506
16507 `${parameter%word}`
16508
16509 to:
16510
16511 `${parameter%[word]}`
16512
16513 *Rationale:* Austin Group Defect Report(s) applied: 457.
16514 See <http://austingroupbugs.net/view.php?id=457>
16515

16516
16517 **Change Number: XCU/TC1/D5/0013** [457]
16518
16519
16520 On Page: 2307 Line: 72758 Section: 2.6.2 Parameter Expansion
16521
16522 Add a sentence:
16523
16524 If present, word shall not begin with an unquoted '%'.
16525
16526 *Rationale:* Austin Group Defect Report(s) applied: 457.
16527 See <http://austingroupbugs.net/view.php?id=457>
16528
16529
16530 **Change Number: XCU/TC1/D5/0014** [457]
16531
16532
16533 On Page: 2307 Line: 72759 Section: 2.6.2 Parameter Expansion
16534
16535 Change from:
16536
16537 \${parameter%%word}
16538
16539 to:
16540
16541 \${parameter%[word]}
16542
16543 *Rationale:* Austin Group Defect Report(s) applied: 457.
16544 See <http://austingroupbugs.net/view.php?id=457>
16545
16546
16547 **Change Number: XCU/TC1/D5/0015** [457]
16548
16549
16550 On Page: 2307 Line: 72762 Section: 2.6.2 Parameter Expansion
16551
16552 Change from:
16553
16554 \${parameter#word}
16555
16556 to:
16557
16558 \${parameter#[word]}
16559
16560 *Rationale:* Austin Group Defect Report(s) applied: 457.
16561 See <http://austingroupbugs.net/view.php?id=457>
16562
16563
16564 **Change Number: XCU/TC1/D5/0016** [457]
16565
16566
16567 On Page: 2307 Line: 72764 Section: 2.6.2 Parameter Expansion
16568
16569 Add a sentence:
16570
16571 If present, word shall not begin with an unquoted '#'.
16572
16573 *Rationale:* Austin Group Defect Report(s) applied: 457.
16574 See <http://austingroupbugs.net/view.php?id=457>
16575
16576
16577 **Change Number: XCU/TC1/D5/0017** [457]
16578

On Page: 2307 Line: 72765 Section: 2.6.2 Parameter Expansion

Change from:

```
${parameter##word}
```

to:

```
${parameter##[word]}
```

Rationale: Austin Group Defect Report(s) applied: 457.

See <http://austingroupbugs.net/view.php?id=457>

Change Number: XCU/TC1/D5/0018 [458]

On Page: 2308 Line: 72768 Section: 2.6.2 Parameter Expansion

In the Examples, insert (with the last line in bold):

```
${parameter}
```

```
    In this example, the effects of omitting braces are demonstrated.
```

```
    a=1
```

```
    set 2
```

```
    echo ${a}b-$ab-${1}0-${10}-${10}
```

```
    1b--20--20
```

Rationale: Austin Group Defect Report(s) applied: 458.

See <http://austingroupbugs.net/view.php?id=458>

Change Number: XCU/TC1/D5/0019 [217]

On Page: 2309 Line: 72840 Section: 2.6.3 Command Substitution

Change from:

If the command substitution consists of a single subshell, such as:

```
$( (command) )
```

a conforming application shall separate the "\$(" and "(" into two tokens (that is, separate them with white space). This is required to avoid any ambiguities with arithmetic expansion.

to:

The syntax of the shell command language has an ambiguity for expansions beginning with "\$(", which can introduce an arithmetic expansion or a command substitution that starts with a subshell. Arithmetic expansion has precedence; that is, the shell shall first determine whether it can parse the expansion as an arithmetic expansion and shall only parse the expansion as a command substitution if it determines that it cannot parse the expansion as an arithmetic expansion. The shell need not evaluate nested expansions when performing this determination. If it encounters the end of input without already having determined that it cannot parse the expansion as an arithmetic expansion, the shell shall treat the expansion as an incomplete arithmetic expansion and report a syntax error. A conforming application shall ensure that it separates the "\$(" and "(" into two tokens (that is, separate them with white space)

in a command substitution that starts with a subshell. For example,
a command substitution containing a single subshell could be written as:

```
$( (command) )
```

Rationale: Austin Group Defect Report(s) applied: 217.
See <http://austingroupbugs.net/view.php?id=217>

Change Number: XCU/TC1/D5/0020 [50]

On Page: 2310 Line: 72861 Section: 2.6.4 Arithmetic Expansion

Change from:

If the shell variable *x* contains a value that forms a valid integer
constant, then ...

to:

If the shell variable *x* contains a value that forms a valid integer
constant, optionally including a leading plus or minus sign, then ...

At L72866, change from:

If the expression is invalid, the expansion fails and the shell shall
write a message to standard error indicating the failure.

to:

If the expression is invalid, or the contents of a shell variable used in
the expression are not recognized by the shell, the expansion fails and
the shell shall write a diagnostic message to standard error indicating
the failure.

Rationale: Austin Group Defect Report(s) applied: 50.
See <http://austingroupbugs.net/view.php?id=50>

Change Number: XCU/TC1/D5/0021 [255]

On Page: 2317 Line: 73101 Section: 2.9.1 Simple Commands

Change:

If no command name results, variable assignments shall affect the current
execution environment. Otherwise, the variable assignments shall be
exported for the execution environment of the command and shall not affect
the current execution environment (except for special built-ins). If any
of the variable assignments attempt to assign a value to a read-only
variable, a variable assignment error shall occur. See Section 2.8.1
(Consequences of Shell Errors) for the consequences of these errors.

to:

If no command name results, or if the command name is a special built-in
or function, variable assignments shall affect the current execution
environment. Otherwise, the variable assignments shall be exported for
the execution environment of the command and shall not affect the current
execution environment except as a side-effect of the expansions performed
in step 4. In this case it is unspecified:

16705

16706 * Whether or not the assignments are visible for subsequent expansions
16707 in step 4

16708

16709 * Whether variable assignments made as side-effects of these expansions
16710 are visible for subsequent expansions in step 4, or in the current shell
16711 execution environment, or both

16712

16713 If any of the variable assignments attempt to assign a value to a variable
16714 for which the readonly attribute is set in the current shell environment
16715 (regardless of whether the assignment is made in that environment), a
16716 variable assignment error shall occur. See Section 2.8.1 (Consequences
16717 of Shell Errors) for the consequences of these errors.

16718

16719 *Rationale:* Austin Group Defect Report(s) applied: 255.

16720 See <http://austingroupbugs.net/view.php?id=255>

16721

16722

16723 **Change Number: XCU/TC1/D5/0022** [168]

16724

16725

16726 On Page: 2317 Line: 73143 Section: 2.9.1.1

16727

16728 Change from:

16729

16730 ... calling the `execve()` function as defined in the System Interfaces
16731 volume of POSIX.1-2008 with the path argument set to the pathname
16732 resulting from the search, `arg0` set to the command name, and the remaining
16733 arguments set to the operands, if any.

16734

16735 to:

16736

16737 ... calling the `execl()` function as defined in the System Interfaces
16738 volume of POSIX.1-2008 with the path argument set to the pathname
16739 resulting from the search, `arg0` set to the command name, and the
16740 remaining `execl()` arguments set to the command arguments (if any) and
16741 the null terminator.

16742

16743 *Rationale:* Austin Group Defect Report(s) applied: 168.

16744 See <http://austingroupbugs.net/view.php?id=168>

16745

16746

16747 **Change Number: XCU/TC1/D5/0023** [168]

16748

16749

16750 On Page: 2318 Line: 73147 Section: 2.9.1.1

16751

16752 Change from:

16753

16754 If the `execve()` function fails ...

16755

16756 to:

16757

16758 If the `execl()` function fails ...

16759

16760 *Rationale:* Austin Group Defect Report(s) applied: 168.

16761 See <http://austingroupbugs.net/view.php?id=168>

16762

16763

16764 **Change Number: XCU/TC1/D5/0024** [168]

16765

16766

16767 On Page: 2318 Line: 73155 Section: 2.9.1.1

16768

16769 After L73155 add a new paragraph (indented the same as the previous
16770 paragraph):

16771

16772 It is unspecified whether environment variables that were passed to the
16773 shell when it was invoked, but were not used to initialize shell variables
16774 (see Section 2.5.3) because they had invalid names, are included in the
16775 environment passed to `execl()` and (if `execl()` fails as described above)
16776 to the new shell.

16777

16778 *Rationale:* Austin Group Defect Report(s) applied: 168.

16779 See <http://austingroupbugs.net/view.php?id=168>

16780

16781

16782 **Change Number: XCU/TC1/D5/0025** [168]

16783

16784

16785 On Page: 2318 Line: 73165 Section: 2.9.1.1

16786

16787 Change from:

16788

16789 ... calling the `execve()` function defined in the System Interfaces volume
16790 of POSIX.1-2008 with the path and `arg0` arguments set to the command name,
16791 and the remaining arguments set to the operands, if any.

16792

16793 to:

16794

16795 ... calling the `execl()` function as defined in the System Interfaces
16796 volume of POSIX.1-2008 with the path and `arg0` arguments set to the command
16797 name, and the remaining `execl()` arguments set to the command arguments
16798 (if any) and the null terminator.

16799

16800 *Rationale:* Austin Group Defect Report(s) applied: 168.

16801 See <http://austingroupbugs.net/view.php?id=168>

16802

16803

16804 **Change Number: XCU/TC1/D5/0026** [168,430]

16805

16806

16807 On Page: 2318 Line: 73169 Section: 2.9.1.1

16808

16809 Change from:

16810

16811 If the `execve()` function fails ...

16812

16813 to:

16814

16815 If the `execl()` function fails ...

16816

16817 *Rationale:* Austin Group Defect Report(s) applied: 168,430.

16818 See <http://austingroupbugs.net/view.php?id=168>

16819 See <http://austingroupbugs.net/view.php?id=430>

16820

16821

16822 **Change Number: XCU/TC1/D5/0027** [168,430]

16823

16824

16825 On Page: 2318 Line: 73173 Section: 2.9.1.1

16826

16827 After L73173 add a new paragraph (indented the same as the previous
16828 paragraph):

16829

16830 It is unspecified whether environment variables that were passed to the

16831 shell when it was invoked, but were not used to initialize shell variables
16832 (see Section 2.5.3) because they had invalid names, are included in the
16833 environment passed to `execl()` and (if `execl()` fails as described above)
16834 to the new shell.

16835
16836 *Rationale:* Austin Group Defect Report(s) applied: 168,430.
16837 See <http://austingroupbugs.net/view.php?id=168>
16838 See <http://austingroupbugs.net/view.php?id=430>
16839
16840

16841 **Change Number: XCU/TC1/D5/0028** [173]
16842
16843

16844 On Page: 2318 Line: 73173 Section: 2.9.1.1
16845

16846 Add at the end of the section after the numbered list (indented the same
16847 as L73119):
16848

16849 If the utility would be executed with file descriptor 0, 1, or 2 closed,
16850 implementations may execute the utility with the file descriptor open to
16851 an unspecified file. If a standard utility or a conforming application
16852 is executed with file descriptor 0 not open for reading or with file
16853 descriptor 1 or 2 not open for writing, the environment in which the
16854 utility or application is executed shall be deemed non-conforming, and
16855 consequently the utility or application might not behave as described
16856 in this standard.

16857
16858 *Rationale:* Austin Group Defect Report(s) applied: 173.
16859 See <http://austingroupbugs.net/view.php?id=173>
16860
16861

16862 **Change Number: XCU/TC1/D5/0029** [205]
16863
16864

16865 On Page: 2318 Line: 73175 Section: 2.9.2 Pipelines
16866

16867 Change from:
16868

16869 The standard output of all but the last command shall be connected to
16870 the standard input of the next command.

16871 to:
16872
16873

16874 For each command but the last, the shell shall connect the standard
16875 output of the command to the standard input of the next command as if
16876 by creating a pipe and passing the write end of the pipe as the standard
16877 output of the command and the read end of the pipe as the standard input
16878 of the next command.

16879
16880 *Rationale:* Austin Group Defect Report(s) applied: 205.
16881 See <http://austingroupbugs.net/view.php?id=205>
16882
16883

16884 **Change Number: XCU/TC1/D5/0030** [52]
16885
16886

16887 On Page: 2319 Line: 73188 Section: 2.9.2 Exit Status
16888

16889 Change from:
16890

16891 If the reserved word ! does not precede the pipeline, the ...
16892
16893

to:

If the pipeline does not begin with the ! reserved word, the ...

Rationale: Austin Group Defect Report(s) applied: 52.
See <http://austingroupbugs.net/view.php?id=52>

Change Number: XCU/TC1/D5/0031 [45]

On Page: 2319 Line: 73195-73196 Section: 2.9.3 Lists

Change from:

A list is a sequence of one or more AND-OR lists separated by the operators ; and & and optionally terminated by ;, &, or <newline>.

to:

A list is a sequence of one or more AND-OR lists separated by the operators ; and &.

Rationale: Austin Group Defect Report(s) applied: 45.
See <http://austingroupbugs.net/view.php?id=45>

Change Number: XCU/TC1/D5/0032 [45]

On Page: 2319 Line: 73201-73202 Section: 2.9.3 Lists

Change from:

A ';' or <newline> terminator shall cause the preceding AND-OR list to be executed sequentially; an '&' shall cause asynchronous execution of the preceding AND-OR list.

to:

A ';' separator or a ';' or <newline> terminator shall cause the preceding AND-OR list to be executed sequentially; an '&' separator or terminator shall cause asynchronous execution of the preceding AND-OR list.

Rationale: Austin Group Defect Report(s) applied: 45.
See <http://austingroupbugs.net/view.php?id=45>

Change Number: XCU/TC1/D5/0033 [217]

On Page: 2321 Line: 73276 Section: 2.9.4.1 Grouping Commands

Add a new paragraph to the end of the description of "(compound-list)":

If a character sequence beginning with "(" would be parsed by the shell as an arithmetic expansion if preceded by a '\$', shells which implement an extension whereby "((expression))" is evaluated as an arithmetic expression may treat the "(((" as introducing as an arithmetic evaluation instead of a grouping command. A conforming application shall ensure that it separates the two leading '(' characters with white space to prevent the shell from performing an arithmetic evaluation.

Rationale: Austin Group Defect Report(s) applied: 217.

16957 See <http://austingroupbugs.net/view.php?id=217>

16958

16959

16960 **Change Number: XCU/TC1/D5/0034** [383]

16961

16962

16963 On Page: 2324 Line: 73380 Section: 2.9.5 Function Definition Command

16964

16965 Change:

16966

16967 The function is named `fname()`; the application shall ensure that it is
16968 a name (see XBD Section 3.230, on page 70).

16969

16970 to:

16971

16972 The function is named `fname()`; the application shall ensure that it is
16973 a name (see XBD Section 3.230, on page 70) and that it is not the name
16974 of a special built-in utility.

16975

16976 *Rationale:* Austin Group Defect Report(s) applied: 383.

16977 See <http://austingroupbugs.net/view.php?id=383>

16978

16979

16980 **Change Number: XCU/TC1/D5/0035** [214]

16981

16982

16983 On Page: 2324 Line: 73400 Section: 2.9.5

16984

16985 Change from:

16986

16987 If the special built-in return is executed in the compound command,
16988 the function completes and execution shall resume with the next command
16989 after the function call.

16990

16991 to:

16992

16993 If the special built-in return (see the return special built-in utility)
16994 is executed in the compound command, the function completes and execution
16995 shall resume with the next command after the function call.

16996

16997 *Rationale:* Austin Group Defect Report(s) applied: 214.

16998 See <http://austingroupbugs.net/view.php?id=214>

16999

17000

17001 **Change Number: XBD/TC1/D5/0036** [44]

17002

17003

17004 On Page: 2329 Line: 73599–73601 Section: 2.10.2 Shell Grammar Rules

17005

17006 Change from:

17007

17008 `else_part : Elif compound_list Then else_part`
17009 `| Else compound_list`
17010 `;`

17011 to:

17012

17013 `else_part : Elif compound_list Then compound_list`
17014 `| Elif compound_list Then compound_list else_part`
17015 `| Else compound_list`
17016 `;`

17017

17018 *Rationale:* Austin Group Defect Report(s) applied: 44.

17019 See <http://austingroupbugs.net/view.php?id=44>

17020
17021
17022 **Change Number: XCU/TC1/D5/0037** [238]
17023
17024
17025 On Page: 2331 Line: 73718 Section: 2.12 Shell Execution Environment
17026
17027 Change from:
17028
17029 A subshell environment shall be created as a duplicate of the shell
17030 environment, except that signal traps set by that shell environment
17031 shall be set to the default values.
17032
17033 to:
17034
17035 A subshell environment shall be created as a duplicate of the shell
17036 environment, except that signal traps that are not being ignored shall
17037 be set to the default action
17038
17039 *Rationale:* Austin Group Defect Report(s) applied: 238.
17040 See <http://austingroupbugs.net/view.php?id=238>
17041
17042
17043 **Change Number: XCU/TC1/D5/0038** [114]
17044
17045
17046 On Page: 2341 Line: 74053 Section: dot
17047
17048 In the EXIT STATUS section, change from:
17049
17050 Returns the value of the last command executed, or a zero exit status
17051 if no command is executed.
17052
17053 to:
17054
17055 If no readable file was found or if the commands in the file could not
17056 be parsed, and the shell is interactive (and therefore does not abort;
17057 see Section 2.8.1), the exit status shall be non-zero. Otherwise,
17058 return the value of the last command executed, or a zero exit status if
17059 no command is executed.
17060
17061 *Rationale:* Austin Group Defect Report(s) applied: 114.
17062 See <http://austingroupbugs.net/view.php?id=114>
17063
17064
17065 **Change Number: XCU/TC1/D5/0039** [214]
17066
17067
17068 On Page: 2342 Line: 74074 Section: dot
17069
17070 In the SEE ALSO section, add a cross-reference to the return special
17071 built-in utility.
17072
17073 *Rationale:* Austin Group Defect Report(s) applied: 214.
17074 See <http://austingroupbugs.net/view.php?id=214>
17075
17076
17077 **Change Number: XCU/TC1/D5/0040** [114]
17078
17079
17080 On Page: 2343 Line: 74110 Section: eval
17081
17082 In the EXIT STATUS section, change from:

17083
17084 If there are no arguments, or only null arguments, eval shall return
17085 a zero exit status; otherwise, it shall return the exit status of the
17086 command defined by the string of concatenated arguments separated by
17087 <space> characters.
17088
17089 to:
17090
17091 If there are no arguments, or only null arguments, eval shall return
17092 a zero exit status; otherwise, it shall return the exit status of the
17093 command defined by the string of concatenated arguments separated by
17094 <space> characters, or a non-zero exit status if the concatenation could
17095 not be parsed as a command and the shell is interactive (and therefore
17096 did not abort).
17097
17098 *Rationale:* Austin Group Defect Report(s) applied: 114.
17099 See <http://austingroupbugs.net/view.php?id=114>
17100
17101
17102 **Change Number: XCU/TC1/D5/0041** [163]
17103
17104
17105 On Page: 2343 Line: 74116 Section: eval
17106
17107 In the APPLICATION USAGE section, change from:
17108
17109 None.
17110
17111 to:
17112
17113 Since eval is not required to recognize the "--" end of options
17114 delimiter, in cases where the argument(s) to eval might begin with '-'
17115 it is recommended that the first argument is prefixed by a string that
17116 will not alter the commands to be executed, such as a space character:
17117
17118 eval " \$commands"
17119
17120 or:
17121
17122 eval " \$(some_command) "
17123
17124 *Rationale:* Austin Group Defect Report(s) applied: 163.
17125 See <http://austingroupbugs.net/view.php?id=163>
17126
17127
17128 **Change Number: XCU/TC1/D5/0042** [163]
17129
17130
17131 On Page: 2344 Line: 74126 Section: eval
17132
17133 In the RATIONALE section, change from:
17134
17135 None.
17136
17137 to:
17138
17139 This standard allows, but does not require, eval to recognize "--".
17140 Although this means applications cannot use "--" to protect against
17141 options supported as an extension (or errors reported for unsupported
17142 options), the nature of the eval utility is such that other means can
17143 be used to provide this protection (see APPLICATION USAGE above).
17144
17145 *Rationale:* Austin Group Defect Report(s) applied: 163.

17146 See <http://austingroupbugs.net/view.php?id=163>
17147
17148
17149 **Change Number: XCU/TC1/D5/0043 [352]**
17150
17151
17152 On Page: 2349 Line: 74290 Section: export
17153
17154 At L74290, add a sentence:
17155
17156 If a variable assignment precedes the command name of export but that
17157 variable is not also listed as an operand of export, then that variable
17158 shall be set in the current shell execution environment after the
17159 completion of the export command, but it is unspecified whether that
17160 variable is marked for export.
17161
17162 *Rationale:* Austin Group Defect Report(s) applied: 352.
17163 See <http://austingroupbugs.net/view.php?id=352>
17164 The standard does not allow long-standing existing practice.
17165
17166
17167 **Change Number: XCU/TC1/D5/0044 [214]**
17168
17169
17170 On Page: 2355 Line: 74432 Section: return
17171
17172 In the NAME section, change from:
17173
17174 return - return from a function
17175
17176 to:
17177
17178 return - return from a function or dot script
17179
17180 *Rationale:* Austin Group Defect Report(s) applied: 214.
17181 See <http://austingroupbugs.net/view.php?id=214>
17182
17183
17184 **Change Number: XCU/TC1/D5/0045 [214]**
17185
17186
17187 On Page: 2356 Line: 74480 Section: return
17188
17189 In the SEE ALSO section, add a cross-reference to:
17190
17191 Section 2.9.5 Function Definition Command
17192 dot special built-in
17193
17194 *Rationale:* Austin Group Defect Report(s) applied: 214.
17195 See <http://austingroupbugs.net/view.php?id=214>
17196
17197
17198 **Change Number: XCU/TC1/D5/0046 [52]**
17199
17200
17201 On Page: 2358 Line: 74545 Section: 2.14 set
17202
17203 In the set special builtin utility, DESCRIPTION section, replace the
17204 description of -e with:
17205
17206 -e When this option is on, when any command fails (for any of the reasons
17207 listed in Section 2.8.1 (on page xxx) or by returning an exit status
17208 greater than zero) the shell immediately shall exit with the following

17209 exceptions:
17210
17211 1. The failure of any individual command in a multi-command pipeline
17212 shall not cause the shell to exit. Only the failure of the pipeline
17213 itself shall be considered.
17214
17215 2. The -e setting shall be ignored when executing the compound list
17216 following the while, until, if, or elif reserved word, a pipeline
17217 beginning with the ! reserved word, or any command of an AND-OR list
17218 other than the last.
17219
17220 3. If the exit status of a compound command other than a subshell command
17221 was the result of a failure while -e was being ignored, then -e shall
17222 not apply to this command.
17223
17224 This requirement applies to the shell environment and each subshell
17225 environment separately. For example, in:
17226
17227 set -e; (false; echo one) | cat; echo two
17228
17229 the false command causes the subshell to exit without executing echo one;
17230 however, echo two is executed because the exit status of the pipeline
17231 (false; echo one) | cat is zero.
17232
17233 *Rationale:* Austin Group Defect Report(s) applied: 52.
17234 See <http://austingroupbugs.net/view.php?id=52>
17235
17236
17237 **Change Number: XCU/TC1/D5/0047 [155,280]**
17238
17239
17240 On Page: 2359 Line: 74590 Section: 2.14 set
17241
17242 In the DESCRIPTION section, change from:
17243
17244 -u The shell shall write a message to standard error when it tries to
17245 expand a variable that is not set and immediately exit. An interactive
17246 shell shall not exit.
17247
17248 to:
17249
17250 -u When the shell tries to expand an unset parameter other than the '@'
17251 and '*' special parameters, it shall write a message to standard error
17252 and shall not execute the command containing the expansion, but for the
17253 purposes of setting the '?' special parameter and the exit status of the
17254 shell the command shall be treated as having been executed and returned
17255 an exit status of between 1 and 125 inclusive. A non-interactive shell
17256 shall immediately exit. An interactive shell shall not exit.
17257
17258 *Rationale:* Austin Group Defect Report(s) applied: 155,280.
17259 See <http://austingroupbugs.net/view.php?id=155>
17260 See <http://austingroupbugs.net/view.php?id=280>
17261
17262
17263 **Change Number: XCU/TC1/D5/0048 [52]**
17264
17265
17266 On Page: 2360 Line: 74631 Section: 2.14 set
17267
17268 In the APPLICATION USAGE section, change from:
17269
17270 None.
17271

17272 to:
17273
17274 Application writers should avoid relying on set -e within functions. For
17275 example, in the following script:
17276
17277 set -e
17278 start() {
17279 some_server
17280 echo some_server started successfully
17281 }
17282 start || echo ">2 some_server failed
17283
17284 the -e setting is ignored within the function body (because the function
17285 is a command in an AND-OR list other than the last). Therefore, if
17286 some_server fails, the function carries on to echo "server started
17287 successfully", and the exit status of the function is zero (which means
17288 "server failed" is not output).
17289
17290 *Rationale:* Austin Group Defect Report(s) applied: 52.
17291 See <http://austingroupbugs.net/view.php?id=52>
17292
17293
17294 **Change Number: XCU/TC1/D5/0049 [52]**
17295
17296
17297 On Page: 2361 Line: 74660 Section: 2.14 set
17298
17299 In the RATIONALE after L74660, add a new paragraph:
17300
17301 The description of the -e option is intended to match the behavior of
17302 the 1988 version of the KornShell.
17303
17304 *Rationale:* Austin Group Defect Report(s) applied: 52.
17305 See <http://austingroupbugs.net/view.php?id=52>
17306
17307
17308 **Change Number: XCU/TC1/D5/0050 [155,430]**
17309
17310
17311 On Page: 2362 Line: 74714 Section: 2.14 set
17312
17313 In the RATIONALE section, add a new paragraph at the end of the section:
17314
17315 Historically, some shells applied the -u option to all parameters including
17316 \$@ and \$*. The standard developers felt that this was a misfeature
17317 since it is normal and common for \$@ and \$* to be used in shell scripts
17318 regardless of whether they were passed any arguments. Treating these
17319 uses as an error when no arguments are passed reduces the value of -u
17320 for its intended purpose of finding spelling mistakes in variable names
17321 and uses of unset positional parameters.
17322
17323 *Rationale:* Austin Group Defect Report(s) applied: 155,430.
17324 See <http://austingroupbugs.net/view.php?id=155>
17325 See <http://austingroupbugs.net/view.php?id=430>
17326
17327
17328 **Change Number: XCU/TC1/D5/0051 [459]**
17329
17330
17331 On Page: 2364 Line: 74772 Section: shift
17332
17333 In the EXIT STATUS section, change from:
17334

17335 The exit status is >0 if n>\$#; otherwise, it is zero.
17336
17337 to:
17338
17339 If the n operand is invalid or is greater than \$#, this may be considered
17340 a syntax error and a non-interactive shell may exit; if the shell does not
17341 exit in this case, a non-zero exit status shall be returned. Otherwise,
17342 zero shall be returned.
17343
17344 *Rationale:* Austin Group Defect Report(s) applied: 459.
17345 See <http://austingroupbugs.net/view.php?id=459>
17346
17347
17348 **Change Number: XCU/TC1/D5/0052** [53,268,440]
17349
17350
17351 On Page: 2368 Line: 74879 Section: 2.14 trap
17352
17353 In the trap special builtin utility DESCRIPTION section, change from:
17354
17355 When a subshell is entered, traps that are not being ignored are set to
17356 the default actions.
17357
17358 to:
17359
17360 When a subshell is entered, traps that are not being ignored shall be set
17361 to the default actions, except in the case of a command substitution
17362 containing only a single trap command, when the traps need not be
17363 altered. Implementations may check for this case using only lexical
17364 analysis; for example, if `trap` and \$(trap --) do not alter the traps
17365 in the subshell, cases such as assigning var=trap and then using \${\$var}
17366 may still alter them.
17367
17368 *Rationale:* Austin Group Defect Report(s) applied: 53,268,440.
17369 See <http://austingroupbugs.net/view.php?id=53>
17370 See <http://austingroupbugs.net/view.php?id=268>
17371 See <http://austingroupbugs.net/view.php?id=440>
17372
17373
17374 **Change Number: XCU/TC1/D5/0053** [53,268,440]
17375
17376
17377 On Page: 2368 Line: 74881 Section: 2.14 trap
17378
17379 In the trap special builtin utility DESCRIPTION section, change from:
17380
17381 The trap command with no arguments shall write to standard output a list
17382 of commands associated with each condition.
17383
17384 to:
17385
17386 The trap command with no operands shall write to standard output a list
17387 of commands associated with each condition. If the command is executed
17388 in a subshell, the implementation does not perform the optional check
17389 described above for a command substitution containing only a single
17390 trap command, and no trap commands with operands have been executed
17391 since entry to the subshell, the list shall contain the commands that
17392 were associated with each condition immediately before the subshell
17393 environment was entered. Otherwise, the list shall contain the commands
17394 currently associated with each condition.
17395
17396 *Rationale:* Austin Group Defect Report(s) applied: 53,268,440.
17397 See <http://austingroupbugs.net/view.php?id=53>

17398 See <http://austingroupbugs.net/view.php?id=268>
17399 See <http://austingroupbugs.net/view.php?id=440>
17400
17401
17402 **Change Number: XCU/TC1/D5/0054** [163]
17403
17404
17405 On Page: 2369 Line: 74932 Section: 2.14 trap
17406
17407 In the EXAMPLES section, change:
17408
17409 '\$HOME/logout'
17410
17411 to:
17412
17413 '\$HOME"/logout'
17414
17415 *Rationale:* Austin Group Defect Report(s) applied: 163.
17416 See <http://austingroupbugs.net/view.php?id=163>
17417
17418
17419 **Change Number: XCU/TC1/D5/0055** [163]
17420
17421
17422 On Page: 2370 Line: 74934 Section: 2.14 trap
17423
17424 In the EXAMPLES section, change:
17425
17426 '\$HOME/logout'
17427
17428 to:
17429
17430 '\$HOME"/logout'
17431
17432 *Rationale:* Austin Group Defect Report(s) applied: 163.
17433 See <http://austingroupbugs.net/view.php?id=163>
17434
17435
17436 **Change Number: XCU/TC1/D5/0056** [163]
17437
17438
17439 On Page: 2370 Line: 74956 Section: 2.14 trap
17440
17441 In the RATIONALE section, change from:
17442
17443 The command:
17444
17445 trap '\$cmd' 0
17446
17447 causes the contents of the shell variable cmd to be executed as a command
17448 when the shell exits. Using double-quotes instead of single-quotes might
17449 have unexpected behavior, since in theory the value of cmd might be a
17450 decimal integer which would be treated as a condition, not an action;
17451 or cmd might begin with '-'. Also, using double-quotes will cause the
17452 value of cmd to be expanded twice, once when trap is executed, and once
17453 when the condition arises.
17454
17455 to:
17456
17457 The command:
17458
17459 trap 'eval " \$cmd"' 0
17460

17461 causes the contents of the shell variable `cmd` to be executed as a command
17462 when the shell exits. Using:

17463
17464 `trap '$cmd' 0`

17465
17466 does not work correctly if `cmd` contains any special characters such as
17467 quoting or redirections. Using:

17468
17469 `trap " $cmd" 0`

17470
17471 also works (the leading space character protects against unlikely cases
17472 where `cmd` is a decimal integer or begins with '-'), but it expands the
17473 `cmd` variable when the trap command is executed, not when the exit action
17474 is executed.

17475
17476 *Rationale:* Austin Group Defect Report(s) applied: 163.
17477 See <http://austingroupbugs.net/view.php?id=163>

17478
17479
17480 **Change Number:** XCU/TC1/D5/0057 [224]

17481
17482
17483 On Page: 2430 Line: 77077-77078 Section: `awk`

17484
17485 In the SYNOPSIS section (on L77077 and L77078) change from:

17486
17487 `-F ERE`

17488
17489 to:
17490
17491 `-F sepstring`

17492
17493 *Rationale:* Austin Group Defect Report(s) applied: 224.
17494 See <http://austingroupbugs.net/view.php?id=224>

17495
17496
17497 **Change Number:** XCU/TC1/D5/0058 [454]

17498
17499
17500 On Page: 2430 Line: 77089 Section: `awk`

17501
17502 In the DESCRIPTION section, change from:

17503
17504 ... by default, a field is a string of non-<blank> characters.
17505 This default white-space field delimiter can ...

17506
17507 to:
17508
17509 ... by default, a field is a string of non-<blank> non-<newline>
17510 characters. This default <blank> and <newline> field delimiter can ...

17511
17512 *Rationale:* Austin Group Defect Report(s) applied: 454.
17513 See <http://austingroupbugs.net/view.php?id=454>

17514
17515
17516 **Change Number:** XCU/TC1/D5/0059 [224]

17517
17518
17519 On Page: 2430 Line: 77091 Section: `awk`

17520
17521 In the DESCRIPTION section, change from:

17522
17523 `-F ERE`

17524
17525 to:
17526
17527 the -F sepstring option
17528
17529 *Rationale:* Austin Group Defect Report(s) applied: 224.
17530 See <http://austingroupbugs.net/view.php?id=224>
17531
17532
17533 **Change Number: XCU/TC1/D5/0060 [224]**
17534
17535
17536 On Page: 2430 Line: 77098 Section: awk
17537
17538 In the OPTIONS section, change from:
17539
17540 -F ERE
17541 Define the input field separator to be the extended regular expression
17542 ERE>, before any input is read; see Regular Expressions.
17543
17544 to:
17545
17546 -F sepstring
17547 Define the input field separator. This option shall be equivalent to:
17548
17549 -v FS=sepstring
17550
17551 except that if -F sepstring and -v FS=sepstring are both used, it
17552 is unspecified whether the FS assignment resulting from -F sepstring
17553 is processed in command line order or is processed after the last -v
17554 FS=sepstring. See the description of the FS built-in variable, and how
17555 it is used, in the EXTENDED DESCRIPTION section.
17556
17557 *Rationale:* Austin Group Defect Report(s) applied: 224.
17558 See <http://austingroupbugs.net/view.php?id=224>
17559
17560
17561 **Change Number: XCU/TC1/D5/0061 [254]**
17562
17563
17564 On Page: 2430 Line: 77100 Section: awk
17565
17566 In the OPTIONS section, after:
17567
17568 ... -f progfile Specify the pathname of the file progfile containing
17569 an awk program.
17570
17571 add:
17572
17573 A pathname of '-' shall denote the standard input.
17574
17575 *Rationale:* Austin Group Defect Report(s) applied: 254.
17576 See <http://austingroupbugs.net/view.php?id=254>
17577
17578
17579 **Change Number: XCU/TC1/D5/0062 [254]**
17580
17581
17582 On Page: 2431 Line: 77143 Section: awk
17583
17584 In the STDIN section, change from:
17585
17586 The standard input shall be used only if no file operands are specified,

17587 or if a file operand is '-';
17588
17589 to:
17590
17591 The standard input shall be used only if no file operands are specified,
17592 or if a file operand is '-', or if a progfile option-argument is '-';
17593
17594 *Rationale:* Austin Group Defect Report(s) applied: 254.
17595 See <http://austingroupbugs.net/view.php?id=254>
17596
17597
17598 **Change Number: XCU/TC1/D5/0063** [224]
17599
17600
17601 On Page: 2440 Line: 77514 Section: awk
17602
17603 In the EXTENDED DESCRIPTION section, change from:
17604
17605 An extended regular expression can be used to separate fields by using
17606 the -F ERE option or by assigning a string containing the expression to
17607 the built-in variable FS.
17608
17609 to:
17610
17611 An extended regular expression can be used to separate fields by assigning
17612 a string containing the expression to the built-in variable FS, either
17613 directly or as a consequence of using the -F sepstring option.
17614
17615 *Rationale:* Austin Group Defect Report(s) applied: 224.
17616 See <http://austingroupbugs.net/view.php?id=224>
17617
17618
17619 **Change Number: XCU/TC1/D5/0064** [454]
17620
17621
17622 On Page: 2440 Line: 77519 Section: awk
17623
17624 In the EXTENDED DESCRIPTION section, Regular Expressions change from:
17625
17626 a. If FS is <space>, skip leading and trailing <blank>
17627 characters; fields shall be delimited by sets of one or
17628 more <blank> characters.
17629
17630 to:
17631
17632 a. If FS is <space>, skip leading and trailing <blank> and
17633 <newline> characters; fields shall be delimited by sets
17634 of one or more <blank> or <newline> characters.
17635
17636 *Rationale:* Austin Group Defect Report(s) applied: 454.
17637 See <http://austingroupbugs.net/view.php?id=454>
17638
17639
17640 **Change Number: XCU/TC1/D5/0065** [192,538]
17641
17642
17643 On Page: 2465 Line: 78649 Section: basename
17644
17645 In the EXAMPLES section, change from:
17646
17647 \$(basename "string")
17648
17649 to:

17650
17651 \$(basename -- "string")
17652
17653 On line 78652, change from:
17654
17655 \$(dirname "string")
17656
17657 to:
17658
17659 \$(dirname -- "string")
17660
17661 *Rationale:* Austin Group Defect Report(s) applied: 192,538.
17662 See <http://austingroupbugs.net/view.php?id=192>
17663 See <http://austingroupbugs.net/view.php?id=538>
17664
17665
17666 **Change Number: XCU/TC1/D5/0066** [192,430,538]
17667
17668
17669 On Page: 2465 Line: 78659 Section: basename
17670
17671 In the EXAMPLES section, change from:
17672
17673 c99 \$(dirname "\$1")/\$(basename "\$1" .c).c
17674 mv a.out \$(basename "\$1" .c)
17675
17676 to:
17677
17678 c99 -- "\$(dirname -- "\$1")/\$(basename -- "\$1" .c).c" &&
17679 mv a.out "\$(basename -- "\$1" .c)"
17680
17681 *Rationale:* Austin Group Defect Report(s) applied: 192,430,538.
17682 See <http://austingroupbugs.net/view.php?id=192>
17683 See <http://austingroupbugs.net/view.php?id=430>
17684 See <http://austingroupbugs.net/view.php?id=538>
17685
17686
17687 **Change Number: XCU/TC1/D5/0067** [192,538]
17688
17689
17690 On Page: 2466 Line: 78664 Section: basename
17691
17692 In the RATIONALE section, change from:
17693
17694 \$(basename "string")
17695
17696 to:
17697
17698 \$(basename -- "string")
17699
17700 On line 78666, change from:
17701
17702 \$(dirname "string")
17703
17704 to:
17705
17706 \$(dirname -- "string")
17707
17708 *Rationale:* Austin Group Defect Report(s) applied: 192,538.
17709 See <http://austingroupbugs.net/view.php?id=192>
17710 See <http://austingroupbugs.net/view.php?id=538>
17711
17712

Change Number: XCU/TC1/D5/0068 [129]

On Page: 2488 Line: 79554 Section: c99

In the OPTIONS section, change from:

The order of specifying the -I, -L, and -l options, and the order of specifying -l options with respect to pathname operands is significant.

to:

The order of specifying the -L and -l options, and the order of specifying -l options with respect to pathname operands is significant.

Rationale: Austin Group Defect Report(s) applied: 129.

See <http://austingroupbugs.net/view.php?id=129>

The exception to the utility syntax guidelines for -I is removed.

Change Number: XCU/TC1/D5/0069 [187]

On Page: 2489 Line: 79580 Section: c99

In the OPTIONS section, -I, change from:

Directories named in -I options shall be searched in the order specified.

to:

Directories named in -I options shall be searched in the order specified. If the -I option is used to specify a directory that is one of the usual places searched by default, the results are unspecified.

Rationale: Austin Group Defect Report(s) applied: 187.

See <http://austingroupbugs.net/view.php?id=187>

Change Number: XCU/TC1/D5/0070 [187]

On Page: 2489 Line: 79585 Section: c99

In the OPTIONS section, -L, change from:

Directories named in -L options shall be searched in the order specified.

to:

Directories named in -L options shall be searched in the order specified. If the -L option is used to specify a directory that is one of the usual places searched by default, the results are unspecified.

Rationale: Austin Group Defect Report(s) applied: 187.

See <http://austingroupbugs.net/view.php?id=187>

Change Number: XCU/TC1/D5/0071 [131]

On Page: 2490 Line: 79625 Section: c99

17776 In the INPUT FILES section, change from:
17777
17778 The input file shall be one of the following: ...
17779
17780 to
17781
17782 Each input file shall be one of the following: ...
17783
17784 *Rationale:* Austin Group Defect Report(s) applied: 131.
17785 See <http://austingroupbugs.net/view.php?id=131>
17786 An editorial clarification is made to the INPUT FILES section.
17787
17788
17789 **Change Number: XCU/TC1/D5/0072** [187]
17790
17791
17792 On Page: 2492 Line: 79718 Section: c99
17793
17794 In the EXTENDED DESCRIPTION section, after L79718 add a new unnumbered
17795 heading and a new paragraph:
17796
17797 Header Search
17798
17799 If a file with the same name as one of the standard headers defined in
17800 XBD Chapter 13, not provided as part of the implementation, is placed
17801 in any of the usual places that are searched by default for headers,
17802 the results are unspecified.
17803
17804 *Rationale:* Austin Group Defect Report(s) applied: 187.
17805 See <http://austingroupbugs.net/view.php?id=187>
17806
17807
17808 **Change Number: XCU/TC1/D5/0073** [364,430]
17809
17810
17811 On Page: 2497 Line: 79922,79928 Section: c99
17812
17813 Change both instances of '~' in the last paragraph of the RATIONALE
17814 section to '|'.
17815
17816 *Rationale:* Austin Group Defect Report(s) applied: 364,430.
17817 See <http://austingroupbugs.net/view.php?id=364>
17818 See <http://austingroupbugs.net/view.php?id=430>
17819
17820
17821 **Change Number: XCU/TC1/D5/0074** [56]
17822
17823
17824 On Page: 2499 Line: 79981 Section: cal
17825
17826 In the DESCRIPTION section, add a new paragraph to the DESCRIPTION
17827 after L79975:
17828
17829 If no operands are given, cal shall produce a one month calendar for the
17830 current month in the current year. If only the year operand is given,
17831 cal shall produce a calendar for all twelve months in the given calendar
17832 year. If both month and year operands are given, cal shall produce a
17833 one month calendar for the given month in the given year.
17834
17835 In the OPERANDS section, for the month operand delete the following text
17836 on L79981:
17837
17838 The default shall be the current month.

17839
17840 In the OPERANDS section, for the year operand delete the following text
17841 on L79983:
17842
17843 The default shall be the current year.
17844
17845 *Rationale:* Austin Group Defect Report(s) applied: 56.
17846 See <http://austingroupbugs.net/view.php?id=56>
17847
17848 **Change Number: XCU/TC1/D5/0075 [56]**
17849
17850
17851 On Page: 2500 Line: 80028 Section: cal
17852
17853 In the RATIONALE section, change from:
17854
17855 None.
17856
17857 to:
17858
17859 Earlier versions of this standard incorrectly required that the command:
17860
17861 cal 2000
17862
17863 write a one-month calendar for the current calendar month (no matter what
17864 the current year is) in the year 2000 to standard output. This did not
17865 match historic practice in any known version of the cal utility. The
17866 description has been updated to match historic practice. When only
17867 the year operand is given, cal writes a twelve-month calendar for the
17868 specified year.
17869
17870 *Rationale:* Austin Group Defect Report(s) applied: 56.
17871 See <http://austingroupbugs.net/view.php?id=56>
17872
17873
17874 **Change Number: XCU/TC1/D5/0076 [230]**
17875
17876
17877 On Page: 2505 Line: 80202 Section: cd
17878
17879 In the DESCRIPTION section, change from:
17880
17881 6. If the -P option is in effect, set curpath to the directory
17882 operand. Otherwise, set curpath to the string formed by the concatenation
17883 of the value of PWD, a <slash> character if the value of PWD did not
17884 end with a <slash> character, and the operand.:
17885
17886 to:
17887
17888 6. Set curpath to the directory operand.
17889
17890 *Rationale:* Austin Group Defect Report(s) applied: 230.
17891 See <http://austingroupbugs.net/view.php?id=230>
17892
17893
17894 **Change Number: XCU/TC1/D5/0077 [240]**
17895
17896
17897 On Page: 2506 Line: 80243-80245 Section: cd
17898
17899 In the DESCRIPTION section, change from:
17900
17901 If, during the execution of the above steps, the PWD environment variable

17902 is changed, the OLDPWD environment variable shall also be changed to
17903 the value of the old working directory (that is the current working
17904 directory immediately prior to the call to cd).
17905
17906 to:
17907
17908 If, during the execution of the above steps, the PWD environment variable
17909 is set, the OLDPWD environment variable shall also be set to the value
17910 of the old working directory (that is the current working directory
17911 immediately prior to the call to cd).
17912
17913 *Rationale:* Austin Group Defect Report(s) applied: 240.
17914 See <http://austingroupbugs.net/view.php?id=240>
17915
17916
17917 **Change Number: XCU/TC1/D5/0078 [240]**
17918
17919
17920 On Page: 2508 Line: 80323 Section: cd
17921
17922 In the EXAMPLES section, change from:
17923
17924 None.
17925
17926 to:
17927
17928 The following template can be used to perform processing in the directory
17929 specified by "location" and end up in the current working directory in
17930 use before the first cd command was issued:
17931
17932 cd location
17933 if [\$? -ne 0]
17934 then
17935 print error message
17936 exit 1
17937 fi
17938 ... do whatever is desired as long as the OLDPWD environment variable
17939 is not modified
17940 cd -
17941
17942 *Rationale:* Austin Group Defect Report(s) applied: 240.
17943 See <http://austingroupbugs.net/view.php?id=240>
17944
17945
17946 **Change Number: XCU/TC1/D5/0079 [123]**
17947
17948
17949 On Page: 2509 Line: 80360 Section: cd
17950
17951 In the CHANGE HISTORY section, change from:
17952
17953 Austin Group Interpretation 1003.1-2001 #199 is applied, clarifying how
17954 the od utility handles concatenation of two pathnames when the first
17955 pathname ends in a <slash> character.
17956
17957 to:
17958
17959 Austin Group Interpretation 1003.1-2001 #199 is applied, clarifying how
17960 the cd utility handles concatenation of two pathnames when the first
17961 pathname ends in a <slash> character.
17962
17963 *Rationale:* Austin Group Defect Report(s) applied: 123.
17964 See <http://austingroupbugs.net/view.php?id=123>

17965 An editorial correction to the description of Austin Group
17966 Interpretation 1003.1-2001 #199 is made in the CHANGE HISTORY section.
17967

17968
17969 **Change Number: XCU/TC1/D5/0080** [237,341]
17970

17971
17972 On Page: 2513 Line: 80511-80516,80526 Section: chgrp
17973

17974 In the OPTIONS section (-h), at L80511 delete the text:
17975

17976 If the system supports group IDs for symbolic links,
17977

17978 In the OPTIONS section (-h), at L80513 delete the text:
17979

17980 If the system does not support group IDs for symbolic links, for each
17981 file operand that names a file of type symbolic link, chgrp shall do
17982 nothing more with the current file and shall go on to any remaining files.
17983

17984 In the OPTIONS section (-P), at L80526 delete the text:
17985

17986 if the system supports this operation
17987

17988 *Rationale:* Austin Group Defect Report(s) applied: 237,341.
17989 See <http://austingroupbugs.net/view.php?id=237>

17990 See <http://austingroupbugs.net/view.php?id=341>
17991

17992
17993 **Change Number: XCU/TC1/D5/0081** [446]
17994

17995
17996 On Page: 2530 Line: 81169 Section: cksum
17997

17998 In the RATIONALE section, change from:
17999

```
18000 /* Input arguments:  
18001  * const char*    b == byte sequence to checksum  
18002  * size_t        n == length of sequence  
18003  */
```

```
18004  
18005     register unsigned    i, c, s = 0;  
18006  
18007     for (i = n; i > 0; --i) {  
18008         c = (unsigned) (*b++);  
18009         s = (s << 8) ^ crctab[(s >> 24) ^ c];  
18010     }
```

18011
18012 to:
18013

```
18014 /* Input arguments:  
18015  * const unsigned char*    b == byte sequence to checksum  
18016  * size_t                  n == length of sequence  
18017  */
```

```
18018  
18019     register size_t i;  
18020     register unsigned c, s = 0;  
18021  
18022     for (i = n; i > 0; --i) {  
18023         c = *b++;  
18024         s = (s << 8) ^ crctab[(s >> 24) ^ c];  
18025     }
```

18026
18027 *Rationale:* Austin Group Defect Report(s) applied: 446.

18028 See <http://austingroupbugs.net/view.php?id=446>
18029
18030
18031 **Change Number: XCU/TC1/D5/0082** [156]
18032
18033
18034 On Page: 2595 Line: 83575 Section: df
18035
18036 In the DESCRIPTION section, change from:
18037
18038 This output shall contain at least the file system names, amount of
18039 available space on each of these file systems, [XSI]and the number of
18040 free file slots ...
18041
18042 to:
18043
18044 This output shall contain at least the file system names, amount of
18045 available space on each of these file systems, [XSI]and, if no options
18046 other than -t are specified, the number of free file slots ...
18047
18048 *Rationale:* Austin Group Defect Report(s) applied: 156.
18049 See <http://austingroupbugs.net/view.php?id=156>
18050
18051
18052 **Change Number: XCU/TC1/D5/0083** [192,430]
18053
18054
18055 On Page: 2609 Line: 84135 Section: dirname
18056
18057 In the EXAMPLES section, add to the end of the section:
18058
18059 See also the examples for the basename utility.
18060
18061 *Rationale:* Austin Group Defect Report(s) applied: 192,430.
18062 See <http://austingroupbugs.net/view.php?id=192>
18063 See <http://austingroupbugs.net/view.php?id=430>
18064
18065
18066 **Change Number: XCU/TC1/D5/0084** [192]
18067
18068
18069 On Page: 2610 Line: 84142 Section: dirname
18070
18071 In the RATIONALE section, change from:
18072
18073 \$(basename "string")
18074
18075 to:
18076
18077 \$(basename -- "string")
18078
18079 *Rationale:* Austin Group Defect Report(s) applied: 192.
18080 See <http://austingroupbugs.net/view.php?id=192>
18081
18082
18083 **Change Number: XCU/TC1/D5/0085** [192]
18084
18085
18086 On Page: 2610 Line: 84144 Section: dirname
18087
18088 In the RATIONALE section, change from:
18089
18090 \$(dirname "string")

18091
18092 to:
18093
18094 \$(dirname -- "string")
18095
18096 *Rationale:* Austin Group Defect Report(s) applied: 192.
18097 See <http://austingroupbugs.net/view.php?id=192>
18098
18099
18100 **Change Number: XCU/TC1/D5/0086** [365]
18101
18102
18103 On Page: 2737 Line: 89056 Section: find
18104
18105 In the DESCRIPTION section, add at the end:
18106
18107 If a file is removed from or added to the directory hierarchy being
18108 searched it is unspecified whether or not find includes that file in
18109 its search.
18110
18111 *Rationale:* Austin Group Defect Report(s) applied: 365.
18112 See <http://austingroupbugs.net/view.php?id=365>
18113
18114
18115 **Change Number: XCU/TC1/D5/0087** [310]
18116
18117
18118 On Page: 2739 Line: 89159 Section: find
18119
18120 In the OPERANDS section, change from:
18121
18122 ... containing the two characters "{}" ...
18123
18124 to:
18125
18126 ... containing only the two characters "{}" ...
18127
18128 *Rationale:* Austin Group Defect Report(s) applied: 310.
18129 See <http://austingroupbugs.net/view.php?id=310>
18130
18131
18132 **Change Number: XCU/TC1/D5/0088** [309,310,430]
18133
18134
18135 On Page: 2740 Line: 89178 Section: find
18136
18137 In the OPERANDS section, change from:
18138
18139 If more than one argument containing only the two characters "{}" is
18140 present, the behavior is unspecified.
18141
18142 to:
18143
18144 If more than one argument containing the two characters "{}" is present,
18145 the behavior is unspecified.
18146
18147 In the OPERANDS section, delete the following text from L89180 and append
18148 it to L89164:
18149
18150 If a utility_name or argument string contains the two characters "{}",
18151 but not just the two characters "{}", it is implementation-defined whether
18152 find replaces those two characters or uses the string without change.
18153

18154 *Rationale:* Austin Group Defect Report(s) applied: 309,310,430.
18155 See <http://austingroupbugs.net/view.php?id=309>
18156 See <http://austingroupbugs.net/view.php?id=310>
18157 See <http://austingroupbugs.net/view.php?id=430>

18158
18159
18160 **Change Number: XCU/TC1/D5/0089** [235]
18161

18162
18163 On Page: 2743 Line: 89314-89317 Section: find
18164

18165 In the EXAMPLES section, change from:

18166
18167 mv /*.old /*.old ../old/
18168

18169 while avoiding an "Argument list too long" error if there are a large
18170 number of files ending with .old (and avoiding "No such file or directory"
18171 errors if no files match /*.old or /*.old).

18172
18173 to:

18174
18175 mv /*.old /*.old /*.old ../old/
18176

18177 while avoiding an "Argument list too long" error if there are a large
18178 number of files ending with .old and without running mv if there are no
18179 such files (and avoiding "No such file or directory" errors if ../old
18180 does not exist or no files match /*.old or /*.old).

18181
18182 *Rationale:* Austin Group Defect Report(s) applied: 235.
18183 See <http://austingroupbugs.net/view.php?id=235>
18184

18185
18186 **Change Number: XCU/TC1/D5/0090** [445]
18187

18188
18189 On Page: 2745 Line: 89400 Section: find
18190

18191 In the RATIONALE section, change from:

18192
18193 It should be noted that this is an incompatible change to the ISO/IEC
18194 9899: 1999 standard. For example, the following command prints all
18195 files with a '-' after their name if they are regular files,
18196 and a '+' otherwise:

18197
18198 to:

18199
18200 It should be noted that this is an incompatible change to the 1003.2-1992
18201 standard. For example, the following command printed all files with a
18202 '-' after their name if they are regular files, and a '+' otherwise:

18203
18204 *Rationale:* Austin Group Defect Report(s) applied: 445.
18205 See <http://austingroupbugs.net/view.php?id=445>
18206 This was an incorrect reference.
18207

18208
18209 **Change Number: XCU/TC1/D5/0091** [125]
18210

18211
18212 On Page: 2775 Line: 90502-90509 Section: getconf
18213

18214 In the EXAMPLES section, delete the text:

18215
18216 Note that:

`sysconf(_SC_2_C_BIND);`

and:

`system("getconf _POSIX2_C_BIND");`

in a C program could give different answers. The `sysconf()` call supplies a value that corresponds to the conditions when the program was either compiled or executed, depending on the implementation; the `system()` call to `getconf` always supplies a value corresponding to conditions when the program is executed.

Rationale: Austin Group Defect Report(s) applied: 125.

See <http://austingroupbugs.net/view.php?id=125>

This example is removed as it contradicts the description of `sysconf()`.

Change Number: XCU/TC1/D5/0092 [159]

On Page: 2778 Line: 90587 Section: `getopts`

In the DESCRIPTION section, change from:

When the end of options is encountered, the `getopts` utility shall exit with a return value greater than zero; the shell variable `OPTIND` shall be set to the index of the first non-option-argument, where the first `--` argument is considered to be an option-argument if there are no other non-option-arguments appearing before it, or the value `"$#+1"` if there are no non-option-arguments; the name variable shall be set to the `<question-mark>` character. Any of the following shall identify the end of options: the special option `--`, finding an argument that does not begin with a `'-'`, or encountering an error.

to:

When the end of options is encountered, the `getopts` utility shall exit with a return value greater than zero; the shell variable `OPTIND` shall be set to the index of the first operand, or the value `"$#+1"` if there are no operands; the name variable shall be set to the `<question-mark>` character. Any of the following shall identify the end of options: the first `--` argument that is not an option-argument, finding an argument that is not an option-argument and does not begin with a `'-'`, or encountering an error.

Rationale: Austin Group Defect Report(s) applied: 159.

See <http://austingroupbugs.net/view.php?id=159>

Change Number: XCU/TC1/D5/0093 [241]

On Page: 2788 Line: 90977 Section: `hash`

In the ENVIRONMENT VARIABLES section, add the XSI margin marker and shading to the `NLSPATH` description.

Rationale: Austin Group Defect Report(s) applied: 241.

See <http://austingroupbugs.net/view.php?id=241>

Change Number: XCU/TC1/D5/0094 [291]

On Page: 2796 Line: 91229 Section: iconv

In the RATIONALE section, add a new paragraph:

The iconv utility may support the conversion between ASCII and EBCDIC-based encodings, but is not required to do so. In an XSI-compliant implementation, the dd utility is the only method guaranteed to support conversion between these two character sets.

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

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

Change Number: XCU/TC1/D5/0095 [291]

On Page: 2796 Line: 91233 Section: iconv

In the SEE ALSO section, add a link to dd.

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

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

Change Number: XCU/TC1/D5/0096 [136]

On Page: 2839 Line: 92851 Section: ln

In the DESCRIPTION section (in item 2 under "For each source_file"), change from:

2. If the -s option is specified, ln shall create a symbolic link named by the destination path and containing as its pathname source_file.

to:

2. If the -s option is specified, actions shall be performed equivalent to the symlink() function with source_file as the path1 argument and the destination path as the path2 argument.

Rationale: Austin Group Defect Report(s) applied: 136.

See <http://austingroupbugs.net/view.php?id=136>

Updates to the -s option to reference the symlink() function.

Change Number: XCU/TC1/D5/0097 [291]

On Page: 2848 Line: 93178 Section: locale

In the RATIONALE section, add a new paragraph:

According to XBD Section 6.1, the standard requires that all supported locales must have the same encoding for <period> and <slash>, because these two characters are used within the locale-independent pathname resolution sequence. Therefore, it would be an error if 'locale -a' listed both ASCII and EBCDIC-based locales, since those two encodings do not share the same representation for either <period> or <slash>. Any system that supports both environments would be expected to provide two

18343 POSIX locales, one in either codeset, where only the locales appropriate
18344 to the current environment can be visible at a time. In an XSI-compliant
18345 implementation, the dd utility is the only portable means for performing
18346 conversions between the two charsets.

18347
18348 *Rationale:* Austin Group Defect Report(s) applied: 291.
18349 See <http://austingroupbugs.net/view.php?id=291>
18350

18351
18352 **Change Number: XCU/TC1/D5/0098** [424]
18353

18354
18355 On Page: 2864 Line: 93719 Section: ls
18356

18357 In the SYNOPSIS section, change from:
18358
18359 XSI ls [-ACFRSacdfiklmnpqrstux1] [-H|-L] <XSI>[-go]</XSI> [file...]
18360

18361 to:
18362 XSI ls [-ikqrs] [-<XSI>g</XSI>ln<XSI>o</XSI>] [-A|-a] [-C|-m|-x|-1] \
18363 [-F|-p] [-H|-L] [-R|-d] [-S|-f|-t] [-c|-u] [file...]
18364

18365 *Rationale:* Austin Group Defect Report(s) applied: 424.
18366 See <http://austingroupbugs.net/view.php?id=424>
18367

18368
18369 **Change Number: XCU/TC1/D5/0099** [424]
18370

18371
18372 On Page: 2864 Line: 93751 Section: ls
18373

18374 In the OPTIONS section, add to the end of the description of the
18375 -C option:
18376

18377 This option disables long format output.
18378
18379 *Rationale:* Austin Group Defect Report(s) applied: 424.
18380 See <http://austingroupbugs.net/view.php?id=424>
18381

18382
18383 **Change Number: XCU/TC1/D5/0100** [424]
18384

18385
18386 On Page: 2865 Line: 93768 Section: ls
18387

18388 In the OPTIONS section, add to the end of the description of the -R
18389 option:
18390

18391 The use of -R with -d or -f produces unspecified results.
18392

18393 *Rationale:* Austin Group Defect Report(s) applied: 424.
18394 See <http://austingroupbugs.net/view.php?id=424>
18395

18396
18397 **Change Number: XCU/TC1/D5/0101** [424]
18398

18399
18400 On Page: 2865 Line: 93778 Section: ls
18401

18402 In the OPTIONS section, -d option, change from:
18403

18404 with -R produces unspecified results
18405

18406 to:
18407
18408 with -R or -f produces unspecified results
18409
18410 *Rationale:* Austin Group Defect Report(s) applied: 424.
18411 See <http://austingroupbugs.net/view.php?id=424>
18412
18413
18414 **Change Number: XCU/TC1/D5/0102 [424]**
18415
18416
18417 On Page: 2865 Line: 93780-93781 Section: ls
18418
18419 In the OPTIONS section, -f option, change the last sentence from:
18420
18421 This option shall turn off -l, -t, -S, -s, and -r, and shall turn on -a.
18422
18423 to:
18424
18425 This option shall turn on -a. When -f is specified, any occurrences of
18426 the -r, -S, and -t options shall be ignored and any occurrences of the
18427 -A, <XSI>-g</XSI>, -l, -n, <XSI>-o</XSI> and -s options may be ignored.
18428 The use of -f with -R or -d produces unspecified results.
18429
18430 *Rationale:* Austin Group Defect Report(s) applied: 424.
18431 See <http://austingroupbugs.net/view.php?id=424>
18432
18433
18434 **Change Number: XCU/TC1/D5/0103 [424]**
18435
18436
18437 On Page: 2865 Line: 93782 Section: ls
18438
18439 In the OPTIONS section, -g option, change from:
18440
18441 The same as -l, except that the owner shall not be written.
18442
18443 to:
18444
18445 Turn on the -l (ell) option, but disable writing the file's owner name
18446 or number. Disable the -C, -m, and -x options.
18447
18448 [Note to editors: XSI shading retained on the entire description of
18449 this option.]
18450
18451 *Rationale*
18452
18453
18454 **Change Number: XCU/TC1/D5/0104 [424]**
18455
18456
18457 On Page: 2865 Line: 93788-93789 Section: ls
18458
18459 In the OPTIONS section, -l (ell) option, change from:
18460
18461 When -l (ell) is specified, -l (one) shall be assumed.
18462
18463 to:
18464
18465 Disable the -C, -m, and -x options.
18466
18467 *Rationale:* Austin Group Defect Report(s) applied: 424.
18468 See <http://austingroupbugs.net/view.php?id=424>

18469
18470
18471 **Change Number: XCU/TC1/D5/0105** [423,424]
18472
18473
18474 On Page: 2865 Line: 93790 Section: ls
18475
18476 In the OPTIONS section, -m, change from:
18477
18478 -m Stream output format; list files across the page, separated by
18479 <comma> characters.
18480
18481 to:
18482
18483 -m Stream output format; list pathnames across the page, separated
18484 by a <comma> character followed by a <space> character. Use
18485 a <newline> character as the list terminator and after the
18486 separator sequence when there is not room on a line for the next
18487 list entry. This option disables long format output.
18488
18489 *Rationale:* Austin Group Defect Report(s) applied: 423,424.
18490 See <http://austingroupbugs.net/view.php?id=423>
18491 See <http://austingroupbugs.net/view.php?id=424>
18492
18493
18494 **Change Number: XCU/TC1/D5/0106** [424]
18495
18496
18497 On Page: 2865 Line: 93791-93792 Section: ls
18498
18499 In the OPTIONS section, -n option, change from:
18500
18501 The same as -l, except that the owner's UID and GID numbers shall be
18502 written, rather than the associated character strings.
18503
18504 to:
18505
18506 Turn on the -l (ell) option, but when writing the file's owner or group,
18507 write the file's numeric UID or GID rather than the user or group name,
18508 respectively. Disable the -C, -m, and -x options.
18509
18510 *Rationale:* Austin Group Defect Report(s) applied: 424.
18511 See <http://austingroupbugs.net/view.php?id=424>
18512
18513
18514 **Change Number: XCU/TC1/D5/0107** [424]
18515
18516
18517 On Page: 2865 Line: 93793 Section: ls
18518
18519 In the OPTIONS section, -o option, change from:
18520
18521 The same as -l, except that the group shall not be written.
18522
18523 to:
18524
18525 Turn on the -l (ell) option, but disable writing the file's group name
18526 or number. Disable the -C, -m, and -x options.
18527
18528 [Note to editors: XSI shading retained on the entire description of
18529 this option.]
18530
18531 *Rationale:* Austin Group Defect Report(s) applied: 424.

See <http://austingroupbugs.net/view.php?id=424>

Change Number: XCU/TC1/D5/0108 [424]

On Page: 2866 Line: 93812 Section: ls

In the OPTIONS section, add to the end of the -l (one) option:

This option does not disable long format output. (Long format output is enabled by <XSI>-g</XSI>, -l (ell), -n, and <XSI>-o</XSI>; and disabled by -C, -m, and -x.)

Rationale: Austin Group Defect Report(s) applied: 424.

See <http://austingroupbugs.net/view.php?id=424>

Change Number: XCU/TC1/D5/0109 [424]

On Page: 2866 Line: 93811 Section: ls

In the OPTIONS section, add to the end of the -x option:

This option disables long format output.

Rationale: Austin Group Defect Report(s) applied: 424.

See <http://austingroupbugs.net/view.php?id=424>

Change Number: XCU/TC1/D5/0110 [424]

On Page: 2866 Line: 93813-93815 Section: ls

In the OPTIONS section, change from:

Specifying more than one of the options in the following mutually-exclusive pairs shall not be considered an error: -C and -l (ell), -m and -l (ell), -x and -l (ell), -C and -l (one), -H and -L, -c and -u, -t and -S. The last option specified in each pair shall determine the output format.

to:

If an option that enables long format output (<XSI>-g</XSI>, -l (ell), -n, and <XSI>-o</XSI>) is given with an option that disables long format output (-C, -m, and -x), this shall not be considered an error. The last of these options specified shall determine whether long format output is written.

If -R, -d, or -f are specified, the results of specifying these mutually-exclusive options are specified by the descriptions of these options above. If more than one of any of the other options shown in the SYNOPSIS section in mutually-exclusive sets are given, this shall not be considered an error; the last option specified in each set shall determine the output.

Note that if -t is specified, -c and -u are not only mutually-exclusive with each other, they are also mutually-exclusive with -S when determining sort order. But even if -S is specified after all occurrences of -c, -t, and -u, the last use of -c or

18595 -u determines the timestamp printed when producing long format
18596 output.
18597
18598 *Rationale:* Austin Group Defect Report(s) applied: 424.
18599 See <http://austingroupbugs.net/view.php?id=424>
18600
18601
18602 **Change Number: XCU/TC1/D5/0111 [423]**
18603
18604
18605 On Page: 2867 Line: 93858-93860 Section: ls
18606
18607 In the STDOUT section, change from:
18608
18609 When -m is specified, the format used shall be:
18610
18611 "%s, %s, ...\\n", <filename1>, <filename2>
18612
18613 where the largest number of filenames shall be written without exceeding
18614 the length of the line.
18615
18616 to:
18617
18618 When -m is specified, the format used for the last element of the list
18619 shall be:
18620
18621 "%s\\n", <filename>
18622
18623 The format used for each other element of the list shall be:
18624
18625 "%s,%s", <filename>, <separator>
18626
18627 where, if there is not room for the next element of the list to fit within
18628 the current line length, <separator> is a string containing an optional
18629 <space> character and a mandatory <newline> character; otherwise it is
18630 a single <space> character.
18631
18632 *Rationale:* Austin Group Defect Report(s) applied: 423.
18633 See <http://austingroupbugs.net/view.php?id=423>
18634
18635
18636 **Change Number: XCU/TC1/D5/0112 [117]**
18637
18638
18639 On Page: 2867 Line: 93864 Section: ls
18640
18641 In the STDOUT section, change from:
18642
18643 If the -l option is specified without -L, the following information
18644 shall be written:
18645
18646 "%s %u %s %s %u %s %s\\n", <file mode>, <number of links>,
18647 <owner name>, <group name>, <number of bytes in the file>,
18648 <date and time>, <pathname>
18649
18650 If the file is a symbolic link, this information shall be about the link
18651 itself and the <pathname> field shall be of the form:
18652
18653 "%s -> %s", <pathname of link>, <contents of link>
18654
18655 If both -l and -L are specified, the following information shall be
18656 written:
18657

18658 "%s %u %s %s %u %s %s\n", <file mode>, <number of links>,
18659 <owner name>, <group name>, <number of bytes in the file>,
18660 <date and time>, <pathname of link>

18661
18662 where all fields except <pathname of link> shall be for the file resolved
18663 from the symbolic link.

18664
18665 to:

18666
18667 If the -l option is specified, the following information shall be written
18668 for files other than character special and block special files:

18669
18670 "%s %u %s %s %u %s %s\n", <file mode>, <number of links>,
18671 <owner name>, <group name>, <size>, <date and time>,
18672 <pathname>

18673
18674 If the -l option is specified, the following information shall be written
18675 for character special and block special files:

18676
18677 "%s %u %s %s %s %s %s\n", <file mode>, <number of links>,
18678 <owner name>, <group name>, <device info>, <date and time>,
18679 <pathname>

18680
18681 In both cases if the file is a symbolic link and the -L option is also
18682 specified, this information shall be for the file resolved from the
18683 symbolic link except that the <pathname> field shall contain the pathname
18684 of the symbolic link itself. If the file is a symbolic link and the -L
18685 option is not specified, this information shall be about the link itself
18686 and the <pathname> field shall be of the form:

18687
18688 "%s -> %s", <pathname of link>, <contents of link>

18689
18690 *Rationale:* Austin Group Defect Report(s) applied: 117.
18691 See <http://austingroupbugs.net/view.php?id=117>

18692
18693
18694 **Change Number: XCU/TC1/D5/0113 [117]**

18695
18696
18697 On Page: 2867 Line: 93879 Section: ls

18698
18699 In the STDOUT section, add two new paragraphs (before the paragraph
18700 commencing "The <date and time> field..."):

18701
18702 The <size> field shall contain the value that would be returned for the
18703 file in the st_size field of struct stat (see XBD <sys/stat.h>). Note
18704 that for some file types this value is unspecified.

18705
18706 The <device info> field shall contain implementation-defined information
18707 associated with the device in question.

18708
18709 *Rationale:* Austin Group Defect Report(s) applied: 117.
18710 See <http://austingroupbugs.net/view.php?id=117>

18711
18712
18713 **Change Number: XCU/TC1/D5/0114 [117]**

18714
18715
18716 On Page: 2868 Line: 93892 Section: ls

18717
18718 In the STDOUT section, delete:

18719
18720 If the file is a character special or block special file, the size of the

file may be replaced with implementation-defined information associated with the device in question.

Rationale: Austin Group Defect Report(s) applied: 117.
See <http://austingroupbugs.net/view.php?id=117>

Change Number: XCU/TC1/D5/0115 [424]

On Page: 2871 Line: 94022-94027 Section: `ls`

In the RATIONALE section, change from:

Generally, this volume of POSIX.1-2008 is silent about what happens when options are given multiple times. In the cases of `-C`, `-l`, and `-1`, however, it does specify the results of these overlapping options. Since `ls` is one of the most aliased commands, it is important that the implementation perform intuitively. For example, if the alias were:

```
alias ls="ls -C"
```

and the user typed `ls -1` (one), single-text-column output should result, not an error.

to:

This volume of POSIX.1-2008 is frequently silent about what happens when mutually-exclusive options are specified. Except for `-R`, `-d`, and `-f`, the `ls` utility is required to accept multiple options from each mutually-exclusive option set without treating them as errors and to use the behavior specified by the last option given in each mutually-exclusive set. Since `ls` is one of the most aliased commands, it is important that the implementation perform intuitively. For example, if the alias were:

```
alias ls="ls -C"
```

and the user typed `ls -1` (one), single-text-column output should result, not an error.

The `-g`, `-l` (ell), `-n`, and `-o` options are not mutually-exclusive options. They all enable long format output. They work together to determine whether the file's owner is written (no if `-g` is present), file's group is written (no if `-o` is present), and if the file's group or owner is written whether it is written as the name (default) or a string representation of the UID or GID number (if `-n` is present). The `-C`, `-m`, `-x`, and `-1` (one) are mutually-exclusive options and the first three of these disable long format output. The `-1` (one) option does not directly change whether or not long format output is enabled, but by overriding `-C`, `-m`, and `-x`, it can re-enable long format output that had been disabled by one of these options.

Rationale: Austin Group Defect Report(s) applied: 424.
See <http://austingroupbugs.net/view.php?id=424>

Change Number: XCU/TC1/D5/0116 [424]

On Page: 2872 Line: 94057 Section: `ls`

In the FUTURE DIRECTIONS section, change from:

18784 None.
18785
18786 to:
18787
18788 Allowing -f to ignore the -A, -g, -l, -n, -o, and -s options may be
18789 removed in a future version.
18790
18791 *Rationale:* Austin Group Defect Report(s) applied: 424.
18792 See <http://austingroupbugs.net/view.php?id=424>
18793
18794
18795 **Change Number: XCU/TC1/D5/0117** [241]
18796
18797
18798 On Page: 2873 Line: 94125 Section: m4
18799
18800 In the ENVIRONMENT VARIABLES section, add the XSI margin marker and
18801 shading to the NLSPATH description.
18802
18803 *Rationale:* Austin Group Defect Report(s) applied: 241.
18804 See <http://austingroupbugs.net/view.php?id=241>
18805
18806
18807 **Change Number: XCU/TC1/D5/0118** [242,431]
18808
18809
18810 On Page: 2878 Line: 94316-94318 Section: m4
18811
18812 In the EXTENDED DESCRIPTION section (shift), change from:
18813
18814 The defining text for the shift macro shall be all of its arguments
18815 except for the first one.
18816
18817 to:
18818
18819 The defining text for the shift macro shall be a comma-separated list
18820 of its arguments except the first one. Each argument shall be quoted
18821 using the current quoting strings.
18822
18823 *Rationale:* Austin Group Defect Report(s) applied: 242,431.
18824 See <http://austingroupbugs.net/view.php?id=242>
18825 See <http://austingroupbugs.net/view.php?id=431>
18826
18827
18828 **Change Number: XCU/TC1/D5/0119** [242,431]
18829
18830
18831 On Page: 2878 Line: 94346 Section: m4
18832
18833 In the EXTENDED DESCRIPTION section (translit), change from:
18834
18835 The defining text of the translit macro shall be the first argument
18836 with every character that occurs in the second argument replaced with
18837 the corresponding character from the third argument.
18838
18839 to:
18840
18841 The defining text of the translit macro shall be the first argument
18842 with every character that occurs in the second argument replaced with
18843 the corresponding character from the third argument. If no replacement
18844 character is specified for some source character because the second
18845 argument is longer than the third argument, that character shall be
18846 deleted from the first argument in translit's defining text.

18847

18848 *Rationale:* Austin Group Defect Report(s) applied: 242,431.

18849 See <http://austingroupbugs.net/view.php?id=242>

18850 See <http://austingroupbugs.net/view.php?id=431>

18851

18852

18853 **Change Number: XCU/TC1/D5/0120** [325,430]

18854

18855

18856 On Page: 2878 Line: 94349 Section: m4

18857

18858 In the EXTENDED DESCRIPTION, subsection translit, add after L94349

18859 another sentence:

18860

18861 The behavior is unspecified if the same character appears more than once

18862 in the second argument.

18863

18864 *Rationale:* Austin Group Defect Report(s) applied: 325,430.

18865 See <http://austingroupbugs.net/view.php?id=325>

18866 See <http://austingroupbugs.net/view.php?id=430>

18867

18868

18869 **Change Number: XCU/TC1/D5/0121** [257]

18870

18871

18872 On Page: 2913 Line: 95706 Section: make

18873

18874 In the EXTENDED DESCRIPTION, Makefile Execution, change from:

18875

18876 ... except that the shell -e option shall also be in effect.

18877

18878 to:

18879

18880 ... except that if errors are not being ignored then the shell -e option

18881 shall also be in effect. If errors are being ignored for the command

18882 (as a result of the -i option, a '-' command prefix, or a .IGNORE special

18883 target), the shell -e option shall not be in effect.

18884

18885 *Rationale:* Austin Group Defect Report(s) applied: 257.

18886 See <http://austingroupbugs.net/view.php?id=257>

18887

18888

18889 **Change Number: XCU/TC1/D5/0122** [161]

18890

18891

18892 On Page: 2937 Line: 96641 Section: mkdir

18893

18894 In the OPTIONS section, change from:

18895

18896 -p Create any missing intermediate pathname components.

18897

18898 For each dir operand that does not name an existing directory, effects

18899 equivalent to those caused by the following command shall occur:

18900

18901 mkdir -p -m \$(umask -S),u+wx \$(dirname dir) &&

18902 mkdir [-m mode] dir

18903

18904 where the -m mode option represents that option supplied to the original

18905 invocation of mkdir, if any.

18906

18907 Each dir operand that names an existing directory shall be ignored

18908 without error.

18909

to:

`-p` Create any missing intermediate pathname components.

For each `dir` operand that does not name an existing directory, before performing the actions described in the DESCRIPTION above, the `mkdir` utility shall create any pathname components of the path prefix of `dir` that do not name an existing directory by performing actions equivalent to first calling the `mkdir()` function with the following arguments:

1. A pathname naming the missing pathname component, ending with a trailing `<slash>` character, as the path argument
2. The value zero as the mode argument

and then calling the `chmod()` function with the following arguments:

1. The same path argument as in the `mkdir()` call
2. The value `(S_IWUSR|S_IXUSR|~filemask)&0777` as the mode argument, where `filemask` is the file mode creation mask of the process (see `umask()`)

Each `dir` operand that names an existing directory shall be ignored without error.

Rationale: Austin Group Defect Report(s) applied: 161.
See <http://austingroupbugs.net/view.php?id=161>

Change Number: XCU/TC1/D5/0123 [265]

On Page: 2943 Line: 96840 Section: more

In the OPTIONS section, replace the description of the `-e` option on L96840-96845 with:

Exit immediately after writing the last line of the last file in the argument list; see the EXTENDED DESCRIPTION section.

Rationale: Austin Group Defect Report(s) applied: 265.
See <http://austingroupbugs.net/view.php?id=265>

Change Number: XCU/TC1/D5/0124 [48]

On Page: 2955 Line: 97288 Section: mv

In the DESCRIPTION, change from:

In this case, if `target_file` ends with a trailing ...

to:

In this case, if `source_file` names a non-directory file and `target_file` ends with a trailing ...

Rationale: Austin Group Defect Report(s) applied: 48.
See <http://austingroupbugs.net/view.php?id=48>

18973 **Change Number: XCU/TC1/D5/0125** [263]
18974
18975
18976 On Page: 2973 Line: 97946 Section: nm
18977
18978 In the OPTIONS section, at L97946 change from:
18979
18980 -v Sort output by value instead of alphabetically.
18981
18982 to:
18983
18984 -v Sort output by value instead of by symbol name.
18985
18986 *Rationale:* Austin Group Defect Report(s) applied: 263.
18987 See <http://austingroupbugs.net/view.php?id=263>
18988
18989
18990 **Change Number: XCU/TC1/D5/0126** [263]
18991
18992
18993 On Page: 2974 Line: 97979 Section: nm
18994
18995 In the STDOUT section, change from:
18996
18997 ... the output shall be sorted alphabetically by symbol name.
18998
18999 to:
19000
19001 ... the output shall be sorted by symbol name according to the collation
19002 sequence in the current locale.
19003
19004 *Rationale:* Austin Group Defect Report(s) applied: 263.
19005 See <http://austingroupbugs.net/view.php?id=263>
19006
19007
19008 **Change Number: XCU/TC1/D5/0127** [291]
19009
19010
19011 On Page: 3001 Line: 98983 Section: pathchk
19012
19013 In the DESCRIPTION section, change from:
19014
19015 Contains any character in any component that is not valid in its
19016 containing directory ...
19017
19018 to:
19019
19020 Contains any byte sequence that is not valid in its containing directory ...
19021
19022 *Rationale:* Austin Group Defect Report(s) applied: 291.
19023 See <http://austingroupbugs.net/view.php?id=291>
19024
19025
19026 **Change Number: XCU/TC1/D5/0128** [260]
19027
19028
19029 On Page: 3014 Line: 99542 Section: pax
19030
19031 In the OPTIONS section, (List Mode Format Specifications) change from:
19032
19033 ... exceptions 1. through 5. ...
19034
19035 to:

19036
19037 ... exceptions 1. through 6. ...
19038
19039 Renumber the list of additional exceptions that follows (to be 7-12
19040 instead of 6-11).
19041
19042 *Rationale:* Austin Group Defect Report(s) applied: 260.
19043 See <http://austingroupbugs.net/view.php?id=260>
19044
19045
19046 **Change Number: XCU/TC1/D5/0129** [261]
19047
19048
19049 On Page: 3015 Line: 99595 Section: pax
19050
19051 In the STDIN section, change from:
19052
19053 It shall be a text file containing a list of pathnames, one per line,
19054 without leading or trailing <blank> characters.
19055
19056 to:
19057
19058 It shall be a file containing a list of pathnames, each terminated by
19059 a <newline> character.
19060
19061 *Rationale:* Austin Group Defect Report(s) applied: 261.
19062 See <http://austingroupbugs.net/view.php?id=261>
19063
19064
19065 **Change Number: XCU/TC1/D5/0130** [261]
19066
19067
19068 On Page: 3017 Line: 99655 Section: pax
19069
19070 In the STDOUT section, (final paragraph), change from:
19071
19072 In list mode, standard output shall not be buffered more than a line at
19073 a time.
19074
19075 to:
19076
19077 In list mode, standard output shall not be buffered more than a pathname
19078 (plus any associated information and a <newline> terminator) at a time.
19079
19080 *Rationale:* Austin Group Defect Report(s) applied: 261.
19081 See <http://austingroupbugs.net/view.php?id=261>
19082
19083
19084 **Change Number: XCU/TC1/D5/0131** [313]
19085
19086
19087 On Page: 3031 Line: 100218 Section: pax
19088
19089 Add the following as a new paragraph to the APPLICATION USAGE section:
19090
19091 When all of the following are true:
19092
19093 1. A file of type directory is being placed into an archive.
19094 2. The ustar archive format is being used.
19095 3. The pathname of the directory is less than or equal to 155 bytes long
19096 (it will fit in the prefix field in the ustar header block).
19097 4. The last component of the pathname of the directory is longer than 100
19098 bytes long (it will not fit in the name field in the ustar header block).

19099
19100 some implementations of the pax utility will place the entire directory
19101 pathname in the prefix field, set the name field to an empty string,
19102 and place the directory in the archive. Other implementations of the pax
19103 utility will give an error under these conditions because the name field
19104 is not large enough to hold the last component of the directory name. This
19105 standard allows either behavior. However, when extracting a directory from
19106 a ustar format archive, this standard requires that all implementations
19107 be able to extract a directory even if the name field contains an empty
19108 string as long as the prefix field does not also contain an empty string.
19109

19110 *Rationale:* Austin Group Defect Report(s) applied: 313.
19111 See <http://austingroupbugs.net/view.php?id=313>
19112
19113

19114 **Change Number: XCU/TC1/D5/0132 [233]**
19115
19116

19117 On Page: 3032 Line: 100280 Section: pax
19118

19119 In the RATIONALE section, change from:
19120

19121 POSIX.1-2008 does not specify any way to create a hard link to a symbolic
19122 link, but many implementations provide this capability as an extension.
19123

19124 to:
19125

19126 Earlier versions of this standard did not specify any way to create a
19127 hard link to a symbolic link, but many implementations provided this
19128 capability as an extension.
19129

19130 *Rationale:* Austin Group Defect Report(s) applied: 233.
19131 See <http://austingroupbugs.net/view.php?id=233>
19132
19133

19134 **Change Number: XCU/TC1/D5/0133 [262]**
19135
19136

19137 On Page: 3153 Line: 104774-104781 Section: sed
19138

19139 In the OPTIONS section, change from:
19140

19141 -e script
19142

19143 Add the editing commands specified by the script option-argument to
19144 the end of the script of editing commands. The script option-argument
19145 shall have the same properties as the script operand, described in the
19146 OPERANDS section.
19147

19148 -f script_file
19149

19150 Add the editing commands in the file script_file to the end of
19151 the script.
19152

19153 -n
19154

19155 [...]
19156

19157 Multiple -e and -f options may be specified. All commands shall be added
19158 to the script in the order specified, regardless of their origin.
19159

19160 to:
19161

19162 -e script
19163

19164 Add the editing commands specified by the script option-argument to the
19165 end of the script of editing commands.
19166

19162
19163 -f script_file
19164 Add the editing commands in the file script_file to the end of the script
19165 of editing commands.
19166
19167 -n
19168 [...]
19169
19170 If any -e or -f options are specified, the script of editing commands
19171 shall initially be empty. The commands specified by each -e or -f
19172 option shall be added to the script in the order specified. When each
19173 addition is made, if the previous addition (if any) was from a -e option,
19174 a <newline> shall be inserted before the new addition. The resulting
19175 script shall have the same properties as the script operand, described
19176 in the OPERANDS section.
19177
19178 *Rationale:* Austin Group Defect Report(s) applied: 262.
19179 See <http://austingroupbugs.net/view.php?id=262>
19180
19181
19182 **Change Number: XCU/TC1/D5/0134** [282,431]
19183
19184
19185 On Page: 3154 Line: 104833 Section: sed
19186
19187 In the EXTENDED DESCRIPTION section, change from:
19188
19189 In default operation, sed cyclically shall append a line of input, less
19190 its terminating <newline>, into the pattern space. Normally the pattern
19191 space will be empty, unless a D command terminated the last cycle. The
19192 sed utility shall then apply in sequence all commands whose addresses
19193 select that pattern space, and at the end of the script copy the pattern
19194 space to standard output (except when -n is specified) and delete the
19195 pattern space.
19196
19197 to:
19198
19199 In default operation, sed cyclically shall append a line of input, less
19200 its terminating <newline> character, into the pattern space. Reading
19201 from input shall be skipped if a <newline> was in the pattern space
19202 prior to a D command ending the previous cycle. The sed utility shall
19203 then apply in sequence all commands whose addresses select that pattern
19204 space, until a command starts the next cycle or quits. If no commands
19205 explicitly started a new cycle, then at the end of the script the pattern
19206 space shall be copied to standard output (except when -n is specified)
19207 and the pattern space shall be deleted.
19208
19209 *Rationale:* Austin Group Defect Report(s) applied: 282,431.
19210 See <http://austingroupbugs.net/view.php?id=282>
19211 See <http://austingroupbugs.net/view.php?id=431>
19212
19213
19214 **Change Number: XCU/TC1/D5/0135** [269]
19215
19216
19217 On Page: 3156 Line: 104905 Section: sed
19218
19219 In the EXTENDED DESCRIPTION section (Editing commands in sed), change
19220 "function" to "editing command" throughout the description of the {
19221 command verb (L104905-104913).
19222
19223 *Rationale:* Austin Group Defect Report(s) applied: 269.
19224 See <http://austingroupbugs.net/view.php?id=269>

Change Number: XCU/TC1/D5/0136 [282,431]

On Page: 3156 Line: 104926 Section: sed

In the EXTENDED DESCRIPTION section (Editing commands in sed), change from:

[2addr]D Delete the initial segment of the pattern space through the first <newline> and start the next cycle.

to:

[2addr]D If the pattern space contains no <newline>, delete the pattern space and start a normal new cycle as if the d command was issued. Otherwise, delete the initial segment of the pattern space through the first <newline>, and start the next cycle with the resultant pattern space and without reading any new input.

Rationale: Austin Group Defect Report(s) applied: 282,431.

See <http://austingroupbugs.net/view.php?id=282>

See <http://austingroupbugs.net/view.php?id=431>

Change Number: XCU/TC1/D5/0137 [152]

On Page: 3165 Line: 105257 Section: sh

In the ENVIRONMENT VARIABLES section, add UP shading to the ENV description.

Rationale: Austin Group Defect Report(s) applied: 152.

See <http://austingroupbugs.net/view.php?id=152>

Change Number: XCU/TC1/D5/0138 [347]

On Page: 3167 Line: 105344 Section: sh

In the ASYNCHRONOUS EVENTS section, change from:

Default.

to:

The sh utility shall take the standard action for all signals (see Utility Description Defaults) with the following exceptions:

If the shell is interactive, SIGINT signals received during command line editing shall be handled as described in the EXTENDED DESCRIPTION, and SIGINT signals received at other times shall be caught but no action performed.

If the shell is interactive:

* SIGQUIT and SIGTERM signals shall be ignored.

* If the -m option is in effect, SIGTTIN, SIGTTOU, and SIGTSTP signals shall be ignored.

19288 * If the -m option is not in effect, it is unspecified whether
19289 SIGTIN, SIGTOU, and SIGTSTP signals are ignored, set to the
19290 default action, or caught. If they are caught the shell shall,
19291 in the signal-catching function, set the signal to the default
19292 action and raise the signal (after taking any appropriate steps
19293 such as restoring terminal settings).
19294
19295 The standard actions, and the actions described above for
19296 interactive shells, can be overridden by use of the trap special
19297 built-in utility (see trap and Section 2.11).
19298
19299 *Rationale:* Austin Group Defect Report(s) applied: 347.
19300 See <http://austingroupbugs.net/view.php?id=347>
19301
19302
19303 **Change Number: XCU/TC1/D5/0139 [347]**
19304
19305
19306 On Page: 3168 Line: 105387 Section: sh
19307
19308 In the EXTENDED DESCRIPTION, Command Line Editing (vi-mode),
19309 change from:
19310
19311 Typing the interrupt character in command mode shall cause sh to ...
19312
19313 to:
19314
19315 If sh receives a SIGINT signal in command mode (whether generated
19316 by typing the interrupt character or by other means), it shall ...
19317
19318 *Rationale:* Austin Group Defect Report(s) applied: 347.
19319 See <http://austingroupbugs.net/view.php?id=347>
19320
19321
19322 **Change Number: XCU/TC1/D5/0140 [347]**
19323
19324
19325 On Page: 3168 Line: 105406 Section: sh
19326
19327 In the EXTENDED DESCRIPTION, vi Line Editing Insert mode, change from:
19328
19329 Terminate command line editing ...
19330
19331 to:
19332
19333 If sh receives a SIGINT signal in insert mode (whether generated
19334 by typing the interrupt character or by other means), it shall
19335 terminate command line editing ...
19336
19337 *Rationale:* Austin Group Defect Report(s) applied: 347.
19338 See <http://austingroupbugs.net/view.php?id=347>
19339
19340
19341 **Change Number: XCU/TC1/D5/0141 [299]**
19342
19343
19344 On Page: 3175 Line: 105724 Section: sh
19345
19346 In the EXIT STATUS section, change from:
19347
19348 1-125 A non-interactive shell detected a syntax, redirection, or variable
19349 assignment error.
19350

19351 to:
19352
19353 1-125 A non-interactive shell detected an error other than `command_file`
19354 not found, including but not limited to syntax, redirection, or variable
19355 assignment errors.
19356
19357 *Rationale:* Austin Group Defect Report(s) applied: 299.
19358 See <http://austingroupbugs.net/view.php?id=299>
19359
19360
19361 **Change Number: XCU/TC1/D5/0142** [347]
19362
19363
19364 On Page: 3178 Line: 105872 Section: `sh`
19365
19366 In the RATIONALE section, add a new paragraph at the end of the
19367 section:
19368
19369 In interactive shells, SIGTERM is ignored so that `kill 0` does not kill
19370 the shell, and SIGINT is caught so that `wait` is interruptible. If the
19371 shell does not ignore SIGTIN, SIGTTOU, and SIGTSTP signals when it is
19372 interactive and the `-m` option is not in effect, these signals suspend
19373 the shell if it is not a session leader. If it is a session leader,
19374 the signals are discarded if they would stop the process, as required
19375 by XSH Section 2.4.3 for orphaned process groups.
19376
19377 *Rationale:* Austin Group Defect Report(s) applied: 347.
19378 See <http://austingroupbugs.net/view.php?id=347>
19379
19380
19381 **Change Number: XCU/TC1/D5/0143** [291]
19382
19383
19384 On Page: 3230 Line: 107745 Section: `test`
19385
19386 In the RATIONALE section, add a paragraph to the end of the section:
19387
19388 It is noted that '[' is not part of the portable filename character set;
19389 however, since it is required to be encoded by a single byte, and is
19390 part of the portable character set, the name of this utility forms a
19391 character string across all supported locales.
19392
19393 *Rationale:* Austin Group Defect Report(s) applied: 291.
19394 See <http://austingroupbugs.net/view.php?id=291>
19395
19396
19397 **Change Number: XCU/TC1/D5/0144** [266]
19398
19399
19400 On Page: 3233 Line: 107837 Section: `time`
19401
19402 In the STDERR section, add the following new sentence to the end of
19403 the paragraph:
19404
19405 The implementation may also prepend a single empty line before the format
19406 shown here.
19407
19408 *Rationale:* Austin Group Defect Report(s) applied: 266.
19409 See <http://austingroupbugs.net/view.php?id=266>
19410
19411
19412 **Change Number: XCU/TC1/D5/0145** [325]
19413

19414
19415 On Page: 3247 Line: 108366 Section: tr
19416
19417 In the EXTENDED DESCRIPTION section, change from:
19418
19419 Each input character found in the array specified by string1 shall be
19420 replaced by the character in the same relative position in the array
19421 specified by string2. When the array specified by string2 is shorter
19422 than the one specified by string1, the results are unspecified.
19423
19424 to:
19425
19426 If string2 is present, each input character found in the array specified
19427 by string1 shall be replaced by the character in the same relative
19428 position in the array specified by string2. If the array specified by
19429 string2 is shorter than the one specified by string1, or if a character
19430 occurs more than once in string1, the results are unspecified.
19431
19432 *Rationale:* Austin Group Defect Report(s) applied: 325.
19433 See <http://austingroupbugs.net/view.php?id=325>
19434
19435
19436 **Change Number: XCU/TC1/D5/0146** [241]
19437
19438
19439 On Page: 3254 Line: 108595 Section: tsort
19440
19441 In the ENVIRONMENT VARIABLES section, add the XSI margin marker and
19442 shading to the NLSPATH description.
19443
19444 *Rationale:* Austin Group Defect Report(s) applied: 241.
19445 See <http://austingroupbugs.net/view.php?id=241>
19446
19447
19448 **Change Number: XCU/TC1/D5/0147** [416]
19449
19450
19451 On Page: 3307 Line: 110388 Section: val
19452
19453 In the STDOUT section, change from:
19454
19455 If standard input is used, a line of input shall be written before
19456 each of the preceding lines for files containing discrepancies:
19457
19458 "%s:\n", <input line>
19459
19460 to:
19461
19462 If standard input is used, for each input line yielding a discrepancy,
19463 the output shall have the following format:
19464
19465 "%s\n\n %s: %s\n", <input>, <pathname>, <unspecified string>
19466
19467 where <input> is the input line minus its terminating <newline>.
19468
19469 *Rationale:* Austin Group Defect Report(s) applied: 416.
19470 See <http://austingroupbugs.net/view.php?id=416>
19471
19472
19473 **Change Number: XCU/TC1/D5/0148** [416]
19474
19475
19476 On Page: 3308 Line: 110423 Section: val

19477
19478 In the EXAMPLES section, remove the blank line between lines 110423
19479 and 110424.

19480
19481 *Rationale:* Austin Group Defect Report(s) applied: 416.
19482 See <http://austingroupbugs.net/view.php?id=416>

19483
19484
19485 **Change Number: XCU/TC1/D5/0149** [342]
19486

19487
19488 On Page: 3381 Line: 113157 Section: xargs
19489

19490 In the DESCRIPTION section, change from:

19491
19492 ('')

19493
19494 to:

19495
19496 ('\')

19497
19498 *Rationale:* Austin Group Defect Report(s) applied: 342.
19499 See <http://austingroupbugs.net/view.php?id=342>

19500 **4. Changes to Rationale**

19501 This section contains the set of changes to the text of the Rationale.

19502 [Note to reviewers: References to defect reports are provided to aid reviewers.]

19503 **Change Number: XRAT/TC1/D5/0001** [348]
19504

19505
19506 On Page: 3417 Line: 114446 Section: A.1.7.1
19507

19508 Change from:

19509
19510 MX This margin code is used to denote IEC 60559:1989 standard
19511 floating-point extensions.

19512
19513 to:

19514
19515 MX and MXX These two margin codes both relate to the IEC 60559
19516 Floating-Point option. The MX code denotes functionality that is mandated
19517 by the ISO C standard for IEC 60559 implementations; the MXX code denotes
19518 IEC 60559 functionality that is an extension to the ISO C standard.

19519
19520 *Rationale:* Austin Group Defect Report(s) applied: 348.
19521 See <http://austingroupbugs.net/view.php?id=348>

19522
19523
19524 **Change Number: XRAT/TC1/D5/0002** [291]
19525

19526
19527 On Page: 3424 Line: 114743 Section: A.3 Definitions
19528

19529 In Controlling Terminal, change "filename" to "pathname".

19530
19531 *Rationale:* Austin Group Defect Report(s) applied: 291.
19532 See <http://austingroupbugs.net/view.php?id=291>
19533

Change Number: XRAT/TC1/D5/0003 [291,420]

On Page: 3426 Line: 114796-114804 Section: A.3 Definitions

Change from:

Filename

At the present time, the primary responsibility for truncating filenames containing multi-byte characters must reside with the application. Some industry groups involved in internationalization believe that in the future the responsibility must reside with the kernel. For the moment, a clearer understanding of the implications of making the kernel responsible for truncation of multi-byte filenames is needed.

Character-level truncation was not adopted because there is no support in POSIX.1 that advises how the kernel distinguishes between single and multi-byte characters. Until that time, it must be incumbent upon application developers to determine where multi-byte characters must be truncated.

to:

Filename

Filenames are sequences of bytes, not sequences of characters. The only bytes that this standard says cannot appear in any filename are the slash byte and the null byte. This is a side effect of the fact that no conforming implementations of the standard currently provide a way to pass information specifying the locale associated with strings passed between user level applications and the kernel. This decision could be revisited if implementations develop a way to associate a locale with the strings passed between kernel space and user space.

Implementations may add other restrictions to the byte sequences allowed in filenames except that any filename consisting no more than {NAME_MAX} bytes from the set of characters in the portable filename character set must be allowed.

See XRAT Section A.3 Definitions, Pathname.

Rationale: Austin Group Defect Report(s) applied: 291,420.

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

See <http://austingroupbugs.net/view.php?id=420>

Change Number: XRAT/TC1/D5/0004 [291]

On Page: 3433 Line: 115093 Section: A.3 Definitions

Add a new section:

Pathname

Pathnames historically allowed all bytes except for the <slash> and <NUL> characters. For compatibility with existing file systems, this usage is maintained throughout the standard by noting that a pathname need not be a valid character string in all locales. However, the properties of the portable filename character set are such that a pathname using

19597 only those characters and the <slash> is portable in all locales as a
19598 character string.
19599
19600 *Rationale:* Austin Group Defect Report(s) applied: 291.
19601 See <http://austingroupbugs.net/view.php?id=291>
19602
19603
19604 **Change Number: XRAT/TC1/D5/0005** [291,441]
19605
19606
19607 On Page: 3437,3438 Line: 115268 Section: A.3 Definitions
19608
19609 In Symbolic Link on L115268, L115269, L115303, and L115312, change all
19610 instances of "filename" to "pathname".
19611
19612 *Rationale:* Austin Group Defect Report(s) applied: 291,441.
19613 See <http://austingroupbugs.net/view.php?id=291>
19614 See <http://austingroupbugs.net/view.php?id=441>
19615
19616
19617 **Change Number: XRAT/TC1/D5/0006** [291]
19618
19619
19620 On Page: 3445 Line: 115565 Section: A.4.6 Filenames
19621
19622 Change "filenames" to "filenames and pathnames".
19623
19624 *Rationale:* Austin Group Defect Report(s) applied: 291.
19625 See <http://austingroupbugs.net/view.php?id=291>
19626
19627
19628 **Change Number: XRAT/TC1/D5/0007** [291]
19629
19630
19631 On Page: 3445 Line: 115603 Section: A.4.6 Filenames
19632
19633 Change from:
19634
19635 The file system implementation historically deals only with bytes,
19636 not with characters, except for <slash> and the null byte.
19637
19638 to:
19639
19640 The file system implementation historically deals only with bytes,
19641 not with characters. Limitations on valid encodings ensure that the
19642 byte sequences for the <slash> character, <period> character, and <NUL>
19643 character will not be confused with any other character in any locale.
19644 However, there exist common single-shift encodings where other single-byte
19645 characters from the portable filename character set can also occur as a
19646 subset of a multi-byte character, making case-folding of portable filename
19647 bytes dependent on the context of whether a shift-state is active.
19648
19649 *Rationale:* Austin Group Defect Report(s) applied: 291.
19650 See <http://austingroupbugs.net/view.php?id=291>
19651
19652
19653 **Change Number: XRAT/TC1/D5/0008** [291]
19654
19655
19656 On Page: 3446 Line: 115615 Section: A.4.6 Filenames
19657
19658 Change from:
19659

19660 Case folding is inconsistent with portable filename character set
19661 definition and filename definition (all characters except <slash> and
19662 null). No known implementations allowing all characters except <slash>
19663 and null also do case folding.
19664
19665 to:
19666
19667 Case folding is inconsistent with the portable filename character set
19668 and filename definitions (all bytes except <slash> and null). No known
19669 implementations allowing all bytes except <slash> and null also do
19670 case folding.
19671
19672 *Rationale:* Austin Group Defect Report(s) applied: 291.
19673 See <http://austingroupbugs.net/view.php?id=291>
19674
19675
19676 **Change Number: XRAT/TC1/D5/0009** [291]
19677
19678
19679 On Page: 3450 Line: 115802 Section: A.4.12 Pathname Resolution
19680
19681 Add a new paragraph:
19682
19683 Earlier versions of this standard were unclear as to whether a pathname
19684 was required to be a character string or just a string. The <current
19685 version> is clear that filenames are just strings, and that pathname
19686 processing is locale-independent.
19687
19688 *Rationale:* Austin Group Defect Report(s) applied: 291.
19689 See <http://austingroupbugs.net/view.php?id=291>
19690
19691
19692 **Change Number: XRAT/TC1/D5/0010** [327,442]
19693
19694
19695 On Page: 3451 Line: 115844 Section: A.4.15 Seconds Since the Epoch
19696
19697 Change from:
19698
19699 The data size for `time_t` is as per the ISO C standard definition, which
19700 is implementation-defined.
19701
19702 to:
19703
19704 This standard requires that `time_t` be an integer type with
19705 implementation-defined size, but does not mandate a particular size. The
19706 requirement that `time_t` be integral is an additional constraint beyond
19707 the ISO C standard, which allows a real-floating `time_t`. Implementation
19708 practice has shown that much existing code is unprepared to deal with a
19709 floating-point `time_t`, and that use of struct `timespec` is a more uniform
19710 way to provide sub-second time manipulation within applications.
19711
19712 *Rationale:* Austin Group Defect Report(s) applied: 327,442.
19713 See <http://austingroupbugs.net/view.php?id=327>
19714 See <http://austingroupbugs.net/view.php?id=442>
19715 Although `time_t` is allowed to be a real-floating type by
19716 the ISO C Standard, all known implementations use an integral type and
19717 many applications assume that it is an integral type. Therefore, the C
19718 requirement is being strengthened in this standard.
19719
19720
19721 **Change Number: XRAT/TC1/D5/0011** [291]
19722

19723
19724 On Page: 3454 Line: 115933 Section: A.6.2 Character Encoding
19725
19726 Add a new paragraph:
19727
19728 The encoding for <slash> and <period> are required to be the same across
19729 all locales, in part because pathname resolution requires recognition
19730 of these bytes. It is a fortunate accident that all common shift-based
19731 encodings did not use either <slash> or <period> as a valid second byte
19732 in a multi-byte character.
19733
19734 *Rationale:* Austin Group Defect Report(s) applied: 291.
19735 See <http://austingroupbugs.net/view.php?id=291>
19736
19737
19738 **Change Number: XRAT/TC1/D5/0012 [22]**
19739
19740
19741 On Page: 3463 Line: 116354 Section: A.7.3.6 LC_MESSAGES
19742
19743 Add a new paragraph to the end of the section:
19744
19745 Affirmative responses like:
19746
19747 y
19748 Yes
19749 Yes!
19750
19751 and negative responses like:
19752
19753 N
19754 No
19755 Never
19756 No way!
19757
19758 should all be recognized as affirmative and negative responses,
19759 respectively, by the EREs identified by the yesexpr and noexpr keywords
19760 for English language-based locales. There is no requirement that
19761 multi-line responses nor ambiguous responses like:
19762
19763 no or yes
19764 yes or no
19765 maybe
19766
19767 be correctly classified by either of these EREs. Application writers
19768 are encouraged to include locale-specific suggestions for affirmative
19769 and negative responses in prompts.
19770
19771 *Rationale:* Austin Group Defect Report(s) applied: 22.
19772 See <http://austingroupbugs.net/view.php?id=22>
19773
19774
19775 **Change Number: XRAT/TC1/D5/0013 [302]**
19776
19777
19778 On Page: 3468 Line: 116554 Section: A.8.2 Internationalization Variables
19779
19780 Change from:
19781
19782 ... the locale of the process ...
19783
19784 to:
19785

19786 ... the global locale ...

19787

19788 *Rationale:* Austin Group Defect Report(s) applied: 302.

19789 See <http://austingroupbugs.net/view.php?id=302>

19790 These changes were overlooked during the revision when
19791 per-thread locales were added.

19792

19793

19794 **Change Number: XRAT/TC1/D5/0014** [305]

19795

19796

19797 On Page: 3474 Line: 116821 Section: A.9.3.5 RE Bracket Expressions

19798

19799 Add a new paragraph:

19800

19801 The standard specifies three possible behaviors for regular expressions
19802 such as "[[:alpha:]]". One behavior is the traditional implementation,
19803 which behaves like "[[:ahlp]]". Another, for alignment with the tr utility,
19804 is to treat it like "[[:alpha:]]". And finally, the standard allows
19805 rejecting the regular expression as invalid, as a means of alerting a
19806 user to the non-portable aspect of that regular expression. The set of
19807 regular expressions with this undefined behavior is limited solely to
19808 the expressions where the outer '[' and ']' of the bracket expression
19809 can be confused with the missing bracket pair '[' and ']' necessary to
19810 form a collating symbol, equivalence class, or character class; thus
19811 "[[:alpha:]]" or "[[::]]" do not trigger the unspecified behavior.

19812

19813 *Rationale:* Austin Group Defect Report(s) applied: 305.

19814 See <http://austingroupbugs.net/view.php?id=305>

19815

19816

19817 **Change Number: XRAT/TC1/D5/0015** [405]

19818

19819

19820 On Page: 3499 Line: 117653-117675 Section: B.2.2.1 POSIX.1 Symbols

19821

19822 Replace lines 117635-117675 with:

19823

19824 The `_POSIX_C_SOURCE` Feature Test Macro

19825

19826 The POSIX.1-1990 standard specified a macro called `_POSIX_SOURCE`. This has
19827 been superseded by `_POSIX_C_SOURCE`. This symbol will allow implementations
19828 to support various versions of this standard simultaneously.

19829 For instance, when `_POSIX_C_SOURCE` is defined as 200809L, the system
19830 should make visible the same name space as permitted and required by the
19831 POSIX.1-2008 standard. A special case is the one where the implementation
19832 wishes to make available support for the 1990 version of the POSIX
19833 standard, in which instance when either `_POSIX_SOURCE` is defined or
19834 `_POSIX_C_SOURCE` is defined as 1, the system should make visible the same
19835 name space as permitted and required by the POSIX.1-1990 standard.

19836

19837 It is expected that C bindings to future POSIX standards will define
19838 new values for `_POSIX_C_SOURCE`, with each new value reserving the name
19839 space for that new standard.

19840

19841 The `_XOPEN_SOURCE` Feature Test Macro

19842

19843 The feature test macro `_XOPEN_SOURCE` is provided as the announcement
19844 mechanism for the application that it requires functionality from the
19845 Single UNIX Specification. `_XOPEN_SOURCE` must be defined to the value
19846 700 before the inclusion of any header to enable the functionality in
19847 the Single UNIX Specification Version 4. Its definition subsumes the
19848 use of `_POSIX_C_SOURCE`.

An extract of code from a conforming application, that appears before any `#include` statements, is given below:

```
#define _XOPEN_SOURCE 700 /* Single UNIX Specification, Version 4 */  
#include ...
```

Note that the definition of `_XOPEN_SOURCE` with the value 700 makes the definition of `_POSIX_C_SOURCE` redundant and it can safely be omitted.

Rationale: Austin Group Defect Report(s) applied: 405.
See <http://austingroupbugs.net/view.php?id=405>

Change Number: XRAT/TC1/D5/0016 [76]

On Page: 3516 Line: 118452 Section: B.2.4.3

Change from:

IEEE Std 1003.1-2001/Cor 1-2002, item XSH/TC1/D6/6 is applied, adding the `abort()` function to the list of `async-cancel-safe` functions.

to:

IEEE Std 1003.1-2001/Cor 1-2002, item XSH/TC1/D6/6 is applied, adding the `abort()` function to the list of `async-signal-safe` functions.

Rationale: Austin Group Defect Report(s) applied: 76.
See <http://austingroupbugs.net/view.php?id=76>

Change Number: XRAT/TC1/D5/0017 [173]

On Page: 3517 Line: 118482 Section: B.2.5

Insert a new paragraph after L118482:

Although the C Standard guarantees that, at program start-up, `stdin` is open for reading and `stdout` and `stderr` are open for writing, this guarantee is contingent (as are all guarantees made by the C and POSIX standards) on the program being executed in a conforming environment. Programs executed with file descriptor 0 not open for reading or with file descriptor 1 or 2 not open for writing are executed in a non-conforming environment. Application writers are warned (in `exec()`, `posix_spawn()`, and Section C.2.7) not to execute a standard utility or a conforming application with file descriptor 0 not open for reading or with file descriptor 1 or 2 not open for writing.

Rationale: Austin Group Defect Report(s) applied: 173.
See <http://austingroupbugs.net/view.php?id=173>

Change Number: XRAT/TC1/D5/0018 [291]

On Page: 3539 Line: 119442 Section: B.2.8.3 Memory Management

In Section B.2.8.3, change "filenames" to "pathnames".

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

19912 See <http://austingroupbugs.net/view.php?id=291>
19913
19914
19915 **Change Number: XRAT/TC1/D5/0019 [381]**
19916
19917
19918 On Page: 3567 Line: 120753,120771 Section: Thread-Specific Data
19919
19920 Add to L120753:
19921
19922 This technique would also require that pthread_t not be an opaque type.
19923
19924 Add a new sentence to the end of L120771:
19925
19926 Therefore, the third model was chosen.
19927
19928 *Rationale:* Austin Group Defect Report(s) applied: 381.
19929 See <http://austingroupbugs.net/view.php?id=381>
19930
19931
19932 **Change Number: XRAT/TC1/D5/0020 [70]**
19933
19934
19935 On Page: 3576 Line: 121157 Section: B.2.9 Threads
19936
19937 In the section Read-Write Locks and Attributes, change from:
19938
19939 In cases where the default attributes are appropriate, the
19940 PTHREAD_RWLOCK_INITIALIZER macro can be used to initialize statically
19941 allocated read-write locks.
19942
19943 to:
19944
19945 In cases where the default attributes are appropriate, the
19946 PTHREAD_RWLOCK_INITIALIZER macro can be used to initialize read-write
19947 locks.
19948
19949 *Rationale:* Austin Group Defect Report(s) applied: 70.
19950 See <http://austingroupbugs.net/view.php?id=70>
19951
19952
19953 **Change Number: XRAT/TC1/D5/0021 [291]**
19954
19955
19956 On Page: 3637 Line: 123577 Section: C.1.1
19957
19958 In section C.1.1 Change History, change "fixed path filenames" to
19959 "fixed pathnames".
19960
19961 *Rationale:* Austin Group Defect Report(s) applied: 291.
19962 See <http://austingroupbugs.net/view.php?id=291>
19963
19964
19965 **Change Number: XRAT/TC1/D5/0022 [124]**
19966
19967
19968 On Page: 3655 Line: 124299 Section: C.2.6.1 Tilde Expansion
19969
19970 Add after L124299:
19971
19972 System administrators on systems where // has an implementation-defined
19973 meaning which is different to /, should not create users with a home
19974 directory of / or //, since this may lead to unexpected filename

19975 resolution on those systems.

19976

19977 *Rationale:* Austin Group Defect Report(s) applied: 124.

19978 See <http://austingroupbugs.net/view.php?id=124>

19979

19980

19981 **Change Number: XRAT/TC1/D5/0023** [222]

19982

19983

19984 On Page: 3656 Line: 124322 Section: C.2.6.2 Parameter Expansion

19985

19986 After L124322 add a new paragraph:

19987

19988 For rationale regarding expansion of "\${...}" within double-quotes, see

19989 Section C.2.2.3.

19990

19991 *Rationale:* Austin Group Defect Report(s) applied: 222.

19992 See <http://austingroupbugs.net/view.php?id=222>

19993

19994

19995 **Change Number: XRAT/TC1/D5/0024** [217]

19996

19997

19998 On Page: 3657 Line: 124359 Section: C.2.6.3 Command Substitution

19999

20000 Change from:

20001

20002 The requirement to separate "\$(" and '(' when a single subshell is

20003 command-substituted is to avoid any ambiguities with arithmetic expansion.

20004

20005 to:

20006

20007 Arithmetic expansions have precedence over command substitutions.

20008 That is, if the shell can parse an expansion beginning with "\$((" as an

20009 arithmetic expansion then it will do so. It will only parse the expansion

20010 as a command substitution (that starts with a subshell) if it determines

20011 that it cannot parse the expansion as an arithmetic expansion. If the

20012 syntax is valid for neither type of expansion, then it is unspecified

20013 what kind of syntax error the shell reports.

20014

20015 How well the shell performs this determination is a quality of

20016 implementation issue. Current shell implementations use heuristics. In

20017 particular, the shell need not evaluate nested expansions when determining

20018 whether it can parse an expansion beginning with "\$((" as an arithmetic

20019 expansion. For example:

20020

20021 `$((a $op b))`

20022

20023 is always an arithmetic expansion if "\$op" expands to, say, '+', but if

20024 "\$op" expands to '(' then the shell might still parse the expansion as

20025 an arithmetic expansion (resulting in a syntax error due to unbalanced

20026 parentheses) or it might perform a command substitution.

20027

20028 This standard requires that conforming applications always separate the

20029 "\$(" and '(' with white space when a command substitution starts with

20030 a subshell. This is because implementations may support extensions in

20031 arithmetic expressions which could result in the shell parsing the input

20032 as an arithmetic expansion even though a minimally conforming shell would

20033 not. For example, many shells support arrays with the array index (which

20034 can be an expression) in square brackets. Therefore, the presence of

20035 "myfile[0-9]" within an expansion beginning "\$((" is no guarantee that

20036 it will be parsed as a command substitution.

20037

The ambiguity is not restricted to the simple case of a single subshell. More complicated ambiguous cases are possible (even with just the standard shell syntax), such as:

```
$( ( cat <<EOH
+ ( (
EOH
) && ( cat <<EOH
) ) + 1 +
EOH
))
```

This can be parsed as an arithmetic expansion, with cat and EOH as the names of shell variables. Ambiguous cases also exist where the end of the expansion is at a different location for the arithmetic expansion and the command substitution:

```
$( (cat <<EOF
+(((
EOF
) && (
cat <<EOF
+
EOF
))
```

This is an incomplete arithmetic expansion, but would have been a (complete) command substitution if it could not have been parsed as an arithmetic expansion. If this expansion occurs at the end of input then the shell reports a syntax error; it does not parse it as a command substitution.

Rationale: Austin Group Defect Report(s) applied: 217.
See <http://austingroupbugs.net/view.php?id=217>

Change Number: XRAT/TC1/D5/0025 [173,441]

On Page: 3660 Line: 124527 Section: C.2.7

After L124527, add a new paragraph:

Applications should not use the [n]<&- or [n]>&- operators to execute a utility or application with file descriptor 0 not open for reading or with file descriptor 1 or 2 not open for writing, as this might cause the executed program (or shell builtin) to misbehave. In order not to pass on these file descriptors to an executed utility or application, applications should not just close them but should reopen them on, for example, /dev/null. Some implementations may reopen them automatically, but applications should not rely on this being done.

Rationale: Austin Group Defect Report(s) applied: 173,441.
See <http://austingroupbugs.net/view.php?id=173>
See <http://austingroupbugs.net/view.php?id=441>

Change Number: XRAT/TC1/D5/0026 [217]

On Page: 3667 Line: 124772 Section: C.2.9.4 Compound Commands

Insert a new paragraph at the end of the Grouping Commands text:

20101
20102 The requirement for conforming applications to separate two leading '('
20103 characters with white space if a grouping command would be parsed as
20104 an arithmetic expansion if preceded by a '\$' is to allow shells which
20105 implement the "((arithmetic expression))" extension to apply the
20106 same disambiguation rules consistently to \$((...)) and ((...)). See
20107 Section C.2.6.3.
20108
20109 *Rationale:* Austin Group Defect Report(s) applied: 217.
20110 See <http://austingroupbugs.net/view.php?id=217>
20111
20112
20113 **Change Number: XRAT/TC1/D5/0027 [291]**
20114
20115
20116 On Page: 3690 Line: 125577 Section: D.2.2 Process Management
20117
20118 In Section D.2.2, change "filename" to "pathname".
20119
20120 *Rationale:* Austin Group Defect Report(s) applied: 291.
20121 See <http://austingroupbugs.net/view.php?id=291>
20122
20123
20124 **Change Number: Frontmatter/TC1/D5/0003 [23]**
20125
20126 On Page: 3825 Line: none Section: Index
20127
20128 Remove the index entry for YESSTR.
20129
20130 *Rationale:* Austin Group Defect Report(s) applied: 23.
20131 See <http://austingroupbugs.net/view.php?id=23>
20132