INTERNATIONAL STANDARD

**ISO/IEC/IEEE 9945**

First edition
2009-09-15

Corrigendum 2
2017-03

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

## TECHNICAL CORRIGENDUM 2

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

*RECTIFICATIF TECHNIQUE 2*



Reference number
ISO/IEC/IEEE 9945:2009/Cor.2:2017(E)



ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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

stds.ipr@ieee.org
www.ieee.org

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, 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 ISO or the IEEE Standards Association.

ISO/IEC/IEEE 9945 was prepared by The Open Group (as The Open Group Technical Standard Base Specifications, Issue 7) and the Portable Applications Standards Committee of the Computer Society of the IEEE (as IEEE Std 1003.1™-2008). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages*, *their environments and system software interfaces*, in parallel with its approval by the ISO/IEC national bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. IEEE is responsible for the maintenance of this document with participation and input from ISO/IEC national bodies.



 

**IEEE Std 1003.1™-2008/Cor 2-2016**
(Corrigendum to
IEEE Std 1003.1-2008)

The Open Group Technical Standard
Base Specifications, Issue 7

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

# Base Specifications, Issue 7—Technical Corrigendum 2

Sponsor

**Portable Applications Standards Committee**
of the
**IEEE Computer Society**

and
**The Open Group**

Approved 30 June 2016

**IEEE-SA Standards Board**

Approved 20 June 2016

**The Open Group**

IEEE Std 1003.1™-2008/Cor 2-2016
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 2

**Abstract:** Problems discovered since the approval of IEEE Std 1003.1-2008 and IEEE Std 1003.1-2008/Cor 1-2013 are addressed.

**Keywords:** API, application program interface, argument, asynchronous, basic regular expression, batch job, batch system, BRE, built-in utility, byte, child, command language interpreter, CPU, ERE, extended regular expression, 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



Published 15 August 2016 by The Open Group.



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

PDF: ISBN 978-1-5044-2148-5 STD1005

**The Open Group** is a global consortium that enables the achievement of business objectives through IT standards. With more than 500 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, 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.

## Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

## Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. 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.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

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.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.



## 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 or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or 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 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. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

## 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

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

## Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, 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.

## Updating of IEEE Standards 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.

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 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 or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org\develop\index.html.

## Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website 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 2-2016 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 draft standard was completed, the Austin Group had the following membership:

**Andrew Josey**, *Chair*

**Donald W. Cragun**, *Organizational Representative, IEEE PASC*
**Nicholas M. Stoughton**, *Organizational Representative, ISO/IEC JTC 1/SC22*
**Roger Faulkner**, *Organizational Representative, The Open Group*

**Cathy Fox**, *Technical Editor*

## Austin Group Technical Reviewers

Bogdan Barbu
Eric Blake
Harvey Block
Mark S. Brown
Milan Cermak
Stephane Chazelas
Alexander Cherepanov
Mark Clancy
Geoff Clare
David Clissold
Alan Coopersmith
Donald W. Cragun
Matthew Dempsky
Dan Douglas
Lawrence D.K.B. Dwyer
Paul Eggert
Steve Emmerson
David Egan Evans
Roger Faulkner
Richard Felker
Thorsten Glaser
Glenn D. Golden
Jim Grisanzio
Philip Guenther
Richard Hansen
Mark Harris
Cyril Hrubis
Jarmo Jaakkola
Felix Janda
Aurelio Jargas
Andrew Josey
John Kearney
Michael Kerrisk
Rob King
Andi Kleen
Bruce Korb
Kaz Kylheku
Antoine Leca
Wojtek Lerch
Sven Mascheck
Margot Hackett Miller
Joseph S. Myers
Szabolcs Nagy
Alexander Nasonov
Jonathan Nieder
Danny Niu
Gian Ntzik
Steffen Nurpmeso
Carlos O'Donell
Vladimir Támara Patiño
Peter Petrov
William Pitcock
Jim Pryor
James C. Pugsley
Yury Pukhalsky
Steve Rago
Chet Ramey
Martin Řehák
Torvald Riegel
Xavier Roche
Askar Safin
Anton Salikhmetov
Jörg Schilling
Ed Schouten
Konrad Schwarz
Jens Schweikhardt
Mike Scudder
Martin Sebor
Nicholas M. Stoughton
Keith Thompson
Jilles Tjoelker
William Toth
Daniel Trebbien
Miloslav Trmac
Fred J. Tydeman
Ted Unangst
Brian Utterback
Stijn van Drongelen
Jonathan Wakely
Nathan Weeks
Florian Weimer
David A. Wheeler
Michael Wilson
André Zepezauer
Mark Ziegast

## Austin Group Working Group Members

Eitan Adler
Iwan Aucamp
Bogdan Barbu
David Bartley
Steve Bartolomei
Fabrice Bauzac
Guido Berhoerster
Eric Blake
Harvey Block
Hans Boehm
Bill Bogstad
Pádraig Brady
Davide Brini
Andries E. Brouwer
Mark S. Brown
David Butenhof
Milan Cermak
Stephane Chazelas
Alexander Cherepanov
Mark Clancy
Geoff Clare
David Clissold
Alan Coopersmith
Donald W. Cragun
Martijn Dekker
Matthew Dempsky
Thomas E. Dickey
Casper Dik
Dan Douglas
Niall Douglas
Ulrich Drepper
Lawrence D.K.B. Dwyer
Paul Eggert
Daniel Eischen
Robert Elz
Steve Emmerson
Laszlo Ersek
Bruce Evans
David Egan Evans
Roger Faulkner
Richard Felker
Jeffrey K. Fellin
Hal Finkel
Glenn Fowler
Cathy Fox
Mike Frysinger
Andrea Giugliano
Thorsten Glaser
Glenn D. Golden
Jim Grisanzio
Philip Guenther
Joseph M. Gwinn
Bruno Haible
Richard Hansen
Mark Harris
Barry E. Hedquist
David Holland
Tom Honermann
Cyril Hrubis
Jarmo Jaakkola
Felix Janda
Aurelio Jargas
Ross Johnson
Andrew Josey
John Kearney
Dan Kegel
Michael Kerrisk
Rob King
Tomas Klacko
Andi Kleen
Bruce Korb
David Korn
Kaz Kylheku
Rob Landley
Antoine Leca
Vincent Lefèvre
Sean Leonard
Wojtek Lerch
Yonggang Luo
Scott Lurndal
Roger Marquis
Sven Mascheck
Per Mildner
Margot Hackett Miller
Joseph S. Myers
Szabolcs Nagy
Alexander Nasonov
Jonathan Nieder
Danny Niu
Gian Ntzik
Steffen Nurpmeso
Carlos O'Donell
Isabella Parakiss
Vladimir Támara Patiño
Phil Pennock
Andres Perera
Peter Petrov
William Pitcock
Wayne Pollock
Jim Pryor
James C. Pugsley
Yury Pukhalsky
Steve Rago
Chet Ramey
Martin Řehák
Tom Ridge
Torvald Riegel
Xavier Roche
Askar Safin
Anton Salikhmetov
Jörg Schilling
Ed Schouten
Konrad Schwarz
Ingo Schwarze
Jens Schweikhardt
Mike Scudder
Martin Sebor
Glen Seeds
Jeffrey Sheinberg
Thor Lancelot Simon
Keld Simonsen
Ranjit Singh
Paul Smith
Steven Stewart-Gallus
Nicholas M. Stoughton
Alfred M. Szmidt
Marcel Telka
Donn Terry
Keith Thompson
Jilles Tjoelker
William Toth
Daniel Trebbien
Miloslav Trmac
Fred J. Tydeman
Ted Unangst
Brian Utterback
Stijn van Drongelen
Christopher Vance
Jonathan Wakely
Nathan Weeks
Florian Weimer
David A. Wheeler
Mats D. Wichmann
Michael Wilson
Garrett Wollman
Jörg Wunsch
James Youngman
André Zepezauer
Mark Ziegast

## The Open Group

When The Open Group approved the Base Specifications, Issue 7, Technical Corrigendum 2 on 20 June 2016, the membership of The Open Group Base Working Group was as follows:

**Andrew Josey**, *Chair*
**Roger Faulkner**, *Austin Group Liaison*

**Cathy Fox**, *Technical Editor*

## Base Working Group Members

Eric Blake
Geoff Clare
David Clissold
Donald W. Cragun
Roger Faulkner
Darrin Johnson
Andrew Josey
Martin Řehák
S. R. Srinivasan

## IEEE

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Portable Applications Standards Committee had the following membership:

**Joseph M. Gwinn**, *Chair*
**Andrew Josey**, *Functional Chair (Interpretations)*
**Donald W. Cragun**, *Shell and Utilities Working Group Chair*
**Barry Hedquist**, *Test Methods Working Group Chair*
**Craig Meyers**, *Distributed Services Working Group Chair*
**Stephen Walli**, *US TAG Institutional Representative*
**Roger Martin**, *Ex-officio Emeritus*
**Nicholas M. Stoughton**, *Secretary*

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

Johann Amsenga
Juan Carreon
Keith Chow
Donald W. Cragun
Sourav Dutta
Andrew Fieldsend
David Fuschi
Randall Groves
Joseph M. Gwinn
Barry E. Hedquist
Werner Hoelzl
Noriyuki Ikeuchi
Andrew Josey
Piotr Karocki
Kenneth Lang
Vincent Lefèvre
Peter Petrov
Stephen Schwarm
Walter Struppler
Mark-Rene Uchida
Oren Yuen

IEEE Std 1003.1™-2008/Cor 2-2016
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 2

When the IEEE-SA Standards Board approved this standard on 30 June 2016, it had the following membership:

**Jean-Philippe Faure,** *Chair*
**Ted Burse,** *Vice Chair*
**John D. Kulick,** *Past Chair*
**Konstantinos Karachalios,** *Secretary*

Chuck Adams
Masayuki Ariyoshi
Stephen Dukes
Jianbin Fan
J. Travis Griffith
Ronald W. Hotchkiss
Gary Hoffman
Michael Janezic
Joseph L. Koepfinger*
Hung Ling
Kevin Lu
Annette D. Reilly
Gary Robinson
Mehmet Ulema
Yingli Wen
Howard Wolfman
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

## Introduction

This introduction is not part of IEEE Std 1003.1-2008/Cor 2, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)/Base Specifications, Issue 7—Technical Corrigendum 2 .

This Technical Corrigendum addresses issues raised in defect reports and interpretation requests submitted up to 24 August 2015 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 the Front Matter ..... 2

2. Changes to Base Definitions..... 3

3. Changes to System Interfaces.....46

4. Changes to Shell and Utilities .....198

5. Changes to Rationale.....283

IEEE Std 1003.1™-2008/Cor 2-2016
IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®)
Base Specifications, Issue 7—Technical Corrigendum 2

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

# Base Specifications, Issue 7—Technical Corrigendum 2



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 the Front Matter

This section contains the set of changes to the text of the front matter.

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

**Change Number: Front Matter/TC2/D4/0001** [885]

On Page: xlv Line: NA Section: Referenced Documents
(2013 edition Page: xlvii Line: NA)

In the line below the **Source Documents** header, change from:

base documents for POSIX.1-2008

to:

base documents for POSIX.1-2001

*Rationale*: Austin Group Defect Report(s) applied: 885. See http://austingroupbugs.net/view.php?id=885.

**Change Number: Front Matter/TC2/D4/0002** [726]

On Page: xxxviii Line: NA Section: Acknowledgements

Remove the period from the first bullet list item.

*Rationale*: Austin Group Defect Report(s) applied: 726. See http://austingroupbugs.net/view.php?id=726.
Editorial change.

## 2. 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: XBD/TC2/D4/0001** [591]

On Page: 9 Line: 266 Section: 1.7.1 Codes

Change from:

```
[OH]#include <sys/types.h>[/OH]
#include <grp.h>
struct group *getgrnam(const char *name);
```

The OH margin legend indicates that the marked header is not required on XSI-conformant systems.

to:

```
[OH]#include <sys/stat.h>[/OH]
#include <fcntl.h>
int open(const char *path, int oflag, ...);
```

The OH margin legend indicates that the optional header defines constants that will be needed if the function is called with certain flag arguments; thus it may be required for some of the functionality described, but is not needed otherwise.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XBD/TC2/D4/0002** [810]

On Page: 16 Line: 493 Section: 2.1.1 Requirements

At the end of section 2.1.1, add a small-font note:

**Note:** If the documented method of setting up a conforming environment includes the need to set one or more environment variables, then the values of those environment variables cannot include any <space> characters, since the *confstr*() function must be able to return them in a <space>-separated list of variable=value pairs. See [xref to XSH confstr()].

*Rationale*: Austin Group Defect Report(s) applied: 810. See http://austingroupbugs.net/view.php?id=810.

**Change Number: XBD/TC2/D4/0003** [637]

On Page: 17 Line: 561 Section: 2.1.3 POSIX Conformance

Change from:

```
    _POSIX_TIMERS

-- Symbolic constants defined with a value greater than zero:
```

```
    _POSIX_JOB_CONTROL
    _POSIX_SAVED_IDS
```

to:

```
    _POSIX_TIMERS
    _POSIX2_C_BIND

-- Symbolic constants defined with a value greater than zero:
    _POSIX_JOB_CONTROL
    _POSIX_REGEXP
    _POSIX_SAVED_IDS
    _POSIX_SHELL
```

*Rationale*: Austin Group Defect Report(s) applied: 637. See http://austingroupbugs.net/view.php?id=637.

**Change Number: XBD/TC2/D4/0004** [937]

On Page: 35 Line: 1195 Section: 3.17 Application
(2013 edition Page: 35 Line: 1199)

After:

A computer program that performs some desired function.

add a new paragraph:

When the User Portability Utilities option is supported, requirements placed on applications relating to the use of standard utilities shall also apply to the actions of a user who is entering shell command language statements into an interactive shell.

*Rationale*: Austin Group Defect Report(s) applied: 937. See http://austingroupbugs.net/view.php?id=937.

**Change Number: XBD/TC2/D4/0005** [516]

On Page: 37 Line: 1229 Section: 3.27 Async-Signal-Safe Function

Change the definition from:

A function that may be invoked, without restriction, from signal-catching functions. No function is async-signal-safe unless explicitly described as such.

to:

A function that can be called, without restriction, from signal-catching functions. Note that, although there is no restriction on the calls themselves, for certain functions there are restrictions on subsequent behavior after the function is called from a signal-catching function. No function is async-signal-safe unless explicitly described as such.

Note: Async-signal-safety is defined in detail in [xref to XSH 2.4.3].

*Rationale*: Austin Group Defect Report(s) applied: 516. See http://austingroupbugs.net/view.php?id=516.
The restrictions on using longjmp() and siglongjmp() are more restrictive than they need to be on POSIX

systems. The loosened restrictions presented here do not break existing implementations and make it easier for application writers to create portable applications.

**Change Number: XBD/TC2/D4/0006** [653]

On Page: 39 Line: 1274 Section: 3.40 Basename

Change from:

The final, or only, filename in a pathname.

to:

For pathnames containing at least one filename: the final, or only, filename in the pathname. For pathnames consisting only of <slash> characters: either "/" or "//" if the pathname consists of exactly two <slash> characters, and "/" otherwise.

*Rationale*: Austin Group Defect Report(s) applied: 653. See http://austingroupbugs.net/view.php?id=653.

**Change Number: XBD/TC2/D4/0007** [834]

On Page: 60 Line: 1775 Section: 3.168 File Mode

Change from:

An object containing the file mode bits and file type of a file.

to:

An object containing the file mode bits and some information about the file type of a file.

*Rationale*: Austin Group Defect Report(s) applied: 834. See http://austingroupbugs.net/view.php?id=834.

**Change Number: XBD/TC2/D4/0008** [511]

On Page: 63 Line: 1855 Section: 3.188 Group ID

Add a sentence:

The value (**gid_t**)-1 shall not be a valid group ID, but does have a defined use in some interfaces defined in this standard.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XBD/TC2/D4/0009** [584]

On Page: 63 Line: 1859 Section: 3.189 Group Name

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0010** [690]

On Page: 66 Line: 1926 Section: 3 Definitions
(2013 edition Page: 66 Line: 1935)

After section 3.208 ("Link Count"), insert a new section **3.209 Live Process** whose contents are copied from section 3.289 ("Process") page 80 lines 2243-2250 (2013 edition section 3.291 page 80 lines 2269-2275):

An address space with one or more threads executing within that address space, and the required system resources for those threads.

<small>**Note:** Many of the system resources defined by POSIX.1-2008 are shared among all of the threads within a process. These include the process ID, the parent process ID, process group ID, session membership, real, effective, and saved set-user-ID, real, effective, and saved set-group-ID, supplementary group IDs, current working directory, root directory, file mode creation mask, and file descriptors.</small>

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XBD/TC2/D4/0011** [625]

On Page: 69 Line: 2000 Section: 3 Definitions

Add the following definitions in alphabetical order:

**Multi-threaded library**

A library containing object files that were produced by compiling with *c99* using the flags output by `getconf POSIX_V7_THREADS_CFLAGS`, or by compiling using a non-standard utility with equivalent flags, and which makes use of interfaces that are only made available by *c99* when the **-l pthread** option is used or makes use of SIGEV_THREAD notifications.

**Multi-threaded process**

A process that contains more than one thread.

**Multi-threaded program**

A program whose executable file was produced by compiling with *c99* using the flags output by `getconf POSIX_V7_THREADS_CFLAGS`, and linking with *c99* using the flags output by `getconf POSIX_V7_THREADS_LDFLAGS` and the **-l pthread** option, or by compiling and linking using a non-standard utility with equivalent flags. Execution of a multi-threaded program initially creates a single-threaded process; the process can create additional threads using *pthread_create*() or SIGEV_THREAD

notifications.

**Single-threaded process**

A process that contains a single thread.

**Single-threaded program**

A program whose executable file was produced by compiling with *c99* without using the flags output by `getconf POSIX_V7_THREADS_CFLAGS` and linking with *c99* using neither the flags output by `getconf POSIX_V7_THREADS_LDFLAGS` nor the **-l pthread** option, or by compiling and linking using a non-standard utility with equivalent flags. Execution of a single-threaded program creates a single-threaded process; if the process attempts to create additional threads using *pthread_create*() or SIGEV_THREAD notifications, the behavior is undefined. If the process uses *dlopen*() to load a multi-threaded library, the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 625. See http://austingroupbugs.net/view.php?id=625. The standard does not adequately address the differences between single-threaded and multi-threaded processes and programs.

**Change Number: XBD/TC2/D4/0012** [584]

On Page: 77 Line: 2193 Section: 3. Definitions

After Section 3.275, add a new section:

**3.2xx Portable Filename**

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

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

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584. This is a layered change on XBD/TC1/D5/0010 [291]. The change is to replace <hyphen> with <hyphen-minus>.

**Change Number: XBD/TC2/D4/0013** [584]

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

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0014** [690]

On Page: 80 Line: 2244-2250 Section: 3.289 Process
(2013 edition Page: 80 Line: 2269-2275 Section: 3.291)

Replace the definition with:

A live process (see Section 3.xxx) or a zombie process (see Section 3.xxx). The lifetime of a process is described in Section 3.xxx.

On Page: 81 Line: 2270-2279 Section: 3.295 Process Lifetime
(2013 edition Page: 81 Line: 2295-2304 Section: 3.297)

Change from:

The period of time that begins when a process is created and ends when its process ID is returned to the system. After a process is created by *fork*(), *posix_spawn*(), or *posix_spawnp*(), it is considered active. At least one thread of control and address space exist until it terminates. It then enters an inactive state where certain resources may be returned to the system, although some resources, such as the process ID, are still in use. When another process executes a *wait*(), *waitid*(), or *waitpid*() function for an inactive process, the remaining resources are returned to the system. The last resource to be returned to the system is the process ID. At this time, the lifetime of the process ends.

<small>**Note:** The *fork*(), *posix_spawn*(), *posix_spawnp*(), *wait*(), *waitid*(), and *waitpid*() functions are defined in detail in the System Interfaces volume of POSIX.1-2008.</small>

to:

The period of time that begins when a process is created and ends when its process ID is returned to the system.

See also *Live Process* in Section 3.xxx, *Process Termination* in Section 3.xxx, and *Zombie Process* in Section 3.xxx.

<small>**Note:** Process creation is defined in detail in the descriptions of the *fork*(), *posix_spawn*(), and *posix_spawnp*() functions in the System Interfaces volume of POSIX.1-2008.</small>

On Page: 81 Line: 2293-2294 Section: 3.297 Process Termination
(2013 edition Page: 81 Line: 2318-2319 Section: 3.299)

Change from:

<small>**Note:** The *_exit*(), *_Exit*(), *abort*(), and *exit*() functions are defined in detail in the System Interfaces volume of POSIX.1-2008.</small>

to:

<small>**Note:** The consequences of process termination can be found in the description of the *_Exit*() function in the System Interfaces volume of POSIX.1-2008. The *_exit*(), *_Exit*(), *abort*(), and *exit*() functions are defined in detail in the System Interfaces volume of POSIX.1-2008.</small>

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XBD/TC2/D4/0015** [896]

On Page: 90 Line: 2485 Section: 3.351 Source Code
(2013 edition Page: 90 Line: 2510 Section: 3.353 Source Code)

Change from:

... *ed, lex,* ...

to:

... *ed, ex, lex,* ...

On Page: 90 Line: 2487 Section: 3.351 Source Code
(2013 edition Page: 90 Line: 2512 Section: 3.353 Source Code)

Change from:

... *ed, lex,* ...

to:

... *ed, ex, lex,* ...

*Rationale*: Austin Group Defect Report(s) applied: 896. See http://austingroupbugs.net/view.php?id=896.

**Change Number: XBD/TC2/D4/0016** [511]

On Page: 102 Line: 2782 Section: 3.428 User ID

Add a sentence:

The value (**uid_t**)-1 shall not be a valid user ID, but does have a defined use in some interfaces defined in this standard.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XBD/TC2/D4/0017** [584]

On Page: 102 Line: 2786 Section: 3.429 User Name

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0018** [690]

On Page: 105 Line: 2845-2846 Section: 3.444 Zombie Process
(2013 edition Page: 105 Line: 2871-2872 Section: 3.446)

Change from:

A process that has terminated and that is deleted when its exit status has been reported to another process which is waiting for that process to terminate.

to:

The remains of a live process (see Section 3.xxx) after it terminates (see Section 3.xxx) and before its status information (see XSH Section 2.13) is consumed by its parent process.

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XBD/TC2/D4/0019** [934]

On Page: 107 Line: 2858 Section: 4 General Concepts
(2013 edition Page: 107 Line: 2884)

Add a new section (and renumber subsequent sections as needed):

**4.2 Default Initialization**

Default initialization causes an object to be initialized according to these rules:

- If it has pointer type, it is initialized to a null pointer.
- If it has arithmetic type, it is initialized to (positive or unsigned) zero.
- If it is an aggregate, every member is initialized (recursively) according to these rules.
- If it is a union, the first named member is initialized (recursively) according to these rules.

For an object of aggregate type with an explicit initializer, the initialization shall occur in initializer list order, each initializer provided for a particular subobject overriding any previously listed initializer for the same subobject; all subobjects that are not initialized explicitly shall be initialized implicitly according to the rules for default initialization.

Objects with static storage duration but no explicit initializer shall be initialized implicitly according to the rules for default initialization.

An explicit initializer of `{ 0 }` works to perform explicit default initialization for any object of scalar or aggregate type, and for any storage duration.

<small>Note: The C standard does not require a compiler to set any field alignment padding bits in a structure or array definition to a particular value. Because of this, a structure initialized using `{ 0 }` might not *memcmp*() as equal to the same structure initialized using *memset*() to zero. For consistent results, portable applications comparing structures should test each field individually.

Note: If an implementation treats the all-zero bit pattern of a pointer object as a null pointer, and the all-zero bit pattern of a floating point object as equivalent to positive 0, then *memset*() to zero and *calloc*() have the same effects as default initialization for all named members of a structure. [MX]Implementations

that define __STDC_IEC_559__ guarantee that the all-zero bit pattern of a floating point object represents 0.0.[/MX]</small>

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934.

**Change Number: XBD/TC2/D4/0020** [584]

On Page: 109 Line: 2914 Section: 4.7 Filename Portability

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0021** [626]

On Page: 109 Line: 2917 Section: 4.8 File Times Update

Insert a new paragraph at the start of the section:

Many operations have requirements to update file timestamps. These requirements do not apply to streams that have no underlying file description (for example, memory streams created by *open_memstream*() have no underlying file description).

*Rationale*: Austin Group Defect Report(s) applied: 626. See http://austingroupbugs.net/view.php?id=626.

**Change Number: XBD/TC2/D4/0022** [863]

On Page: 110 Line: 2980 Section: 4.11 Memory Synchronization
(2013 edition Page: 110 Line: 3004)

Change from:

The *pthread_once*() function shall synchronize memory for the first call in each thread for a given **pthread_once_t** object.

to:

The *pthread_once*() function shall synchronize memory for the first call in each thread for a given **pthread_once_t** object. If the *init_routine* called by *pthread_once*() is a cancellation point and is canceled, a call to *pthread_once*() for the same **pthread_once_t** object made from a cancellation cleanup handler shall also synchronize memory.

*Rationale*: Austin Group Defect Report(s) applied: 863. See http://austingroupbugs.net/view.php?id=863.

**Change Number: XBD/TC2/D4/0023** [541,649,825]

On Page: 112 Line: 3026 Section: 4.12 Pathname Resolution

Change from:

In all other cases, the system shall prefix the remaining pathname, if any, with the contents of the symbolic link. If the combined length exceeds {PATH_MAX}, and the implementation considers this to be an error, *errno* shall be set to [ENAMETOOLONG] and an error indication shall be returned.

to:

In all other cases, the system shall prefix the remaining pathname, if any, with the contents of the symbolic link, except that if the contents of the symbolic link is the empty string, then either pathname resolution shall fail with functions reporting an [ENOENT] error and utilities writing an equivalent diagnostic message, or the pathname of the directory containing the symbolic link shall be used in place of the contents of the symbolic link. If the contents of the symbolic link consist solely of <slash> characters, then all leading <slash> characters of the remaining pathname shall be omitted from the resulting combined pathname, leaving only the leading <slash> characters from the symbolic link contents. In the cases where prefixing occurs, if the combined length exceeds {PATH_MAX}, and the implementation considers this to be an error, pathname resolution shall fail with functions reporting an [ENAMETOOLONG] error and utilities writing an equivalent diagnostic message.

*Rationale*: Austin Group Defect Report(s) applied: 541,649,825. See
http://austingroupbugs.net/view.php?id=541.
See http://austingroupbugs.net/view.php?id=649.
See http://austingroupbugs.net/view.php?id=825.
The previous text was not historic practice on implementations that support //hostname as a special case in pathname resolution. For implementations that do not treat // as special, the change in handling of all-slash symbolic links has no effect on pathname resolution.

**Change Number: XBD/TC2/D4/0024** [825]

On Page: 112 Line: 3032 Section: 4.12 Pathname Resolution
(2013 edition Page: 112 Line: 3056 Section: 4.12)

Change from:

If the system detects a loop in the pathname resolution process, it shall set *errno* to [ELOOP] and return an error indication.

to:

If the system detects a loop in the pathname resolution process, pathname resolution shall fail with functions reporting an [ELOOP] error and utilities writing an equivalent diagnostic message.

*Rationale*: Austin Group Defect Report(s) applied: 825. See http://austingroupbugs.net/view.php?id=825.

**Change Number: XBD/TC2/D4/0025** [543]

On Page: 116 Line: 3205 Section: 4.19 Treatment of Error Conditions for Mathematical Functions

Add a new paragraph before the last paragraph:

On implementations that support the IEC 60559 Floating-Point option, whether or when functions in the <**math.h**> header raise an undeserved underflow floating-point exception is unspecified. Otherwise, as implied by [xref to XSH *feraiseexcept*()] the <**math.h**> functions do not raise spurious floating-point exceptions (detectable by the user), other than the inexact floating-point exception.

*Rationale*: Austin Group Defect Report(s) applied: 543. See http://austingroupbugs.net/view.php?id=543.

**Change Number: XBD/TC2/D4/0026** [584]

On Page: 123 Line: 3411 Section: 5 File Format Notation

Change from:

indicates either a <plus-sign> or minus-sign),

to:

indicates either a <plus-sign> or <hyphen-minus>),

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0027** [584,967]

On Page: 125 Line: 3454 Section: 6.1 Portable Character Set
(2013 edition Page: 125 Line: 3479)

Change from:

The first eight entries in [xref to Table 6-1] are defined in the ISO/IEC 6429:1992 standard and the rest of the characters are defined in the ISO/IEC 10646-1:2000 standard.

to:

The first eight entries in [xref to Table 6-1] and all characters in [xref to Table 6-2] are defined in the ISO/IEC 6429:1992 standard. The rest of the characters in [xref to Table 6-1] are defined in the ISO/IEC 10646-1:2000 standard.

On Page: 125 Line: 3458 Section: 6.1 Portable Character Set
(2013 edition Page: 125 Line: 3483)

Change from:

**Symbolic Name**

to:

**Symbolic Name(s)**

On Page: 125 Line: 3460-3466 Section: 6.1 Portable Character Set
(2013 edition Page: 125 Line: 3485-3491)

Change from:

| | | | |
|---|---|---|---|
| <alert> | | <U0007> | BELL (BEL) |
| <backspace> | | <U0008> | BACKSPACE (BS) |
| <tab> | | <U0009> | CHARACTER TABULATION (HT) |
| <carriage-return> | | <U000D> | CARRIAGE RETURN (CR) |
| <newline> | | <U000A> | LINE FEED (LF) |
| <vertical-tab> | | <U000B> | LINE TABULATION (VT) |
| <form-feed> | | <U000C> | FORM FEED (FF) |

to (note the description column changes to match the Unicode spec, and the change in line order):

| | | | |
|---|---|---|---|
| <alert>, <BEL> | | <U0007> | BELL |
| <backspace>, <BS> | | <U0008> | BACKSPACE |
| <tab>, <HT> | | <U0009> | CHARACTER TABULATION |
| <newline>, <LF> | | <U000A> | LINE FEED (LF) |
| <vertical-tab>, <VT> | | <U000B> | LINE TABULATION |
| <form-feed>, <FF> | | <U000C> | FORM FEED (FF) |
| <carriage-return>, <CR> | | <U000D> | CARRIAGE RETURN (CR) |

On Page: 125 Line: 3480-3485 Section: 6.1 Portable Character Set
(2013 edition Page: 125 Line: 3505-3510)

Change from:

| | | | |
|---|---|---|---|
| <hyphen-minus> | - | <U002D> | HYPHEN-MINUS |
| <hyphen> | - | <U002D> | HYPHEN-MINUS |
| <full-stop> | . | <U002E> | FULL STOP |
| <period> | . | <U002E> | FULL STOP |
| <slash> | / | <U002F> | SOLIDUS |
| <solidus> | / | <U002F> | SOLIDUS |

to (note that the glyph used for <hyphen-minus>, <hyphen> should be the same one that is used in the construct "('-')" in the rest of the standard):

| | | | |
|---|---|---|---|
| <hyphen-minus>, <hyphen> | - | <U002D> | HYPHEN-MINUS |
| <full-stop>, <period> | . | <U002E> | FULL STOP |
| <slash>, <solidus> | / | <U002F> | SOLIDUS |

On Page: 126 Line: 3531-3532 Section: 6.1 Portable Character Set
(2013 edition Page: 126 Line: 3556-3557)

Change from:

| | | | |
|---|---|---|---|
| <backslash> | \ | <U005C> | REVERSE SOLIDUS |
| <reverse-solidus> | \ | <U005C> | REVERSE SOLIDUS |

to:

| | | | |
|---|---|---|---|
| <backslash>, <reverse-solidus> | \ | <U005C> | REVERSE SOLIDUS |

On Page: 126 Line: 3534-3537 Section: 6.1 Portable Character Set
(2013 edition Page: 126 Line: 3559-3562)

Change from:

| | | | |
|---|---|---|---|
| <circumflex-accent> | ^ | <U005E> | CIRCUMFLEX ACCENT |

| | | | |
|---|---|---|---|
| <circumflex> | ^ | <U005E> | CIRCUMFLEX ACCENT |
| <low-line> | _ | <U005F> | LOW LINE |
| <underscore> | _ | <U005F> | LOW LINE |

395

396 to:

| | | | |
|---|---|---|---|
| <circumflex-accent>, <circumflex> | ^ | <U005E> | CIRCUMFLEX ACCENT |
| <low-line>, <underscore> | _ | <U005F> | LOW LINE |

397

398 On Page: 127 Line: 3566-3567 Section: 6.1 Portable Character Set
399 (2013 edition Page: 127 Line: 3591-3592)

400 Change from:

| | | | |
|---|---|---|---|
| <left-brace> | { | <U007B> | LEFT CURLY BRACKET |
| <left-curly-bracket> | { | <U007B> | LEFT CURLY BRACKET |

401

402 to:

| | | | |
|---|---|---|---|
| <left-brace>, <left-curly-bracket> | { | <U007B> | LEFT CURLY BRACKET |

403

404 On Page: 127 Line: 3569-3570 Section: 6.1 Portable Character Set
405 (2013 edition Page: 127 Line: 3594-3595)

406 Change from:

| | | | |
|---|---|---|---|
| <right-brace> | } | <U007D> | RIGHT CURLY BRACKET |
| <right-curly-bracket> | } | <U007D> | RIGHT CURLY BRACKET |

407

408 to:

| | | | |
|---|---|---|---|
| <right-brace>, <right-curly-bracket> | } | <U007D> | RIGHT CURLY BRACKET |

On Page: 127 Line: 3576-3578 Section: 6.1 Portable Character Set
(2013 edition Page: 127 Line: 3601-3603)

Delete the sentence:

The table contains more than one symbolic character name for characters whose traditional name differs from the chosen name.

*Rationale*: Austin Group Defect Report(s) applied: 584,967. See http://austingroupbugs.net/view.php?id=584.
See http://austingroupbugs.net/view.php?id=967.

**Change Number: XBD/TC2/D4/0028** [745]

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

Add a sentence as a new bullet item:

- The encoded values associated with <period>, <slash>, <newline> and <carriage-return> shall be invariant across all locales supported by the implementation.

*Rationale*: Austin Group Defect Report(s) applied: 745. See http://austingroupbugs.net/view.php?id=745.
This is a layered change on XBD/TC1/D5/0019 [291].

**Change Number: XBD/TC2/D4/0029** [663,967]

On Page: 128 Line: 3596 Section: 6.2 Character Encoding
(2013 edition Page: 128 Line: 3623)

Change from:

The POSIX locale contains the characters in [xref to Table 6-1], which have the properties listed in [xref to 7.3.1]. In other locales, the presence, meaning, and representation of any additional characters are locale-specific.

to:

The POSIX locale shall contain 256 single-byte characters including the characters in [xref to Table 6-1] and [xref to Table 6-2], which have the properties listed in [xref to 7.3.1]. It is unspecified whether characters not listed in those two tables are classified as **punct** or **cntrl**, or neither. Other locales shall contain the characters in [xref to Table 6-1] and may contain any or all of the control characters identified in [xref to Table 6-2]; the presence, meaning, and representation of any additional characters are locale-specific.

*Rationale*: Austin Group Defect Report(s) applied: 663,967. See
http://austingroupbugs.net/view.php?id=663.
See http://austingroupbugs.net/view.php?id=967.

This is a layered change on XBD/TC1/D5/0020 [216].
The intention was always that the POSIX locale should have an 8-bit-clean single-byte encoding. The omission of an explicit statement to that effect was an oversight.

**Change Number: XBD/TC2/D4/0030** [745]

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

Add a sentence:

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

*Rationale*: Austin Group Defect Report(s) applied: 745. See http://austingroupbugs.net/view.php?id=745.
This is a layered change on XBD/TC1/D5/0021 [291].

**Change Number: XBD/TC2/D4/0031** [967]

On Page: 129 Line: 3655 Section: 6.4 Character Set Description File
(2013 edition Page: 129 Line: 3683)

Change from:

Each symbolic name specified in [xref to Table 6-1] shall be included in the file and shall be mapped to a unique coding value, except as noted below. The glyphs represented by the C character constants `'{'`, `'}'`, `'_'`, `'-'`, `'/'`, `'\\'`, `'.'`, and `'^'` have more than one symbolic name; all symbolic names for each such glyph shall be included, each with identical encoding. If some or all of the control characters identified in [xref to Table 6-2] are supported by the implementation, 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. Some of the encodings associated with the symbolic names in [xref to Table 6-2] may be the same as characters found in [xref to Table 6-1]; both names shall be provided for each encoding.

to:

Each symbolic name specified in [xref to Table 6-1] shall be included in the file. Each character in [xref to Table 6-1] (each row in the table) shall be mapped to a unique coding value. For each character in [xref to Table 6-2] that exists in the character set described by the file, the character's symbolic name(s) from [xref to Table 6-2] and the character's single-byte encoding value shall be included in the file.

On Page: 130 Line: 3664-3670 Section: 6.4 Character Set Description File
(2013 edition Page: 130 Line: 3692-3698)

Change from:

**Table 6-2** Control Character Set

<ACK> <DC2> <ENQ> <FS> <IS4> <SOH>

<BEL> <DC3> <EOT> <GS> <LF> <STX>

<BS> <DC4> <ESC> <HT> <NAK> <SUB>

<CAN> <DEL> <ETB> <IS1> <RS> <SYN>

<CR> <DLE> <ETX> <IS2> <SI> <US>

<DC1> <EM> <FF> <IS3> <SO> <VT>

to:

**Table 6-2** Non-Portable Control Characters

| Symbolic Name(s) | UCS | Description |
|---|---|---|
| <SOH> | <U0001> | START OF HEADING |
| <STX> | <U0002> | START OF TEXT |
| <ETX> | <U0003> | END OF TEXT |
| <EOT> | <U0004> | END OF TRANSMISSION |
| <ENQ> | <U0005> | ENQUIRY |
| <ACK> | <U0006> | ACKNOWLEDGE |
| <SO> | <U000E> | SHIFT OUT |
| <SI> | <U000F> | SHIFT IN |
| <DLE> | <U0010> | DATA LINK ESCAPE |
| <DC1> | <U0011> | DEVICE CONTROL ONE |
| <DC2> | <U0012> | DEVICE CONTROL TWO |
| <DC3> | <U0013> | DEVICE CONTROL THREE |
| <DC4> | <U0014> | DEVICE CONTROL FOUR |
| <NAK> | <U0015> | NEGATIVE ACKNOWLEDGE |
| <SYN> | <U0016> | SYNCHRONOUS IDLE |
| <ETB> | <U0017> | END OF TRANSMISSION BLOCK |

| | | |
|---|---|---|
| <CAN> | <U0018> | CANCEL |
| <EM> | <U0019> | END OF MEDIUM |
| <SUB> | <U001A> | SUBSTITUTE |
| <ESC> | <U001B> | ESCAPE |
| <IS4>, <FS> | <U001C> | INFORMATION SEPARATOR FOUR |
| <IS3>, <GS> | <U001D> | INFORMATION SEPARATOR THREE |
| <IS2>, <RS> | <U001E> | INFORMATION SEPARATOR TWO |
| <IS1>, <US> | <U001F> | INFORMATION SEPARATOR ONE |
| <DEL> | <U007F> | DELETE |

*Rationale*: Austin Group Defect Report(s) applied: 967. See http://austingroupbugs.net/view.php?id=967.

**Change Number: XBD/TC2/D4/0032** [796]

On Page: 136 Line: 3846 Section: 7.2 POSIX Locale
(2013 edition Page: 136 Line: 3882)

After:

Conforming systems shall provide a POSIX locale, also known as the C locale.

add:

In POSIX.1 the requirements for the POSIX locale are more extensive than the requirements for the C locale as specified in the C standard. However, in a conforming POSIX implementation, the POSIX locale and the C locale are identical.

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XBD/TC2/D4/0033** [663]

On Page: 136 Line: 3849 Section: 7.2 POSIX locale
(2013 edition Page: 136 Line: 3885)

Delete:

The tables in Section 7.3 describe the characteristics and behavior of the POSIX locale for data consisting entirely of characters from the portable character set and the control character set. For other characters, the behavior is unspecified.

On Page: 139 Line: 3996 Section: 7.3.1 LC_CTYPE
(2013 edition Page: 139 Line: 4032)

Change from:

In the POSIX locale, the 26 uppercase letters shall be included:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

to:

In the POSIX locale, only:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

shall be included.

On Page: 139 Line: 4003 Section: 7.3.1 LC_CTYPE
(2013 edition Page: 139 Line: 4039)

Change from:

In the POSIX locale, the 26 lowercase letters shall be included:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

to:

In the POSIX locale, only:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

shall be included.

On Page: 139 Line: 4009 Section: 7.3.1 LC_CTYPE
(2013 edition Page: 140 Line: 4045)

Change from:

In the POSIX locale, all characters in the classes **upper** and **lower** shall be included.

to:

In the POSIX locale, only characters in the classes **upper** and **lower** shall be included.

On Page: 141 Line: 4091 Section: 7.3.1 LC_CTYPE
(2013 edition Page: 141 Line: 4127)

Change from:

In the POSIX locale, at a minimum, the 26 lowercase characters:

to:

In the POSIX locale, the 26 lowercase characters:

On Page: 142 Line: 4105 Section: 7.3.1 LC_CTYPE
(2013 edition Page: 142 Line: 4141)

Change from:

In the POSIX locale, at a minimum, the 26 uppercase characters:

to:

In the POSIX locale, the 26 uppercase characters:

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XBD/TC2/D4/0034** [663]

On Page: 143 Line: 4142 Section: 7.3.1.1 LC_CTYPE Category in the POSIX Locale
(2013 edition Page: 143 Line: 4178)

Change from:

The character classifications for the POSIX locale follow; the code listing depicts the *localedef* input, and the table represents the same information, sorted by character.

to:

The minimum character classifications for the POSIX locale follow; the code listing depicts the *localedef* input, and the table represents the same information, sorted by character. Implementations may add additional characters to the **cntrl** and **punct** classifications but shall not make any other additions.

On Page: 143 Line: 4145 Section: 7.3.1.1 LC_CTYPE Category in the POSIX Locale
(2013 edition Page: 143 Line: 4181)

Change from:

```
# The following is the POSIX locale LC_CTYPE.
# "alpha" is by default "upper" and "lower"
# "alnum" is by definition "alpha" and "digit"
# "print" is by default "alnum", "punct", and the <space>
# "graph" is by default "alnum" and "punct"
```

to:

```
# The following is the minimum POSIX locale LC_CTYPE.
# "alpha" is by definition "upper" and "lower"
# "alnum" is by definition "alpha" and "digit"
# "print" is by definition "alnum", "punct", and the <space>
# "graph" is by definition "alnum" and "punct"
```

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XBD/TC2/D4/0035** [584]

On Page: 143 Line: 4173 Section: 7.3.1.1 LC_CTYPE Category in the POSIX Locale

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0036** [584]

On Page: 144 Line: 4243 Section: 7.3.1.1 LC_CTYPE Category in the POSIX Locale

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0037** [938]

On Page: 146 Line: 4330 Section: 7.3.2 LC_COLLATE
(2013 edition Page: 147 Line: 4366)

Change from:

(*sort*, *uniq*, and so on)

to:

(*ls*, *sort*, and so on)

On Page: 147 Line: 4357 Section: 7.3.2 LC_COLLATE
(2013 edition Page: 147 Line: 4393)

Add a new paragraph and two small-font notes after the numbered list:

All implementation-provided locales (either preinstalled or provided as locale definitions which can be installed later) should define a collation sequence that has a total ordering of all characters unless the locale name has an '@' modifier indicating that it has a special collation sequence (for example, `@icase` could indicate that each upper- and lower-case character pair collates equally).

<small>Note: a future version of this standard may require these locales to define a collation sequence that has a total ordering of all characters (by changing "should" to "shall").</small>

<small>Note: users installing their own locales should ensure that they define a collation sequence with a total ordering of all characters unless an '@' modifier in the locale name (such as @icase) indicates that it has a special collation sequence.</small>

On Page: 149 Line: 4467 Section: 7.3.2.4 Collation Order
(2013 edition Page: 150 Line: 4503)

Add a new paragraph and a small-font note:

Weights should be assigned such that the collation sequence has a total ordering of all characters unless an '@' modifier in the locale name indicates that it has a special collation sequence.

<small>Note: a future version of this standard may require a total ordering of all characters for implementation-provided locales that do not have an '@' modifier in the locale name. See [xref to 7.3.2].</small>

On Page: 150 Line: 4481 Section: 7.3.2.4 Collation Order
(2013 edition Page: 150 Line: 4517)

Change from:

Characters specified via an explicit or implicit UNDEFINED special symbol shall by default be assigned the same primary weight (that is, they belong to the same equivalence class).

to:

Characters specified via an explicit or implicit UNDEFINED special symbol shall by default be assigned the same primary weight (that is, they belong to the same equivalence class) if the collation order has more than one weight level. If the collation order has only one weight level, these characters should be assigned unique primary weights, equal to the relative order of their character in the character collation sequence, but may be assigned the same primary weight.

<small>Note: a future version of this standard may require these characters to be assigned unique primary weights if the collation order has only one weight level.</small>

On Page: 150 Line: 4503 Section: 7.3.2.4 Collation Order
(2013 edition Page: 151 Line: 4539)

Delete the first entry in the example collation order:

```
UNDEFINED IGNORE;IGNORE
```

On Page: 150 Line: 4516 Section: 7.3.2.4 Collation Order
(2013 edition Page: 151 Line: 4552)

Add the following as the last entry in the example collation order:

```
UNDEFINED IGNORE;...
```

On Page: 151 Line: 4519 Section: 7.3.2.4 Collation Order
(2013 edition Page: 151 Line: 4555)

Delete item 1 in the numbered list:

The **UNDEFINED** means that all characters not specified in this definition (explicitly or via the ellipsis) shall be ignored for collation purposes.

and renumber items 2-4 to be 1-3.

On Page: 151 Line: 4527 Section: 7.3.2.4 Collation Order
(2013 edition Page: 151 Line: 4563)

Add a new item 4 to the numbered list:

The **UNDEFINED** means that all characters not specified in this definition (explicitly or via the ellipsis) shall be ignored when comparing primary weights, and have individual secondary weights based on their ordinal encoded values.

*Rationale*: Austin Group Defect Report(s) applied: 938. See http://austingroupbugs.net/view.php?id=938.

**Change Number: XBD/TC2/D4/0038** [663]

On Page: 151 Line: 4531 Section: 7.3.2.6 LC_COLLATE Category in the POSIX Locale
(2013 edition Page: 151 Line: 4567)

Change from:

The collation sequence definition of the POSIX locale follows; the code listing depicts the *localedef* input.

to:

The minimum collation sequence definition of the POSIX locale follows; the code listing depicts the *localedef* input. All characters not explicitly listed here shall be inserted in the character collation order after the listed characters and shall be assigned unique primary weights. If the listed characters have ASCII encoding, the other characters shall be in ascending order according to their coded character set values; otherwise, the order of the other characters is unspecified. The collation sequence shall not include any multi-character collating elements.

On Page: 151 Line: 4534 Section: 7.3.2.6 LC_COLLATE Category in the POSIX Locale
(2013 edition Page: 151 Line: 4570)

Change from:

```
# This is the POSIX locale definition for the LC_COLLATE category.
# The order is the same as in the ASCII codeset.
```

to:

```
# This is the minimum input for the POSIX locale definition for the
# LC_COLLATE category. Characters in this list are in the same order
# as in the ASCII codeset.
```

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XBD/TC2/D4/0039** [584]

On Page: 152 Line: 4582 Section: 7.3.2.6 LC_COLLATE Category in the POSIX Locale

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0040** [912]

On Page: 163 Line: 5073 Section: 7.3.5.3 LC_TIME Category in the POSIX Locale
(2013 edition Page: 163 Line: 5109)

In the LC_TIME category definition, change from:

<percent_sign>

to:

<percent-sign>

*Rationale*: Austin Group Defect Report(s) applied: 912. See http://austingroupbugs.net/view.php?id=912.

**Change Number: XBD/TC2/D4/0041** [584]

On Page: 179 Line: 5728 Section: 8.3 Other Environment Variables

In the description of TZ change from:

<plus-sign> ('+') character, or the minus-sign

to:

<plus-sign> ('+') character, or the <hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0042** [554]

On Page: 182 Line: 5842-5848 Section: 9.2 Regular Expression General Requirements

Change from:

In the regular expression processing described in POSIX.1-2008, the <newline> is regarded as an ordinary character and both a <period> and a non-matching list can match one. The Shell and Utilities volume of POSIX.1-2008 specifies within the individual descriptions of those standard utilities employing regular expressions whether they permit matching of <newline> characters; if not stated otherwise, the use of literal <newline> characters or any escape sequence equivalent produces undefined results.

to:

In the functions processing regular expressions described in the System Interfaces volume of POSIX.1-2008, the <newline> is regarded as an ordinary character and both a <period> and a non-matching list can match one. The Shell and Utilities volume of POSIX.1-2008 specifies within the individual descriptions of those standard utilities employing regular expressions whether they permit matching of <newline> characters; if not stated otherwise, the use of literal <newline> characters or any escape sequence equivalent in either patterns or matched text produces undefined results .

**Change Number: XBD/TC2/D4/0043** [554]

On Page: 183 Line: 5875 Section: 9.3.2 BRE Ordinary Characters

Change from:

The interpretation of an ordinary character preceded by a <backslash> ('\\') is undefined, except for:

to:

The interpretation of an ordinary character preceded by an unescaped <backslash> ('\\') is undefined, except for:

On Page: 183 Line: 5887 Section: 9.3.3 BRE Special Characters

Change from:

that is not preceded by a <backslash>

to:

that is unescaped

On Page: 184 Line: 5894-5896 Section: 9.3.3 BRE Special Characters

Change from:

The <circumflex> shall be special when used as:
- An anchor (see Section XXX on page YYY)
- The first character of a bracket expression (see Section AAA on page BBB)

to:

The <circumflex> shall be special when used as an anchor (see Section XXX, on page YYY). The <circumflex> shall signify a non matching list expression when it occurs first in a list, immediately following a <left-square-bracket> (see Section AAA on page BBB).

*Rationale*: Austin Group Defect Report(s) applied: 554. See http://austingroupbugs.net/view.php?id=554.

**Change Number: XBD/TC2/D4/0044** [938]

On Page: 184 Line: 5902 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 184 Line: 5944)

Change from:

A bracket expression (an expression enclosed in square brackets, "[ ]") is an RE that shall match a single collating element contained in the non-empty set of collating elements represented by the bracket expression.

to:

A bracket expression (an expression enclosed in square brackets, "[ ]") is an RE that shall match a specific set of single characters, and may match a specific set of multi-character collating elements, based on the non-empty set of list expressions contained in the bracket expression.

On Page: 184 Line: 5907 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 184 Line: 5949)

In list item 1, change from:

It consists of one or more expressions: collating elements, collating symbols, ...

to:

It consists of one or more expressions: ordinary characters, collating elements, collating symbols, ...

On Page: 184 Line: 5921 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 184 Line: 5963)

In list item 2, change from:

A matching list expression specifies a list that shall match any single-character collating element in any of the expressions represented in the list. The first character in the list shall not be the <circumflex>; for example, "[abc]" is an RE that matches any of the characters 'a', 'b', or 'c'.

to:

A matching list expression specifies a list that shall match any single character that is matched by one of the expressions represented in the list. The first character in the list can not be the <circumflex>. An ordinary character in the list should only match that character, but may match any single character that collates equally with that character; for example, "[abc]" is an RE that should only match one of the characters 'a', 'b', or 'c'.

<small>Note: a future version of this standard may require that an ordinary character in the list only matches that character.</small>

*Rationale*: Austin Group Defect Report(s) applied: 938. See http://austingroupbugs.net/view.php?id=938.

**Change Number: XBD/TC2/D4/0045** [872]

On Page: 184 Line: 5926 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 184 Line: 5968)

In list item 3, change from:

A non-matching list expression begins with a <circumflex> ('^'), and specifies a list that shall match any single-character collating element except for the expressions represented in the list after the leading <circumflex>.

to:

A non-matching list expression begins with a <circumflex> ('^'), and the matching behavior shall be the logical inverse of the corresponding matching list expression (the same bracket expression but without the leading <circumflex>).

*Rationale*: Austin Group Defect Report(s) applied: 872. See http://austingroupbugs.net/view.php?id=872.

**Change Number: XBD/TC2/D4/0046** [938]

On Page: 184 Line: 5928 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 184 Line: 5970)

In list item 3, change from:

For example, "[^abc]" is an RE that matches any character except the characters 'a', 'b', or 'c'.

to:

For example, if the RE "[abc]" only matches 'a', 'b', or 'c', then "[^abc]" is an RE that matches any character except 'a', 'b', or 'c'.

On Page: 185 Line: 5953 Section: 9.3.5 RE Bracket Expression
(2013 edition Page: 185 Line: 5995)

In list item 6, change from:

The set of single-character collating elements whose characters belong to the character class

to:

The set of single characters that belong to the character class

*Rationale*: Austin Group Defect Report(s) applied: 938. See http://austingroupbugs.net/view.php?id=938.

**Change Number: XBD/TC2/D4/0047** [584]

On Page: 185 Line: 5972 Section: 9.3.5 RE Bracket Expression

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0048** [584]

On Page: 186 Line: 5982,5989,5992 Section: 9.3.5 RE Bracket Expression

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0049** [595]

On Page: 186 Line: 6016 Section: 9.3.6 BREs Matching Multiple Characters

Change rule 3 from:

For example, the expression "^\(.*\)\1$" matches lines consisting of two adjacent appearances of the same string,

to:

For example, the expression "^\(.*\)\1$" matches strings consisting of two adjacent appearances of the same substring,

On Page: 187 Line: 6053 Section: 9.3.8 BRE Expression Anchoring

Change from:

A BRE can be limited to matching strings that begin or end a line; this is called "anchoring".

to:

A BRE can be limited to matching expressions that begin or end a string; this is called "anchoring".

*Rationale*: Austin Group Defect Report(s) applied: 595. See http://austingroupbugs.net/view.php?id=595.

**Change Number: XBD/TC2/D4/0050** [554]

On Page: 188 Line: 6082-6083 Section: 9.4.2 ERE Ordinary Characters

Change from:

The interpretation of an ordinary character preceded by a <backslash> ('\\') is undefined.

to:

The interpretation of an ordinary character preceded by an unescaped <backslash> ('\\') is undefined, except in the context of a bracket expression (see XREF to ERE bracket expression).

On Page: 188 Line: 6092 Section: 9.4.3 ERE Special Characters

Add the following text after line 6092:

A <left-square-bracket> that is unescaped and is not part of a bracket expression also produces undefined results.

On Page: 188 Line: 6098-6099 Section: 9.4.3 ERE Special Characters

Change from:

If these characters appear first in an ERE, or immediately following a <vertical-line>, <circumflex>, or <left-parenthesis>

to:

If these characters appear first in an ERE, or immediately following an unescaped <vertical-line>, <circumflex>, <dollar-sign>, or <left-parenthesis>

On Page: 189 Line: 6106-6109 Section: 9.4.3 ERE Special Characters

Change from:

The <circumflex> shall be special when used as:
- An anchor (see Section XXX on page YYY)
- The first character of a bracket expression (see Section AAA on page BBB)

to:

The <circumflex> shall be special when used as an anchor (see Section XXX, on page YYY). The <circumflex> shall signify a non matching list expression when it occurs first in a list, immediately following a <left-square-bracket> (see Section AAA on page BBB).

*Rationale*: Austin Group Defect Report(s) applied: 554. See http://austingroupbugs.net/view.php?id=554.

**Change Number: XBD/TC2/D4/0051** [595]

On Page: 190 Line: 6175 Section: 9.4.9 ERE Expression Anchoring

Change from:

An ERE can be limited to matching strings that begin or end a line; this is called "anchoring".

to:

An ERE can be limited to matching expressions that begin or end a string; this is called "anchoring".

*Rationale*: Austin Group Defect Report(s) applied: 595. See http://austingroupbugs.net/view.php?id=595.

**Change Number: XBD/TC2/D4/0052** [554]

On Page: 192 Line: 6233 Section: 9.5.1 BRE/ERE Grammar Lexical Conventions

For token SPEC_CHAR delete the phrase:

or when first in a bracket expression

On Page: 195 Line: 6379-6380 Section: 9.5.3 ERE Grammar

Change from:

The ERE grammar does not permit several constructs that previous sections specify as having undefined results:

to:

The ERE grammar does not permit several constructs that previous sections specify as having undefined results. Additionally, there are some constructs which the grammar permits but which still give undefined results:

On Page: 195 Line: 6381 Section: 9.5.3 ERE Grammar

Change from:

ORD_CHAR preceded by a <backslash> character

to:

ORD_CHAR preceded by an unescaped <backslash> character

On Page: 195 Line: 6383 Section: 9.5.3 ERE Grammar

Add '$' to the list of characters.

*Rationale*: Austin Group Defect Report(s) applied: 554. See http://austingroupbugs.net/view.php?id=554.

**Change Number: XBD/TC2/D4/0053** [916]

On Page: 195 Line: 6387 Section: 9.5.3 ERE Grammar
(2013 edition Page: 195 Line: 6433)

Change from:

Conforming applications cannot use such constructs.

to:

Strictly conforming applications cannot use such constructs.

*Rationale*: Austin Group Defect Report(s) applied: 916. See http://austingroupbugs.net/view.php?id=916.

**Change Number: XBD/TC2/D4/0054** [967]

On Page: 198 Line: 6438-6454 Section: 10.2 Output Devices and Terminal Types
(2013 edition Page: 198 Line: 6484-6500)

Delete the two **Symbolic Name** columns from Table 10-1.

*Rationale*: Austin Group Defect Report(s) applied: 967. See http://austingroupbugs.net/view.php?id=967.

**Change Number: XBD/TC2/D4/0055** [745]

On Page: 204 Line: 6682,6684 Section: 11.1.9 Special Characters

On line 6682, for CR change from:

carriage-return

to:

<carriage-return>

On line 6684, append a sentence to the description of CR:

It cannot be changed.

*Rationale*: Austin Group Defect Report(s) applied: 745. See http://austingroupbugs.net/view.php?id=745.

**Change Number: XBD/TC2/D4/0056** [584]

On Page: 213 Line: 7006 Section: 12.1 Utility Argument Syntax

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XBD/TC2/D4/0057** [813]

On Page: 214 Line: 7056 Section: 12.1 Utility Argument Syntax

After the third bullet item, add a new bullet item:

- When the utility description states that the number is a file size-related value (such as a file size or offset, line number, or block count), numerals in the range 0 to the maximum file size supported by the implementation are syntactically recognized as numeric values (see XCU 1.5 Considerations for Utilities in Support of Files of Arbitrary Size). Where negative values are permitted, any value in the range -(maximum file size) to the maximum file size is accepted.

*Rationale*: Austin Group Defect Report(s) applied: 813. See http://austingroupbugs.net/view.php?id=813. The normative requirement in XCU 1.5 does not match the syntax utility guidelines which suggest very large values may be supported but are not required to be.

**Change Number: XBD/TC2/D4/0058** [579]

On Page: 220 Line: 7211 Section: <aio.h>

In the DESCRIPTION section, change from:

The tag **sigevent** shall be declared as naming an incomplete structure type, the contents of which are described in the <**signal.h**> header.

to:

The <**aio.h**> header shall define the **sigevent** structure and **sigval** union as described in <**signal.h**>.

*Rationale*: Austin Group Defect Report(s) applied: 579. See http://austingroupbugs.net/view.php?id=579.

**Change Number: XBD/TC2/D4/0059** [496]

On Page: 236 Line: 7737 Section: <errno.h>

In the DESCRIPTION section, change from:

[ENOSYS] Function not supported.

to:

[ENOSYS] Functionality not supported.

*Rationale*: Austin Group Defect Report(s) applied: 496. See http://austingroupbugs.net/view.php?id=496.

**Change Number: XBD/TC2/D4/0060** [847]

On Page: 240 Line: 7878 Section: <fcntl.h>
(2013 edition Page: 238 Line: 7885)

In the DESCRIPTION section, change from:

O_DIRECTORY Fail if not a directory.

to:

O_DIRECTORY Fail if file is a non-directory file.

*Rationale*: Austin Group Defect Report(s) applied: 847. See http://austingroupbugs.net/view.php?id=847.

**Change Number: XBD/TC2/D4/0061** [666]

On Page: 270 Line: 8853-8856 Section: <limits.h>

In the DESCRIPTION section, delete:

{RE_DUP_MAX}
Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6 (on page 186) and Section 9.4.6 (on page 189).
Minimum Acceptable Value: {_POSIX2_RE_DUP_MAX}

On Page: 273 Line: 9001-9003 Section: <limits.h>

Change from:

{RE_DUP_MAX}
Maximum number of repeated occurrences of a regular expression permitted when using the interval notation \{*m*,*n*\}; see Chapter 9 (on page 181).

to (copied from line 8854):

{RE_DUP_MAX}
Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6 (on page 186) and Section 9.4.6 (on page 189).

On Page: 273 Line: 9004 Section: <limits.h>

Change from:

{_POSIX2_RE_DUP_MAX}

to:

{_POSIX_RE_DUP_MAX}

On Page: 275 Line: 9076-9077 Section: <limits.h>

Change from:

The number of repeated occurrences of a BRE permitted by the *regexec*() and *regcomp*() functions when using the interval notation {\(*m*,*n*\}; see Section 9.3.6 (on page 186).

to (copied from line 8854):

Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6 (on page 186) and Section 9.4.6 (on page 189).

On Page: 277 Line: 9174-9175 Section: <limits.h>

Change from:

Maximum number of repeated occurrences of a regular expression permitted when using the interval notation \{*m*,*n*\}; see Chapter 9 (on page 181).

to (copied from line 8854):

Maximum number of repeated occurrences of a BRE or ERE interval expression; see Section 9.3.6 (on page 186) and Section 9.4.6 (on page 189).

*Rationale*: Austin Group Defect Report(s) applied: 666. See http://austingroupbugs.net/view.php?id=666.

**Change Number: XBD/TC2/D4/0062** [781]

On Page: 283 Line: 9438 Section: <locale.h>
(2013 edition Page: 286 Line: 9521)

In the DESCRIPTION section, change from:

Implementations may add additional masks using the form *LC*_* and an uppercase letter.

to:

Additional macro definitions, beginning with the characters *LC*_ and an uppercase letter, may also be specified by the implementation.

On Page: 284 Line: 9447 Section: <locale.h>
(2013 edition Page: 287 Line: 9530)

Delete:

[CX]Implementations may add additional masks using the form *LC*_*_*MASK*.[/CX]

On Page: 284 Line: 9464 Section: <locale.h>
(2013 edition Page: 287 Line: 9547)

In the RATIONALE section, change from:

None.

to:

It is suggested that each category macro name for use in *setlocale*() have a corresponding macro name ending in *_MASK* for use in *newlocale*().

*Rationale*: Austin Group Defect Report(s) applied: 781. See http://austingroupbugs.net/view.php?id=781.

**Change Number: XBD/TC2/D4/0063** [801]

On Page: 286 Line: 9512 Section: <math.h>
(2013 edition Page: 289 Line: 9596)

In the DESCRIPTION section, change from:

shall be accurate within the precision of the **double** type

to:

shall be accurate to at least the precision of the **double** type

*Rationale*: Austin Group Defect Report(s) applied: 801. See http://austingroupbugs.net/view.php?id=801.

**Change Number: XBD/TC2/D4/0064** [801]

On Page: 292 Line: 9776 Section: <math.h>
(2013 edition Page: 295 Line: 9860)

Add a note to the APPLICATION USAGE section:

Note that if FLT_EVAL_METHOD is neither 0 nor 1, then some constants might not compare equal as expected, for example `(double)M_PI == M_PI` can fail.

*Rationale*: Austin Group Defect Report(s) applied: 801. See http://austingroupbugs.net/view.php?id=801.

**Change Number: XBD/TC2/D4/0065** [934]

On Page: 303 Line: 10154 Section: <netinet/in.h>
(2013 edition Page: 306 Line: 10238)

In the DESCRIPTION section, change from:

The **sockaddr_in6** structure shall be set to zero by an application prior to using it, since implementations are free to have additional, implementation-defined fields in **sockaddr_in6**.

to:

Prior to calling a function in this standard which reads values from a **sockaddr_in6** structure (for example, *bind*() or *connect*()), the application shall ensure that all members of the structure, including any additional non-standard members, if any, are initialized. If the **sockaddr_in6** structure has a non-standard member, and that member has a value other than the value that would result from default initialization, the behavior of any function in this standard that reads values from the **sockaddr_in6** structure is implementation-

defined. All functions in this standard that return data in a **sockaddr_in6** structure (for example, *getaddrinfo*() or *accept*()) shall initialize the structure in a way that meets the above requirements, and shall ensure that each non-standard member, if any, has a value that produces the same behavior as default initialization would in all functions in this standard which read values from a **sockaddr_in6** structure.

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934.

**Change Number: XBD/TC2/D4/0066** [952]

On Page: 304 Line: 10185 Section: <netinet/in.h>
(2013 edition Page: 307 Line: 10269)

In the DESCRIPTION section, change from:

The **<netinet/in.h>** header shall define the following symbolic constants for use as destination addresses for *connect*(), *sendmsg*(), and *sendto*():

INADDR_ANY  IPv4 local host address.

INADDR_BROADCAST  IPv4 broadcast address.

to:

The **<netinet/in.h>** header shall define the following symbolic constant for use as a local address in the structure passed to *bind*():

INADDR_ANY  IPv4 wildcard address.

The **<netinet/in.h>** header shall define the following symbolic constant for use as a destination address in the structures passed to *connect*(), *sendmsg*(), and *sendto*():

INADDR_BROADCAST  IPv4 broadcast address.

*Rationale*: Austin Group Defect Report(s) applied: 952. See http://austingroupbugs.net/view.php?id=952.

**Change Number: XBD/TC2/D4/0067** [934]

On Page: 306 Line: 10239 Section: <netinet/in.h>
(2013 edition Page: 309 Line: 10323)

In the APPLICATION USAGE section, change from:

None.

to:

Although applications are required to initialize all members (including any non-standard ones) of a **sockaddr_in6** structure, the same is not required for the **sockaddr_in** structure, since historically many applications only initialized the standard members. Despite this, applications are encouraged to initialize **sockaddr_in** structures in a manner similar to the required initialization of **sockaddr_in6** structures.

Although it is common practice to initialize a **sockaddr_in6** structure using:

```
struct sockaddr_in6 sa;
memset(&sa, 0, sizeof sa);
```

this method is not portable according to this standard, because the structure can contain pointer or floating point members that are not required to have an all-bits-zero representation after default initialization. Portable methods make use of default initialization, for example:

```
struct sockaddr_in6 sa = { 0 };
```

or

```
static struct sockaddr_in6 sa_init;
struct sockaddr_in6 sa = sa_init;
```

A future version of this standard may require that a pointer object with an all-bits-zero representation is a null pointer, and that **sockaddr_in6** does not have any floating-point members if a floating-point object with an all-bits-zero representation does not have the value 0.0.

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934.

**Change Number: XBD/TC2/D4/0068** [952]

On Page: 306 Line: 10241 Section: <netinet/in.h>
(2013 edition Page: 309 Line: 10325)

In the RATIONALE section, change from:

None.

to:

The INADDR_ANY and INADDR_BROADCAST values are byte-order-neutral and thus their byte order is not specified. Many implementations have additional constants as extensions, such as INADDR_LOOPBACK, that are not byte-order-neutral. Traditionally, these constants are in host byte order, requiring the use of *htonl*() when using them in a **sockaddr_in** structure.

*Rationale*: Austin Group Defect Report(s) applied: 952. See http://austingroupbugs.net/view.php?id=952.

**Change Number: XBD/TC2/D4/0069** [624]

On Page: 312 Line: 10448-10449 Section: <pthread.h>

In the DESCRIPTION section, delete pthread_cleanup_push and pthread_cleanup_pop from the list of functions.

Add a new paragraph after line 10542:

The following may be declared as functions, or defined as macros, or both.
If functions are declared, function prototypes shall be provided.

*pthread_cleanup_pop*
*pthread_cleanup_push*

*Rationale*: Austin Group Defect Report(s) applied: 624. See http://austingroupbugs.net/view.php?id=624.

**Change Number: XBD/TC2/D4/0070** [536]

On Page: 329 Line: 11005-11006 Section: <signal.h>

In the DESCRIPTION section, change from:

[CX]The ISO C standard only requires the signal names SIGABRT, SIGFPE, SIGILL, SIGINT, SIGSEGV, and SIGTERM to be defined.[/CX]

to:

The ISO C standard only requires the signal names SIGABRT, SIGFPE, SIGILL, SIGINT, SIGSEGV, and SIGTERM to be defined. An implementation need not generate any of these six signals, except as a result of explicit use of interfaces that generate signals, such as *raise*(), [CX]*kill*(), the General Terminal Interface (see section 11.1.9), and the *kill* utility, unless otherwise stated (see for example XSH 2.8.3.3)[/CX].

*Rationale*: Austin Group Defect Report(s) applied: 536. See http://austingroupbugs.net/view.php?id=536.

**Change Number: XBD/TC2/D4/0071** [690]

On Page: 331 Line: 11073 Section: <signal.h>
(2013 edition Page: 334 Line: 11158)

In the DESCRIPTION section change from:

[CX]SA_NOCLDWAIT Causes implementations not to create zombie processes on child death.[/CX]

to:

[XSI]SA_NOCLDWAIT Causes implementations not to create zombie processes or status information on child termination. See *sigaction*(), page 1932.[/XSI]

(Note the change from CX shading to XSI shading.)

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XBD/TC2/D4/0072** [594]

On Page: 334 Line: 11164 Section: <signal.h>

In the DESCRIPTION section, in the "Value" column for "**int** *si_status*" change from:

Exit value or signal.

to:

If *si_code* is equal to CLD_EXITED, then *si_status* holds the exit value of the process; otherwise, it is equal to the signal that caused the process to change state. The exit value in *si_status* shall be equal to the full exit value (that is, the value passed to *_exit*(), *_Exit*() or *exit*(), or returned from *main*()); it shall not be limited to the least significant eight bits of the value.

*Rationale*: Austin Group Defect Report(s) applied: 594. See http://austingroupbugs.net/view.php?id=594. The standard is unclear about when si_status contains an exit status and when it contains a signal.

**Change Number: XBD/TC2/D4/0073** [844]

On Page: 334 Line: 11193 Section: <signal.h>
(2013 edition Page: 337 Line: 11278)

In the DESCRIPTION section, change from:

```
int    sigqueue(pid_t, int, const union sigval);
```

to:

```
int    sigqueue(pid_t, int, union sigval);
```

*Rationale*: Austin Group Defect Report(s) applied: 844. See http://austingroupbugs.net/view.php?id=844.

**Change Number: XBD/TC2/D4/0074** [536]

On Page: 335 Line: 11213 Section: <signal.h>

In the SEE ALSO section, add:

XCU *kill*

*Rationale*: Austin Group Defect Report(s) applied: 536. See http://austingroupbugs.net/view.php?id=536.

**Change Number: XBD/TC2/D4/0075** [663]

On Page: 355 Line: 11953 Section: <stdlib.h>
(2013 edition Page: 358 Line: 12042)

After:

{MB_CUR_MAX} Maximum number of bytes in a character specified by the current locale (category LC_CTYPE).

add a new sentence:

[CX]In the POSIX locale the value of {MB_CUR_MAX} shall be 1.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663. The intention was always that the POSIX locale should have an 8-bit-clean single-byte encoding. The omission of an explicit statement to that effect was an oversight.

**Change Number: XBD/TC2/D4/0076** [801]

On Page: 365 Line: 12336 Section: <stropts.h>
(2013 edition Page: 368 Line: 12425)

In the DESCRIPTION section, add a new paragraph before the text "The following shall be declared...":

The **<stropts.h>** header may also define macros for message types using names that start with M_.

*Rationale*: Austin Group Defect Report(s) applied: 801. See http://austingroupbugs.net/view.php?id=801.

**Change Number: XBD/TC2/D4/0077** [934]

On Page: 385 Line: 12911 Section: <sys/socket.h>
(2013 edition Page: 388 Line: 13000)

In the DESCRIPTION section, add a new paragraph:

The value of AF_UNSPEC shall be 0.

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934. Historically some applications initialize just the standard members of sockaddr_in, and some initialize the whole structure (using default initialization or memset()). There may be applications which do the latter and then rely on the zero value of the sin_family member being set to AF_UNSPEC. Therefore the standard should require that AF_UNSPEC has the value 0.

**Change Number: XBD/TC2/D4/0078** [531]

On Page: 390 Line: 13129 Section: <sys/stat.h>

At the end of the DESCRIPTION section, add:

Inclusion of the <**sys/stat.h**> header may make visible all symbols from the <**time.h**> header.

*Rationale*: Austin Group Defect Report(s) applied: 531. See http://austingroupbugs.net/view.php?id=531.

**Change Number: XBD/TC2/D4/0079** [856]

On Page: 399 Line: 13386 Section: <sys/types.h>
(2013 edition Page: 402 Line: 13477)

In the DESCRIPTION section, remove the XSI shading from the line:

**suseconds_t** Used for time in microseconds.

*Rationale*: Austin Group Defect Report(s) applied: 856. See http://austingroupbugs.net/view.php?id=856.

**Change Number: XBD/TC2/D4/0080** [659]

On Page: 399 Line: 13408 Section: <sys/types.h>

In the DESCRIPTION section:

Add **timer_t** to the list of exceptions to the arithmetic types requirement.

On Page: 400 Line: 13438 Section: <sys/types.h>

In the DESCRIPTION section:

Add **timer_t** to the list of types that have no comparison or assignment operators.

*Rationale*: Austin Group Defect Report(s) applied: 659. See http://austingroupbugs.net/view.php?id=659.

**Change Number: XBD/TC2/D4/0081** [934]

On Page: 403 Line: 13525 Section: <sys/un.h>
(2013 edition Page: 406 Line: 13620)

In the APPLICATION USAGE section, add a new paragraph:

Although applications are required to initialize all members (including any non-standard ones) of a **sockaddr_in6** structure (see **<netinet/in.h>**), the same is not required for the **sockaddr_un** structure, since historically many applications only initialized the standard members. Despite this, applications are encouraged to initialize **sockaddr_un** structures in a manner similar to the required initialization of **sockaddr_in6** structures.

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934.

**Change Number: XBD/TC2/D4/0082** [579]

On Page: 405 Line: 13598 Section: <sys/wait.h>

In the DESCRIPTION section, change from:

The <**sys/wait.h**> header shall define the **siginfo_t** type as described in <**signal.h**>.

to:

The <**sys/wait.h**> header shall define the **siginfo_t** type and the **sigval** union as described in <**signal.h**>.

*Rationale*: Austin Group Defect Report(s) applied: 579. See http://austingroupbugs.net/view.php?id=579.

**Change Number: XBD/TC2/D4/0083** [564]

On Page: 406 Line: 13621 Section: <sys/wait.h>

In the CHANGE HISTORY section, add a new paragraph:

The requirement for <**sys/wait.h**> to define the **rusage** structure as described in <**sys/resource.h**> is removed, and <**sys/wait.h**> is no longer allowed to make visible all symbols from <**sys/resource.h**>.

*Rationale*: Austin Group Defect Report(s) applied: 564. See http://austingroupbugs.net/view.php?id=564.

**Change Number: XBD/TC2/D4/0084** [707]

On Page: 409 Line: 13695-13698 Section: <tar.h>
(2013 edition Page: 412 Line: 13791-13794)

In the DESCRIPTION section, replace the entries in the "General Definitions" table with:

| | | |
|---|---|---|
| TMAGIC | `"ustar"` | Used in the *magic* field in the **ustar** header block, including the trailing null byte |
| TMAGLEN | `6` | Length in octets of the *magic* field |
| TVERSION | `"00"` | Used in the *version* field in the **ustar** header block, excluding the trailing null byte |
| TVERSLEN | `2` | Length in octets of the *version* field |

*Rationale*: Austin Group Defect Report(s) applied: 707. See http://austingroupbugs.net/view.php?id=707.

**Change Number: XBD/TC2/D4/0085** [783]

On Page: 431 Line: 14489 Section: <unistd.h>

In the DESCRIPTION section:

Add IP6 margin marker and shading to _POSIX_IPV6 (lines 14489-14492)

*Rationale*: Austin Group Defect Report(s) applied: 783. See http://austingroupbugs.net/view.php?id=783.

**Change Number: XBD/TC2/D4/0086** [911]

On Page: 437 Line: 14773 Section: <unistd.h>
(2013 edition Page: 440 Line: 14878)

In the DESCRIPTION section, change from:

This is the value for the *PATH* environment variable that finds all standard utilities.

to:

This is the value for the *PATH* environment variable that finds all of the standard utilities that are provided in a manner accessible via the *exec* family of functions.

*Rationale*: Austin Group Defect Report(s) applied: 911. See http://austingroupbugs.net/view.php?id=911.

**Change Number: XBD/TC2/D4/0087** [566]

On Page: 443 Line: 15037 Section: <unistd.h>

In the DESCRIPTION section, change from:

**intptr_t** type as described in <**inttypes.h**>.

to:

**intptr_t** type as described in <**stdint.h**>.

On Page: 445 Line: 15127 Section: <unistd.h>

In the DESCRIPTION section, add a new paragraph:

Inclusion of the <**unistd.h**> header may make visible all symbols from the headers <**stddef.h**>, <**stdint.h**>, and <**stdio.h**>.

*Rationale*: Austin Group Defect Report(s) applied: 566. See http://austingroupbugs.net/view.php?id=566.

**Change Number: XBD/TC2/D4/0088** [73]

On Page: 454 Line: 15481-15482 Section: <wchar.h>

In the DESCRIPTION section, change from:

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided for use with ISO C standard compilers.

to:

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided for use with ISO C standard compilers. Arguments to functions in this list can point to arrays containing **wchar_t** values that do not correspond to members of the character set of the current locale. Such values shall be processed according to the specified semantics, unless otherwise stated.

*Rationale*: Austin Group Defect Report(s) applied: 73. See http://austingroupbugs.net/view.php?id=73.

A clarification has been made in the C11 standard.

## 3. 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/TC2/D4/0001** [801]

On Page: 471 Line: 15977 Section: 2.2.2 The Name Space
(2013 edition Page: 475 Line: 16085)

Add an M_ prefix reservation for <math.h> in the table, shaded XSI.

*Rationale*: Austin Group Defect Report(s) applied: 801. See http://austingroupbugs.net/view.php?id=801.

**Change Number: XSH/TC2/D4/0002** [780]

On Page: 471 Line: 15991-15992 Section: 2.2.2 The Name Space

Add CX shading to the two unshaded lines in the **<signal.h>** table row.

*Rationale*: Austin Group Defect Report(s) applied: 780. See http://austingroupbugs.net/view.php?id=780.

**Change Number: XSH/TC2/D4/0003** [790]

On Page: 471 Line: 15997 Section: 2.2.2 The Name Space

Remove the <stdint.h> row from the table and append the following sentence to the small-font note after the table:

The C standard reserves int[0-9a-z_]*_t and uint[0-9a-z_]*_t in **<stdint.h>**; this is not included in the table above because it is covered by the reserved _t suffix for any header.

*Rationale*: Austin Group Defect Report(s) applied: 790. See http://austingroupbugs.net/view.php?id=790.
These patterns are redundant as _t is a reserved suffix.

**Change Number: XSH/TC2/D4/0004** [780]

On Page: 472 Line: 16025-16035 Section: 2.2.2 The Name Space

On lines 16025-16027 add CX shading to the **<time.h>** table row.

On line 16029 add OB shading to the **<utime.h>** table row.

On line 16035 add CX shading to the **ANY header** table row.

*Rationale*: Austin Group Defect Report(s) applied: 780. See http://austingroupbugs.net/view.php?id=780.

**Change Number: XSH/TC2/D4/0005** [790]

On Page: 472 Line: 16036 Section: 2.2.2 The Name Space

Change from:

The notation [A-Z] indicates any uppercase letter in the portable character set.

to:

The notation [0-9] indicates any digit. The notation [A-Z] indicates any uppercase letter in the portable character set.

*Rationale*: Austin Group Defect Report(s) applied: 790. See http://austingroupbugs.net/view.php?id=790.

**Change Number: XSH/TC2/D4/0006** [782]

On Page: 472 Line: 16038 Section: 2.2.2 The Name Space
(2013 edition Page: 476 Line: 16146)

Add a new paragraph (not as part of the preceding Note):

Implementations may also add symbols to the **<complex.h>** header with the following complete names or the same names suffixed with f or l:

| | | |
|---|---|---|
| cerf | cexpm1 | clog2 |
| cerfc | clog10 | clgamma |
| cexp2 | clog1p | ctgamma |

*Rationale*: Austin Group Defect Report(s) applied: 782. See http://austingroupbugs.net/view.php?id=782.

**Change Number: XSH/TC2/D4/0007** [790]

On Page: 473 Line: 16069 Section: 2.2.2 The Name Space

Change from:

The notation [0-9] indicates any digit. The notation [A-Z] indicates any uppercase letter in the portable character set. The notation [0-9a-z_] indicates any digit, any lowercase letter in the portable character set, or <underscore>.

to:

The notation [0-9] indicates any digit. The notation [A-Z] indicates any uppercase letter in the portable character set. The notation [Xa-z] indicates the character 'X' or any lowercase letter in the portable character set. The notation [0-9A-Za-z_]* indicates zero or more occurrences of any of the following: a digit, an uppercase or lowercase letter in the portable character set, or an <underscore>.

*Rationale*: Austin Group Defect Report(s) applied: 790. See http://austingroupbugs.net/view.php?id=790.

**Change Number: XSH/TC2/D4/0008** [790]

On Page: 475-476 Line: 16095-16178 Section: 2.2.2 The Name Space

In the table of reserved identifiers:

Change cexmp1 to cexpm1.
Change cexmp1f to cexpm1f.
Change cexmp1l to cexpm1l.
Change ltime to ctime.
Delete isblank and iswblank.
Delete strtof, strtoimax, strtold, strtoll, strtoull, and strtoumax.
Delete wcstof, wcstoimax, wcstold, wcstoll, wcstoull, and wcstoumax.
Delete wcwidth.
Delete wmem[a-z]*.

Add the following to the table of reserved identifiers in the correct alphabetical position:

btowc
ctanh
ctanhf
ctanhl
erf
erfc
hypot
lgamma
lldiv
math_errhandling
nextafter
remainder
snprintf
va_copy
vsnprintf
wmemchr
wmemcmp
wmemcpy
wmemmove
wmemset

Remove one of the two appearances of each of the following from the table of reserved identifiers:

acosl
asinl
atanf
atanh
atanl
catanh
catanhf
catanhl
ceilf
ceill
ldiv

On line 16178 add a new paragraph after the table:

<small>**Note:** The notation [a-z] indicates any lowercase letter in the portable character set. The notation * indicates any sequence of zero or more characters that are valid in identifiers with external linkage.</small>

*Rationale*: Austin Group Defect Report(s) applied: 790. See http://austingroupbugs.net/view.php?id=790.

**Change Number: XSH/TC2/D4/0009** [496]

On Page: 482 Line: 16415 Section: 2.3 Error Numbers

Change the description of [ENOSYS] from:

Function not implemented. An attempt was made to use a function that is not available in this implementation.

to:

Functionality not supported. An attempt was made to use optional functionality that is not supported in this implementation.

*Rationale*: Austin Group Defect Report(s) applied: 496. See http://austingroupbugs.net/view.php?id=496.

**Change Number: XSH/TC2/D4/0010** [681]

On Page: 482 Line: 16442 Section: 2.3 Error Numbers
(2013 edition Page: 486 Line: 16552)

Change from:

EOPNOTSUP

to:

EOPNOTSUPP

*Rationale*: Austin Group Defect Report(s) applied: 681. See http://austingroupbugs.net/view.php?id=681.

**Change Number: XSH/TC2/D4/0011** [690]

On Page: 487 Line: 16657 Section: 2.4.3 Signal Actions
(2013 edition Page: 491 Line: 16773-16779)

In the description of SIG_IGN, change from:

If a process sets the action for the SIGCHLD signal to SIG_IGN, the behavior is unspecified, [XSI]except as specified below.

If the action for the SIGCHLD signal is set to SIG_IGN, child processes of the calling processes shall not be transformed into zombie processes when they terminate. If the calling process subsequently waits for its

children, and the process has no unwaited-for children that were transformed into zombie processes, it shall block until all of its children terminate, and *wait*(), *waitid*(), and *waitpid*() shall fail and set *errno* to [ECHILD].[/XSI]

to:

If a process sets the action for the SIGCHLD signal to SIG_IGN, the behavior is unspecified[XSI], except as specified under "Consequences of Process Termination" in the description of the *_Exit*() function on page 549[/XSI].

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0012** [516]

On Page: 489 Line: 16721 Section: 2.4.3 Signal Concepts

Change from:

Therefore, applications can invoke them, without restriction, from signal-catching functions:

to:

Therefore, applications can call them, without restriction, from signal-catching functions. Note that, although there is no restriction on the calls themselves, for certain functions there are restrictions on subsequent behavior after the function is called from a signal-catching function (see [xref to *longjmp*()]).

At line 16722-16755 add *longjmp*() and *siglongjmp*() to the table of async-signal-safe functions.

*Rationale*: Austin Group Defect Report(s) applied: 516. See http://austingroupbugs.net/view.php?id=516.

**Change Number: XSH/TC2/D4/0013** [692]

On Page: 489 Line: 16722 Section: 2.4.3 Signal Actions
(2013 edition Page: 494 Line: 16845-16879)

In the table of async-signal-safe functions, add the following in proper sorted order:

*ffs*(), *htonl*(), *htons*(), *memccpy*(), *memchr*(), *memcmp*(), *memcpy*(), *memmove*(), *memset*(), *ntohl*(), *ntohs*(), *stpcpy*(), *stpncpy*(), *strcat*(), *strchr*(), *strcmp*(), *strcpy*(), *strcspn*(), *strlen*(), *strncat*(), *strncmp*(), *strncpy*(), *strnlen*(), *strpbrk*(), *strrchr*(), *strspn*(), *strstr*(), *strtok_r*(), *wcpcpy*(), *wcpncpy*(), *wcscat*(), *wcschr*(), *wcscmp*(), *wcscpy*(), *wcscspn*(), *wcslen*(), *wcsncat*(), *wcsncmp*(), *wcsncpy*(), *wcsnlen*(), *wcspbrk*(), *wcsrchr*(), *wcsspn*(), *wcsstr*(), *wcstok*(), *wmemchr*(), *wmemcmp*(), *wmemcpy*(), *wmemmove*(), *wmemset*()

*Rationale*: Austin Group Defect Report(s) applied: 692. See http://austingroupbugs.net/view.php?id=692. Since these functions do not modify or depend on global state, there are no known implementations where these functions cannot be safely used in aysnc-signal-safe code.

**Change Number: XSH/TC2/D4/0014** [615]

On Page: 489 Line: 16756 Section: 2.4.3 Signal Concepts

Change from:

All functions not in the above table are considered to be unsafe with respect to signals.

to:

Any function not in the above table may be unsafe with respect to signals. Implementations may make other interfaces async-signal-safe.

*Rationale*: Austin Group Defect Report(s) applied: 615. See http://austingroupbugs.net/view.php?id=615.

**Change Number: XSH/TC2/D4/0015** [516]

On Page: 490 Line: 16758 Section: 2.4.3 Signal Concepts

Change from:

... with a single exception: when a signal interrupts an unsafe function and the signal-catching function calls an unsafe function, the behavior is undefined.

to:

... with the exception that when a signal interrupts an unsafe function or equivalent (such as the processing equivalent to *exit*() performed after a return from the initial call to *main*()) and the signal-catching function calls an unsafe function, the behavior is undefined. Additional exceptions are specified in the descriptions of individual functions such as *longjmp*().

*Rationale*: Austin Group Defect Report(s) applied: 516. See http://austingroupbugs.net/view.php?id=516. The restrictions on using longjmp() and siglongjmp() are more restrictive than they need to be on POSIX systems. The loosened restrictions presented here do not break existing implementations and make it easier for application writers to create portable applications.

**Change Number: XSH/TC2/D4/0016** [807]

On Page: 490 Line: 16761 Section: 2.4.3 Signal Concepts

Change from:

Operations which obtain the value of *errno* and operations which assign a value to *errno* shall be async-signal-safe.

to:

Operations which obtain the value of *errno* and operations which assign a value to *errno* shall be async-signal-safe, provided that the signal-catching function saves the value of *errno* upon entry and restores it before it returns.

*Rationale*: Austin Group Defect Report(s) applied: 807. See http://austingroupbugs.net/view.php?id=807.

**Change Number: XSH/TC2/D4/0017** [807]

On Page: 490 Line: 16775 Section: 2.4.4 Signal Effects on Other Functions
(2013 edition Page: 495 Line: 16899)

Change from:

... except as noted for unsafe functions.

to:

... except as noted for unsafe functions. After returning from a signal-catching function, the value of *errno* is unspecified if the signal-catching function or any function it called assigned a value to *errno* and the signal-catching function did not save and restore the original value of *errno*.

*Rationale*: Austin Group Defect Report(s) applied: 807. See http://austingroupbugs.net/view.php?id=807.

**Change Number: XSH/TC2/D4/0018** [608]

On Page: 490 Line: 16788 Section: 2.5 Standard I/O Streams

Change from:

All input takes place as if bytes were read by successive calls to *fgetc*(); all output takes place as if bytes were written by successive calls to *fputc*().

to:

The wide character input functions shall read characters from the stream and convert them to wide characters as if they were read by successive calls to the *fgetwc*() function. Each conversion shall occur as if by a call to the *mbrtowc*() function, with the conversion state described by the stream's own **mbstate_t** object (see [xref to 2.5.2]). The byte input functions shall read characters from the stream as if by successive calls to the *fgetc*() function.

The wide character output functions shall convert wide characters to characters and write them to the stream as if they were written by successive calls to the *fputwc*() function. Each conversion shall occur as if by a call to the *wcrtomb*() function, with the conversion state described by the stream's own **mbstate_t** object (see [xref to 2.5.2]). The byte output functions shall write characters to the stream as if by successive calls to the *fputc*() function.

The *perror*(), *psiginfo*() and *psignal*() functions shall behave as described above for the byte output functions if the stream is already byte-oriented, and shall behave as described above for the wide character output functions if the stream is already wide-oriented. If the stream has no orientation, they shall behave as described for the byte output functions except that they shall not change the orientation of the stream.

Functions other than *perror*(), *psiginfo*() and *psignal*() that write to streams but are neither wide character output nor byte output functions (*getopt*() and *wordexp*()), shall behave as described above for the byte output functions, except that if the stream has no orientation, it is unspecified whether they set the stream to byte orientation or leave it with no orientation.

*Rationale*: Austin Group Defect Report(s) applied: 608. See http://austingroupbugs.net/view.php?id=608.

**Change Number: XSH/TC2/D4/0019** [480]

On Page: 491 Line: 16844 Section: 2.5.1 Interaction of File Descriptors and Standard I/O Streams

Change from:

A handle which is a stream is considered to be closed when either an *fclose*() or *freopen*() is executed on it (the result of *freopen*() is a new stream, which cannot be a handle on the same open file description as its previous value),

to:

A handle which is a stream is considered to be closed when either an *fclose*(), or *freopen*() with non-null filename, is executed on it (for *freopen*() with a null filename, it is implementation-defined whether a new handle is created or the existing one reused),

*Rationale*: Austin Group Defect Report(s) applied: 480. See http://austingroupbugs.net/view.php?id=480. The standard is contradictory on whether freopen() can reuse a file description when passed a null filename.

**Change Number: XSH/TC2/D4/0020** [631]

On Page: 507 Line: 17490-17492 Section: 2.9.1 Thread-Safety

Remove getc_unlocked(), getchar_unlocked(), putc_unlocked(), and putchar_unlocked() from the list of functions that need not be thread-safe.

*Rationale*: Austin Group Defect Report(s) applied: 631. See http://austingroupbugs.net/view.php?id=631.

**Change Number: XSH/TC2/D4/0021** [826]

On Page: 507 Line: 17498 Section: 2.9.1 Thread-Safety
(2013 edition Page: 512 Line: 17626)

Add setlocale() to the list of functions that need not be thread-safe.

*Rationale*: Austin Group Defect Report(s) applied: 826. See http://austingroupbugs.net/view.php?id=826.

**Change Number: XSH/TC2/D4/0022** [631]

On Page: 507 Line: 17512 Section: 2.9.1 Thread-Safety

Change from:

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

to:

The *wcrtomb*() and *wcsrtombs*() functions need not be thread-safe if passed a NULL *ps* argument. The *getc_unlocked*(), *getchar_unlocked*(), *putc_unlocked*(), and *putchar_unlocked*() functions need not be thread-safe unless the invoking thread owns the (**FILE ***) object accessed by the call, as is the case after a successful call to the *flockfile*() or *ftrylockfile*() functions.

*Rationale*: Austin Group Defect Report(s) applied: 631. See http://austingroupbugs.net/view.php?id=631.

**Change Number: XSH/TC2/D4/0023** [627]

On Page: 512 Line: 17704 Section: 2.9.5.2 Cancellation Points

Remove system() from the list of mandatory cancellation points.

*Rationale*: Austin Group Defect Report(s) applied: 627. See http://austingroupbugs.net/view.php?id=627. The system() function is not required to be thread-safe, yet the standard requires a cancellation point to occur when executing it.

**Change Number: XSH/TC2/D4/0024** [627,632]

On Page: 513-514 Line: 17715-17789 Section: 2.9.5.2 Cancellation Points

Remove the following functions (which are not required to be thread-safe) from the list of optional cancellation points:

| | | | |
|---|---|---|---|
| asctime() | endutxent() | getopt() | getutxid() |
| catgets() | ftw() | getprotobyname() | getutxline() |
| ctime() | getdate() | getprotobynumber() | localtime() |
| dbm_close() | getgrent() | getprotoent() | nftw() |
| dbm_delete() | getgrgid() | getpwent() | pututxline() |
| dbm_fetch() | getgrnam() | getpwnam() | readdir() |
| dbm_nextkey() | gethostent() | getpwuid() | setgrent() |
| dbm_open() | getlogin() | getservbyname() | setpwent() |
| dbm_store() | getnetbyaddr() | getservbyport() | setutxent() |
| endgrent() | getnetbyname() | getservent() | strerror() |
| endpwent() | getnetent() | getutxent() | ttyname() |

Remove pclose() from the list of optional cancellation points.

*Rationale*: Austin Group Defect Report(s) applied: 627,632. See http://austingroupbugs.net/view.php?id=627. See http://austingroupbugs.net/view.php?id=632.

**Change Number: XSH/TC2/D4/0025** [627]

On Page: 514 Line: 17789 Section: 2.9.5.2 Cancellation Points

Add a new paragraph after the list of optional cancellation points:

In addition, a cancellation point may occur when a thread is executing any function that this standard does not require to be thread-safe but the implementation documents as being thread-safe. If a thread is cancelled while executing a non-thread-safe function, the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 627. See http://austingroupbugs.net/view.php?id=627.

**Change Number: XSH/TC2/D4/0026** [632]

On Page: 514 Line: 17795 Section: 2.9.5.2 Cancellation Points

Change from:

Any such side-effects occur before any cancellation cleanup handlers are called.

to:

Any such side-effects occur before any cancellation cleanup handlers are called. For functions that are explicitly required not to return when interrupted (for example, *pclose*()), if a thread is canceled while executing the function, the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 632. See http://austingroupbugs.net/view.php?id=632.

**Change Number: XSH/TC2/D4/0027** [622]

On Page: 515 Line: 17840 Section: 2.9.5.4 Async-Cancel Safety

Add two paragraphs to the end of the section:

If a thread has asynchronous cancellation enabled and is cancelled during execution of a function that is not async-cancel-safe, the behavior is undefined.

If a thread has deferred cancellation enabled, a signal catching function is called in that thread during execution of a function that is not async-cancel-safe, and the signal catching function calls any function that is a cancellation point while a cancellation is pending for the thread, the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 622. See http://austingroupbugs.net/view.php?id=622.

**Change Number: XSH/TC2/D4/0028** [498]

On Page: 516 Line: 17869 Section: 2.9.7 Thread Interactions with Regular File Operations

Append the following text to the last paragraph:

The requirement on the *close*() function shall also apply whenever a file descriptor is successfully closed however caused (for example, as a consequence of calling *close*(), calling *dup2*(), or of process

termination).

*Rationale*: Austin Group Defect Report(s) applied: 498. See http://austingroupbugs.net/view.php?id=498.

**Change Number: XSH/TC2/D4/0029** [972]

On Page: 517 Line: 17899 Section: 2.9 Threads
(2013 edition Page: 522 Line: 18031)

Add a new section:

**2.9.9 Synchronization Object Copies and Alternative Mappings**

For barriers, condition variables, mutexes, and read-write locks, [TSH]if the *process-shared* attribute is set to PTHREAD_PROCESS_PRIVATE,[/TSH] only the synchronization object at the address used to initialize it can be used for performing synchronization. The effect of referring to another mapping of the same object when locking, unlocking, or destroying the object is undefined. [TSH]If the *process-shared* attribute is set to PTHREAD_PROCESS_SHARED, only the synchronization object itself can be used for performing synchronization; however, it need not be referenced at the address used to initalize it (that is, another mapping of the same object can be used).[/TSH] The effect of referring to a copy of the object when locking, unlocking, or destroying it is undefined.

For spin locks, the above requirements shall apply as if spin locks have a *process-shared* attribute that is set from the *pshared* argument to *pthread_spin_init*(). For semaphores, the above requirements shall apply as if semaphores have a *process-shared* attribute that is set to PTHREAD_PROCESS_PRIVATE if the *pshared* argument to *sem_init*() is zero and set to PTHREAD_PROCESS_SHARED if *pshared* is non-zero.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.
Most implementations of these private barriers, condition variables, mutexes, reader-writer locks, semaphores and spin locks use the address of the object to locate the object when using PTHREAD_PROCESS_PRIVATE (or equivalent pshared argument). The optimizations that can be performed when an object is not shared with other processes disappear if multiple mappings of one of these objects are allowed in this case. (If a process needs multiple mappings of an object, it can use PTHREAD_PROCESS_SHARED instead of PTHREAD_PROCESS_PRIVATE to get the desired behavior.)

**Change Number: XSH/TC2/D4/0030** [733]

On Page: 541 Line: 18857 Section: 2.12.1 Defined Types
(2013 edition Page: 546 Line: 18989)

In the description of sig_atomic_t, change from:

Integer type ...

to:

Possibly volatile-qualified integer type ...

*Rationale*: Austin Group Defect Report(s) applied: 733. See http://austingroupbugs.net/view.php?id=733.

**Change Number: XSH/TC2/D4/0031** [690]

On Page: 541 Line: 18886 Section: 2 General Information
(2013 edition Page: 546 Line: 19012)

Add new section:

**2.13 Status Information**

Status information is data associated with a process detailing a change in the state of the process. It shall consist of:

- The state the process transitioned into ('stopped', 'continued', or 'terminated').
- The information necessary to populate the **siginfo_t** structure provided by *waitid*().
- If the new state is 'terminated':
  - The low-order 8 bits of the status argument that the process passed to *_Exit*(), *_exit*(), or *exit*(), or the low-order 8 bits of the value the process returned from *main*(). Note that these 8 bits are part of the complete value that is used to set the *si_status* member of the **siginfo_t** structure provided by *waitid*().
  - Whether the process terminated due to the receipt of a signal that was not caught, and if so, the number of the signal that caused the termination of the process.
- If the new state is 'stopped':
  - The number of the signal that caused the process to stop.

A process might not have any status information (such as immediately after a process has started).

Status information for a process shall be generated (made available to the parent process) when the process stops, continues, or terminates except in the following case:

- If the parent process sets the action for the SIGCHLD signal to SIG_IGN, or if the parent sets the SA_NOCLDWAIT flag for the SIGCHLD signal action, process termination shall not generate new status information but shall cause any existing status information for the process to be discarded.

If new status information is generated, and the process already had status information, the existing status information shall be discarded and replaced with the new status information.

Only the process's parent process can obtain the process's status information. The parent obtains a child's status information by calling *wait*(), *waitid*(), or *waitpid*().

Except when *waitid*() is called with the WNOWAIT flag set in the options argument, the status information obtained by a wait function shall be consumed (discarded) by that wait function; no two calls to *wait*(), *waitid*() (without WNOWAIT), or *waitpid*() shall obtain the same status information.

When status information becomes available to the parent process and more than one thread in the parent process is waiting for the status information (blocked in a call to *wait*(), *waitid*(), or *waitpid*() with arguments that would match the status information):

- If none of the matching threads is in a call to *waitid*() with the WNOWAIT flag set in the options argument, the thread that obtains the status information is unspecified.
- Otherwise (at least one of the matching threads is in a call to *waitid*() with the WNOWAIT flag set), the matching thread or threads that obtain the status information is unspecified except that at least one of the matching threads shall obtain the status information and at most one of the

matching threads that are not calling *waitid*() with the WNOWAIT flag set shall obtain it.

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0032** [835]

On Page: 541 Line: 18886 Section: 2 General Information

Add new section:

**2.14 File Descriptor Allocation**

All functions that open one or more file descriptors shall, unless specified otherwise, atomically allocate the lowest numbered available (that is, not already open in the calling process) file descriptor at the time of each allocation. Where a single function allocates two file descriptors (for example *pipe*() or *socketpair*()), the allocations may be independent and therefore applications should not expect them to have adjacent values or depend on which has the higher value.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0033** [594]

On Page: 545 Line: 18913 Section: _Exit()

In the DESCRIPTION section, change from:

the least significant 8 bits (that is, *status* & 0377) shall be available to a waiting parent process

to:

the least significant 8 bits (that is, *status* & 0377) shall be available from *wait*() and *waitpid*(); the full value shall be available from *waitid*() and in the **siginfo_t** passed to a signal handler for SIGCHLD.

*Rationale*: Austin Group Defect Report(s) applied: 594. See http://austingroupbugs.net/view.php?id=594.

**Change Number: XSH/TC2/D4/0034** [594,690]

On Page: 545 Line: 18925 Section: _Exit()
(2013 edition Page: 549 Line: 19051)

At "Consequences of Process Termination" change from:

- If the parent process of the calling process is executing a *wait*(), *waitid*(), or *waitpid*(), [XSI]and has neither set its SA_NOCLDWAIT flag nor set SIGCHLD to SIG_IGN,[/XSI] it shall be notified of termination of the calling process and the low-order eight bits (that is, *bits* 0377) of *status* shall be made available to it. If the parent is not waiting, the child's status shall be made available to it when the parent subsequently executes *wait*(), *waitid*(), or *waitpid*().

  The semantics of the *waitid*() function shall be equivalent to *wait*().

- If the parent process of the calling process is not executing a *wait*(), *waitid*(), or *waitpid*(),

[XSI]and has neither set its SA_NOCLDWAIT flag nor set SIGCHLD to SIG_IGN,[/XSI] the calling process shall be transformed into a zombie process. A zombie process is an inactive process and it shall be deleted at some later time when its parent process executes *wait*(), *waitid*(), or *waitpid*().

[XSI]The semantics of the *waitid*() function shall be equivalent to *wait*().[/XSI]

- Termination of a process does not directly terminate its children. The sending of a SIGHUP signal as described below indirectly terminates children in some circumstances.

- Either:

  If the implementation supports the SIGCHLD signal, a SIGCHLD shall be sent to the parent process.

  Or:

  [XSI]If the parent process has set its SA_NOCLDWAIT flag, or set SIGCHLD to SIG_IGN, the status shall be discarded, and the lifetime of the calling process shall end immediately. If SA_NOCLDWAIT is set, it is implementation-defined whether a SIGCHLD signal is sent to the parent process.[/XSI]

to:

- [XSI]If the parent process of the calling process has set its SA_NOCLDWAIT flag or has set the action for the SIGCHLD signal to SIG_IGN:
  - The process's status information (see XSH Section 2.13), if any, shall be discarded.
  - The lifetime of the calling process shall end immediately. If SA_NOCLDWAIT is set, it is implementation-defined whether a SIGCHLD signal is sent to the parent process.
  - If a thread in the parent process of the calling process is blocked in *wait*(), *waitpid*(), or *waitid*(), and the parent process has no remaining child processes in the set of waited-for children, the *wait*(), *waitid*(), or *waitpid*() function shall fail and set *errno* to [ECHILD].

  Otherwise:[/XSI]
  - Status information (see XSH Section 2.13) shall be generated.
  - The calling process shall be transformed into a zombie process. Its status information shall be made available to the parent process until the process's lifetime ends.
  - The process's lifetime shall end once its parent obtains the process's status information via a currently-blocked or future call to *wait*(), *waitid*() (without WNOWAIT), or *waitpid*().
  - If one or more threads in the parent process of the calling process is blocked in a call to *wait*(), *waitid*(), or *waitpid*() awaiting termination of the process, one (or, if any are calling *waitid*() with WNOWAIT, possibly more) of these threads shall obtain the process's status information as specified in XSH Section 2.13 and become unblocked.
  - A SIGCHLD shall be sent to the parent process.
- Termination of a process does not directly terminate its children. The sending of a SIGHUP signal as described below indirectly terminates children in some circumstances.

*Rationale*: Austin Group Defect Report(s) applied: 594,690. See http://austingroupbugs.net/view.php?id=594.
See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0035** [835]

On Page: 559 Line: 19374 Section: accept()
(2013 edition Page: 563 Line: 19500)

After the text:

... and allocate a new file descriptor for that socket.

add:

The file descriptor shall be allocated as described in [xref to new section 2.14].

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0036** [836]

On Page: 559 Line: 19398 Section: accept()

In the RETURN VALUE section, change from:

Otherwise, -1 shall be returned and *errno* set to indicate the error.

to:

Otherwise, -1 shall be returned, *errno* shall be set to indicate the error, and any object pointed to by *address_len* shall remain unchanged.

*Rationale*: Austin Group Defect Report(s) applied: 836. See http://austingroupbugs.net/view.php?id=836.

**Change Number: XSH/TC2/D4/0037** [873]

On Page: 561 Line: 19442 Section: access()
(2013 edition Page: 565 Line: 19569)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0038** [591]

On Page: 561 Line: 19444 Section: access()

Before the faccessat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0039** [838]

On Page: 561 Line: 19449 Section: access()
(2013 edition Page: 565 Line: 19576)

In the DESCRIPTION section, change from:

... for accessibility according to the bit pattern contained in *amode*, using the real user ID in place of the effective user ID and the real group ID in place of the effective group ID.

to:

... for accessibility according to the bit pattern contained in *amode*. The checks for accessibility (including directory permissions checked during pathname resolution) shall be performed using the real user ID in place of the effective user ID and the real group ID in place of the effective group ID.

On Page: 561 Line: 19457 Section: access()
(2013 edition Page: 565 Line: 19584)

In the DESCRIPTION section, change from:

The *faccessat*() function shall be equivalent to the *access*() function, except ...

to:

The *faccessat*() function when called with a *flag* value of zero shall be equivalent to the *access*() function, except ...

*Rationale*: Austin Group Defect Report(s) applied: 838. See http://austingroupbugs.net/view.php?id=838.

**Change Number: XSH/TC2/D4/0040** [817]

On Page: 561 Line: 19460 Section: access()
(2013 edition Page: 565 Line: 19587)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0041** [487]

On Page: 561 Line: 19464 Section: access()

In the DESCRIPTION section, change from:

If *faccessat*() is passed the special value AT_FDCWD in the *fd* parameter, the current working directory is used and the behavior shall be identical to a call to *access*().

to:

If *faccessat*() is passed the special value AT_FDCWD in the *fd* parameter, the current working directory shall be used and, if *flag* is zero, the behavior shall be identical to a call to *access*().

*Rationale*: Austin Group Defect Report(s) applied: 487. See http://austingroupbugs.net/view.php?id=487.

**Change Number: XSH/TC2/D4/0042** [838]

On Page: 561 Line: 19468 Section: access()
(2013 edition Page: 565 Line: 19595)

In the DESCRIPTION section, for AT_EACCESS change from:

The checks for accessibility are performed using the effective user and group IDs instead of the real user and group ID

to:

The checks for accessibility (including directory permissions checked during pathname resolution) shall be performed using the effective user ID and group ID instead of the real user ID and group ID

*Rationale*: Austin Group Defect Report(s) applied: 838. See http://austingroupbugs.net/view.php?id=838.

**Change Number: XSH/TC2/D4/0043** [817]

On Page: 562 Line: 19488 Section: access()
(2013 edition Page: 566 Line: 19616)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0044** [838]

On Page: 562 Line: 19515 Section: access()
(2013 edition Page: 566 Line: 19643)

In the APPLICATION USAGE section, add the following paragraphs to the beginning of the section:

Use of these functions is discouraged since by the time the returned information is acted upon, it is out of date. (That is, acting upon the information always leads to a time-of-check-to-time-of-use race condition.) An application should instead attempt the action itself and handle the EACCES error that occurs if the file is not accessible (with a change of effective user and group IDs beforehand, and perhaps a change back afterwards, in the case where *access*() or *faccessat*() without AT_EACCES would have been used.)

Historically, one of the uses of *access*() was in set-user-ID root programs to check whether the user running the program had access to a file. This relied on "super-user" privileges which were granted based on the effective user ID being zero, so that when *access*() used the real user ID to check accessibility those privileges were not taken into account. On newer systems where privileges can be assigned which have no association with user or group IDs, if a program with such privileges calls *access*(), the change of IDs has no effect on the privileges and therefore they are taken into account in the accessibility checks. Thus *access*() (and *faccessat*() with *flag* zero) cannot be used for this historical purpose in such programs. Likewise, if a system provides any additional or alternate file access control mechanisms that are not user ID based, they will still be taken into account.

If a relative pathname is used, no account is taken of whether the current directory (or the directory associated with the file descriptor *fd*) is accessible via any absolute pathname. Applications using *access*(), or *faccessat*() without AT_EACCES, may consequently act as if the file would be accessible to a user with the real user ID and group ID of the process when such a user would not in practice be able to access the file because access would be denied at some point above the current directory (or the directory associated with the file descriptor *fd*) in the file hierarchy.

If *access*() or *faccessat*() is used with W_OK to check for write access to a directory which has the S_ISVTX bit set, a return value indicating the directory is writable can be misleading since some operations on files in the directory would not be permitted based on the ownership of those files (see [xref to XBD 4.2]).

On Page: 563 Line: 19544 Section: access()
(2013 edition Page: 567 Line: 19672)

Change the FUTURE DIRECTIONS section from:

None.

to:

These functions may be formally deprecated (for example by shading them OB) in a future revision of this standard.

*Rationale*: Austin Group Defect Report(s) applied: 838. See http://austingroupbugs.net/view.php?id=838.

**Change Number: XSH/TC2/D4/0045** [671]

On Page: 573 Line: 19824 Section: aio_fsync()

In the ERRORS section, change from:

The *aio_fildes* member of the **aiocb** structure referenced by the *aiocbp* argument is not a valid file descriptor open for writing.

to:

The *aio_fildes* member of the **aiocb** structure referenced by the *aiocbp* argument is not a valid file descriptor.

On Page: 574 Line: 19973 Section: aio_fsync()

In the APPLICATION USAGE section, change from:

None.

to:

Note that even if the file descriptor is not open for writing, if there are any pending write requests on the underlying file, then that I/O will be completed prior to the return of a call to *aio_error*() or *aio_return*() indicating that the operation has completed.

*Rationale*: Austin Group Defect Report(s) applied: 671. See http://austingroupbugs.net/view.php?id=671. The access mode of the file descriptor does not affect whether there are pending I/O operations on the underlying file.

**Change Number: XSH/TC2/D4/0046** [892]

On Page: 613 Line: 20931 Section: atol()
(2013 edition Page: 618 Line: 21108)

In the SYNOPSIS section, change from:

```
long atol(const char *str);
```

to:

```
long atol(const char *nptr);
```

On Page: 613 Line: 20937 Section: atol()
(2013 edition Page: 618 Line: 21114)

In the DESCRIPTION section, change from:

The call *atol*(*str*) shall be equivalent to:

```
strtol(str, (char **)NULL, 10)
```

to:

Except as noted below, the call *atol*(*nptr*) shall be equivalent to:

```
strtol(nptr, (char **)NULL, 10)
```

On Page: 613 Line: 20939 section atol()
(2013 edition Page: 618 Line: 21116)

In the DESCRIPTION section, change from:

The call to *atoll*(*nptr*) shall be equivalent ...

to:

Except as noted below, the call to *atoll*(*nptr*) shall be equivalent ...

On Page: 613 Line: 20941 section atol()
(2013 edition Page: 618 Line: 21118)

In the DESCRIPTION section, change from:

except that the handling of errors may differ.

to:

The handling of errors may differ.

On Page 613 Line: 20950 section atol()
(2013 edition Page: 618 Line: 21127)

In the APPLICATION USAGE section, change from:

The *atol*( ) function is subsumed by *strtol*() but is retained because it is used extensively in existing code. If the number is not known to be in range, *strtol*() should be used because *atol*() is not required to perform any error checking.

to:

If the number is not known to be in range, *strtol*() or *strtoll*() should be used because *atol*() and *atoll*() are not required to perform any error checking.

*Rationale*: Austin Group Defect Report(s) applied: 892. See http://austingroupbugs.net/view.php?id=892.

**Change Number: XSH/TC2/D4/0047** [656]

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*(). The returned pointer might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0041 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P619, L21156-21158:

The returned pointer might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0048** [928]

On Page: 614 Line: 20992 Section: basename()
(2013 edition Page: 619 Line: 21170)

In the EXAMPLES section, change from:

```
char *name = "/usr/lib";
```

to:

```
char name[] = "/usr/lib";
```

*Rationale*: Austin Group Defect Report(s) applied: 928. See http://austingroupbugs.net/view.php?id=928.

**Change Number: XSH/TC2/D4/0049** [612]

On Page: 614 Line: 20996-21004 Section: basename()

In the EXAMPLES section change from:

**Sample Input and Output Strings for basename()**

In the following table, the input string is the value pointed to by *path*, and the output string is the return value of the *basename*() function.

| **Input String** | **Output String** |
|---|---|
| `"/usr/lib"` | `"lib"` |
| `"/usr/"` | `"usr"` |
| `"/"` | `"/"` |

| | |
|---|---|
| "///" | "/" |
| "//usr//lib//" | "lib" |

1897

1898 to:

1899 **Sample Input and Output Strings for the *basename*() and *dirname*() functions and the *basename* and**
1900 ***dirname* utilities**

| ***basename*() and *dirname*() functions *path* argument** | **string returned by *basename*()** | **string returned by *dirname*()** | ***basename* and *dirname* utilities *string* operand** | **output written by *basename* utility** | **output written by *dirname* utility** |
|---|---|---|---|---|---|
| "usr" | "usr" | "." | usr | usr | . |
| "usr/" | "usr" | "." | usr/ | usr | . |
| "" | "." | "." | "" | . or *empty string* | . |
| "/" | "/" | "/" | / | / | / |
| "//" | "/" or "//" | "/" or "//" | // | / or // | / or // |
| "///" | "/" | "/" | /// | / | / |
| "/usr/" | "usr" | "/" | /usr/ | usr | / |
| "/usr/lib" | "lib" | "/usr" | /usr/lib | lib | /usr |
| "//usr//lib//" | "lib" | "//usr" | //usr//lib// | lib | //usr |
| "/home//dwc//test" | "test" | "/home//dwc" | /home//dwc//test | test | /home//dwc |

1901

1902 *Rationale*: Austin Group Defect Report(s) applied: 612. See http://austingroupbugs.net/view.php?id=612.

1903 **Change Number: XSH/TC2/D4/0050** [822]

1904 On Page: 617 Line: 21086 Section: bind()
1905 (2013 edition Page: 622 Line: 21267)

1906 In the ERRORS section, change from:

1907 [ENOENT]

A component of the pathname does not name an existing file or the pathname is an empty string.

to:

[ENOENT]
A component of the path prefix of the pathname in *address* does not name an existing file or the pathname is an empty string.

[ENOENT] or [ENOTDIR]
The pathname in *address* contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If the pathname without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822.

This change layers upon XSH/TC1/D5/0043 [146], changing the last sentence of the change.

**Change Number: XSH/TC2/D4/0051** [756]

On Page: 620 Line: 21229 Section: bsearch()

In the RATIONALE section, change from:

```
((char *)p - (char *(base) % width == 0
```

to:

```
( (char *)p - (char *)base ) % width == 0
```

*Rationale*: Austin Group Defect Report(s) applied: 756. See http://austingroupbugs.net/view.php?id=756.

**Change Number: XSH/TC2/D4/0052** [663]

On Page: 622 Line: 21263 Section: btowc()
(2013 edition Page: 627 Line: 21451)

In the RETURN VALUE section, add a new sentence:

[CX]In the POSIX locale, *btowc*() shall not return WEOF if *c* has a value in the range 0 to 255 inclusive.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XSH/TC2/D4/0053** [526]

On Page: 627 Line: 21395,21399 Section: calloc()

In the DESCRIPTION section, at line 21395 change from:

If the size of the space requested is 0, the behavior is implementation-defined: the value returned shall be either a null pointer or a unique pointer.

to:

If the size of the space requested is 0, the behavior is implementation-defined: either a null pointer shall be returned, or the behavior shall be as if the size were some nonzero value, except that the behavior is undefined if the returned pointer is used to access an object.

In the RETURN VALUE section, at line 21399 change from:

either a null pointer or a unique pointer that can be successfully passed to *free*() shall be returned.

to:

either:

- a null pointer shall be returned [CX]and *errno* may be set to an implementation-defined value[/CX], or
- a pointer to the allocated space shall be returned. The application shall ensure that the pointer is not used to access an object.

*Rationale*: Austin Group Defect Report(s) applied: 526. See http://austingroupbugs.net/view.php?id=526.

**Change Number: XSH/TC2/D4/0054** [645]

On Page: 639 Line: 21683 Section: catopen()

In the DESCRIPTION section, change from:

complete name

to:

pathname

*Rationale*: Austin Group Defect Report(s) applied: 645. See http://austingroupbugs.net/view.php?id=645.

**Change Number: XSH/TC2/D4/0055** [497]

On Page: 639 Line: 21685 Section: catopen()

In the DESCRIPTION section, change from:

... on page 173). If *NLSPATH* exists in the environment ...

to:

... on page 173); if *NLSPATH* exists in the environment ...

*Rationale*: Austin Group Defect Report(s) applied: 497. See http://austingroupbugs.net/view.php?id=497.

**Change Number: XSH/TC2/D4/0056** [497]

On Page: 640 Line: 21734 Section: catopen()

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

To be sure that messages produced by an application running with appropriate privileges cannot be used by an attacker setting an unexpected value for *NLSPATH* in the environment to confuse a system administrator, such applications should use pathnames containing a '/' to get defined behavior when using *catopen*() to open a message catalog.

*Rationale*: Austin Group Defect Report(s) applied: 497. See http://austingroupbugs.net/view.php?id=497.

**Change Number: XSH/TC2/D4/0057** [873]

On Page: 655 Line: 22155 Section: chmod()
(2013 edition Page: 660 Line: 22337)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0058** [591]

On Page: 655 Line: 22157 Section: chmod()

Before the fchmodat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0059** [817]

On Page: 655 Line: 22178 Section: chmod()
(2013 edition Page: 660 Line: 22358)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0060** [817]

On Page: 656 Line: 22207 Section: chmod()
(2013 edition Page: 661 Line: 22388)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0061** [893]

On Page: 657 Line: 22256 Section: chmod()
(2013 edition Page: 662 Line: 22437)

In the EXAMPLES section, change from:

```
chmod("home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
status = stat("home/cnd/mod1", &buffer);
```

to:

```
chmod("/home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
status = stat("/home/cnd/mod1", &buffer);
```

*Rationale*: Austin Group Defect Report(s) applied: 893. See http://austingroupbugs.net/view.php?id=893.

**Change Number: XSH/TC2/D4/0062** [873]

On Page: 659 Line: 22310 Section: chown()
(2013 edition Page: 664 Line: 22494)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0063** [591]

On Page: 659 Line: 22312 Section: chown()

Before the fchownat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0064** [485]

On Page: 659 Line: 22337 Section: chown()

In the DESCRIPTION section, change from:

If owner or group is specified as (**uid_t**)-1 or (**gid_t**)-1, respectively, the corresponding ID of the file shall not be changed. If both owner and group are -1, the times need not be updated.

Upon successful completion, *chown*() shall mark for update the last file status change timestamp of the file.

to:

If owner or group is specified as (**uid_t**)-1 or (**gid_t**)-1, respectively, the corresponding ID of the file shall not be changed.

Upon successful completion, *chown*() shall mark for update the last file status change timestamp of the file, except that if owner is (**uid_t**)-1 and group is (**gid_t**)-1, the file status change timestamp need not be marked for update.

*Rationale*: Austin Group Defect Report(s) applied: 485. See http://austingroupbugs.net/view.php?id=485. The wording for chown when both arguments are -1 was difficult to follow.

**Change Number: XSH/TC2/D4/0065** [817]

On Page: 659 Line: 22343 Section: chown()
(2013 edition Page: 664 Line: 22527)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0066** [817]

On Page: 660 Line: 22376 Section: chown()
(2013 edition Page: 665 Line: 22562)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0067** [686]

On Page: 665 Line: 22541-22552 Section: clock()

In the APPLICATION USAGE section, on line 22541 insert a new first paragraph:

In programming environments where **clock_t** is a 32-bit integer type and CLOCKS_PER_SEC is one million, *clock*() will start failing in less than 36 minutes of processor time for signed **clock_t**, or 72 minutes for unsigned **clock_t**. Applications intended to be portable to such environments should use *times*() instead (or *clock_gettime*() with CLOCK_PROCESS_CPUTIME_ID, if supported).

In the APPLICATION USAGE section, on line 22545 delete:

The value returned by *clock*() may wrap around on some implementations. For example, on a machine with 32-bit values for **clock_t**, it wraps after 2147 seconds or 36 minutes.

In the SEE ALSO section, on line 22552 add *clock_getres*() and *times*() to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 686. See http://austingroupbugs.net/view.php?id=686.

**Change Number: XSH/TC2/D4/0068** [909]

On Page: 671 Line: 22759 Section: clock_nanosleep()
(2013 edition Page: 676 Line: 22954)

In the RETURN VALUE section, change from:

If the *rmtp* argument is NULL, the remaining time is not returned.

to:

The *rqtp* and *rmtp* arguments can point to the same object. If the *rmtp* argument is NULL, the remaining time is not returned.

*Rationale*: Austin Group Defect Report(s) applied: 909. See http://austingroupbugs.net/view.php?id=909.

**Change Number: XSH/TC2/D4/0069** [555]

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

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

Usage of *close*() on file descriptors STDIN_FILENO, STDOUT_FILENO or STDERR_FILENO should immediately be followed by an operation to reopen these file descriptors. Unexpected behavior will result if any of these file descriptors is left in a closed state (for example, an EBADF error from *perror*()) or if an unrelated *open*() or similar call later in the application accidentally allocates a file to one of these well-known file descriptors. Furthermore, a *close*() followed by a reopen operation (e.g. *open*(), *dup*() etc) is not atomic; *dup2*() should be used to change standard file descriptors.

*Rationale*: Austin Group Defect Report(s) applied: 555. See http://austingroupbugs.net/view.php?id=555.

**Change Number: XSH/TC2/D4/0070** [810]

On Page: 687 Line: 23229 Section: confstr()

In the DESCRIPTION section, change from:

the string stored in *buf* will contain the <space>-separated list of variable=value environment variable pairs required by the implementation to create a conforming environment, as described in the implementations' conformance documentation.

to:

the string stored in *buf* shall contain a <space>-separated list of the variable=value environment variable pairs an implementation requires as part of specifying a conforming environment, as described in the implementations' conformance documentation

*Rationale*: Austin Group Defect Report(s) applied: 810. See http://austingroupbugs.net/view.php?id=810.

**Change Number: XSH/TC2/D4/0071** [911]

On Page: 687 Line: 23234 Section: confstr()
(2013 edition Page: 692 Line: 23446)

In the DESCRIPTION section, change from:

can be used as a value of the *PATH* environment variable that accesses all of the standard utilities of POSIX.1-2008

to:

can be used as a value of the *PATH* environment variable that accesses all of the standard utilities of POSIX.1-2008 that are provided in a manner accessible via the *exec* family of functions

*Rationale*: Austin Group Defect Report(s) applied: 911. See http://austingroupbugs.net/view.php?id=911.

**Change Number: XSH/TC2/D4/0072** [630]

On Page: 696 Line: 23586 Section: cosh()

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 **double**, 710.5 < |*x*| implies that *cosh*(*x*) has overflowed.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

**Change Number: XSH/TC2/D4/0073** [899]

On Page: 705 Line: 23827 Section: crypt()
(2013 edition Page: 710 Line: 24043)

In the APPLICATION USAGE section, add a new paragraph:

Several implementations offer extensions via characters outside of the set specified for the *salt* argument for specifying alternative algorithms; while not portable, these extensions may offer better security. The use of *crypt*() for anything other than password hashing is not recommended.

*Rationale*: Austin Group Defect Report(s) applied: 899. See http://austingroupbugs.net/view.php?id=899.

**Change Number: XSH/TC2/D4/0074** [656]

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 an 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*(). The returned pointer might also be invalidated if the calling thread is terminated. If *s* is not a null pointer, ...

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0065 [75,428].

The change is to add the following text to the end of the paragraph at 2013 edition P718, L24231-24234:

The returned pointer might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0075** [664]

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:

If *ctime_r*() sets *tzname*, it shall also set *daylight* and *timezone*. If *ctime_r*() does not set *tzname*, it shall not set *daylight* and shall not set *timezone*.

*Rationale*: Austin Group Defect Report(s) applied: 664. See http://austingroupbugs.net/view.php?id=664. This change is layered upon XSH/TC1/D5/0066 [321,428]. The change replaces the last paragraph of the description in 2013 edition, page 720 lines 24308-24309. The change is made for consistent behavior.

**Change Number: XSH/TC2/D4/0076** [572]

On Page: 723 Line: 24333-24354 Section: dirfd()

In the DESCRIPTION section, at line 24333 change from:

and may set *errno*

to:

and shall set *errno*

In the ERRORS section, at line 24388 delete the ENOTSUP error.

In the RATIONALE section, at line 24354 delete:

An implementation that does not support file descriptors referring to directories may fail with [ENOTSUP].

*Rationale*: Austin Group Defect Report(s) applied: 572. See http://austingroupbugs.net/view.php?id=572. All implementations must support a file descriptor referring to a directory, even if that descriptor is not used by readdir(). Furthermore, failure of dirfd() without setting errno is not useful to applications.

**Change Number: XSH/TC2/D4/0077** [830]

On Page: 725 Line: 24372 Section: dirname()
(2013 edition Page: 730 Line: 24593)

In the DESCRIPTION section, after the text:

... return a pointer to a string that is a pathname of the parent directory of that file.

add a sentence:

The *dirname*() function shall not perform pathname resolution; the result shall not be affected by whether or not *path* exists or by its file type.

*Rationale*: Austin Group Defect Report(s) applied: 830. See http://austingroupbugs.net/view.php?id=830. There is no requirement that the strings processed by basename() and dirname() refer to existing pathnames.

**Change Number: XSH/TC2/D4/0078** [612]

On Page: 725 Line: 24372,24375 Section: dirname()

In the DESCRIPTION section, at line 24372 change from:

Trailing '/' characters in the path are not counted as part of the path.

to:

Trailing '/' characters in the path that are not also leading '/' characters shall not be counted as part of the path.

Add a new paragraph after line 24375 (after paragraph two):

The *dirname*() 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 *dirname*().

*Rationale*: Austin Group Defect Report(s) applied: 612. See http://austingroupbugs.net/view.php?id=612.

**Change Number: XSH/TC2/D4/0079** [830]

On Page: 725 Line: 24378 Section: dirname()
(2013 edition Page: 730 Line: 24599)

In the RETURN VALUE section, change from:

The *dirname*() function shall return a pointer to a string that is the parent directory of *path*. If *path* is a null pointer or points to an empty string, a pointer to a string "." is returned.

to:

The *dirname*() function shall return a pointer to a string as described above.

*Rationale*: Austin Group Defect Report(s) applied: 830. See http://austingroupbugs.net/view.php?id=830.

**Change Number: XSH/TC2/D4/0080** [656]

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

In the RETURN VALUE section, change from:

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

to:

The *dirname*() 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 *dirname*(). The returned pointer might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0068 [75]

The change is to add the following text to the end of the paragraph at 2013 edition P730, L24601-24603:

The returned pointer might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0081** [612]

On Page: 726 Line: 24404-24413 Section: dirname()

In the EXAMPLES section, replace lines 24404-24413 with:

The EXAMPLES section of the *basename*() function (see XREF basename() EXAMPLES section on page XXX) includes a table showing examples of the results of processing several sample pathnames by the *basename*() and *dirname*() functions and by the *basename* and *dirname* utilities.

*Rationale*: Austin Group Defect Report(s) applied: 612. See http://austingroupbugs.net/view.php?id=612.

**Change Number: XSH/TC2/D4/0082** [656]

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

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

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 *dlerror*() in the same thread (if *dlerror*() is thread-safe) or in any thread (if *dlerror*() is not thread-safe). The returned pointer might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0070 [75] impacting the first part of that change.

The change is to add the following text to the end of the paragraph at 2013 edition P735, L24774-24776:

The returned pointer might also be invalidated if the calling thread is terminated.

Note that the deletion of text on L24555 in XSH/TC1/D5/0070 is not impacted by this change.

**Change Number: XSH/TC2/D4/0083** [743]

On Page: 739 Line: 24845 Section: drand48()
(2013 edition Page: 744 Line: 25096)

In the APPLICATION USAGE section, change from:

None.

to:

These functions should be avoided whenever non-trivial requirements (including safety) have to be fulfilled.

On Page: 739 Line: 24851 Section: drand48()
(2013 edition Page: 744 Line: 25102)

In the SEE ALSO section, add *random*().

*Rationale*: Austin Group Defect Report(s) applied: 743. See http://austingroupbugs.net/view.php?id=743.

**Change Number: XSH/TC2/D4/0084** [753]

On Page: 743 Line: 24972 Section: duplocale()

In the EXAMPLES section, change from:

```
return retval; }
```

to:

```
return retval;
```

*Rationale*: Austin Group Defect Report(s) applied: 753. See http://austingroupbugs.net/view.php?id=753.

**Change Number: XSH/TC2/D4/0085** [899]

On Page: 745 Line: 25023 Section: encrypt()
(2013 edition Page: 750 Line: 25291)

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of the standard may mark this interface as obsolete or remove it altogether.

*Rationale*: Austin Group Defect Report(s) applied: 899. See http://austingroupbugs.net/view.php?id=899.

**Change Number: XSH/TC2/D4/0086** [493]

On Page: 747 Line: 25044-25057 Section: endgrent()

In the DESCRIPTION section, at line 25044 change from:

When first called, *getgrent*() shall return a pointer to a **group** structure containing the first entry in the group database.

to:

If the group database is not already open, *getgrent*() shall open it and return a pointer to a **group** structure containing the first entry in the database.

In the DESCRIPTION section, at line 25053 change from:

The *setgrent*() function shall rewind the group database to allow repeated searches.

to:

The *setgrent*() function shall rewind the group database so that the next *getgrent*() call returns the first entry, allowing repeated searches.

In the DESCRIPTION section, at line 25054 change from:

The *endgrent*() function may be called to close the group database when processing is complete.

to:

The *endgrent*() function shall close the group database.

The *setgrent*() and *endgrent*() functions shall not change the setting of *errno* if successful.

On error, the *setgrent*() and *endgrent*() functions shall set *errno* to indicate the error.

Since no value is returned by the *setgrent*() and *endgrent*() functions, an application wishing to check for error situations should set *errno* to 0, then call the function, then check *errno*.

In the RETURN VALUE section, at line 25057 change from:

When first called, *getgrent*() shall return a pointer to the first **group** structure in the group database. Upon subsequent calls it shall return the next **group** structure in the group database. The *getgrent*() function shall return a null pointer on end-of-file or an error and *errno* may be set to indicate the error.

to:

On successful completion, *getgrent*() shall return a pointer to a **group** structure. On end-of-file, *getgrent*() shall return a null pointer and shall not change the setting of *errno*. On error, *getgrent*() shall return a null pointer and *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 493. See http://austingroupbugs.net/view.php?id=493. The changes are made for consistency.

**Change Number: XSH/TC2/D4/0087** [656]

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

In the RETURN VALUE section, change from:

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

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 *getgrgid*(), *getgrnam*(), or *getgrent*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.

This item is a layered change on XSH/TC1/D5/0080 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P752 L25329-25332:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0088** [493]

On Page: 747 Line: 25064 Section: endgrent()

In the ERRORS section, change from:

The *getgrent*() function may fail if:

[EINTR] A signal was caught during the operation.

[EIO] An I/O error has occurred.

to:

These functions may fail if:

[EINTR] A signal was caught during the operation.

[EIO] An I/O error has occurred.

In addition, the *getgrent*() and *setgrent*() functions may fail if:

*Rationale*: Austin Group Defect Report(s) applied: 493. See http://austingroupbugs.net/view.php?id=493.
The changes are made for consistency.

**Change Number: XSH/TC2/D4/0089** [656]

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

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 *gethostent*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0081 [75,428].

The change is to add the following text to the end of the paragraph at 2013 edition P754, L25393-25396:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0090** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0083 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P756, L25452-25455:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0091** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0085 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P758, L25512-25515:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0092** [493]

On Page: 755 Line: 25259-25271 Section: endpwent()

In the DESCRIPTION section, at line 25259 change from:

When first called, *getpwent*() shall return a pointer to a **passwd** structure containing the first entry in the user database.

to:

If the user database is not already open, *getpwent*() shall open it and return a pointer to a **passwd** structure containing the first entry in the database.

In the DESCRIPTION section, at line 25267 change from:

The *setpwent*() function effectively rewinds the user database to allow repeated searches.

to:

The *setpwent*() function shall rewind the user database so that the next *getpwent*() call returns the first entry, allowing repeated searches.

In the DESCRIPTION section, at line 25268 change from:

The *endpwent*() function may be called to close the user database when processing is complete.

to:

The *endpwent*() function shall close the user database.

The *setpwent*() and *endpwent*() functions shall not change the setting of *errno* if successful.

On error, the *setpwent*() and *endpwent*() functions shall set *errno* to indicate the error.

Since no value is returned by the *setpwent*() and *endpwent*() functions, an application wishing to check for error situations should set *errno* to 0, then call the function, then check *errno*.

In the RETURN VALUE section, at line 25271 change from:

The *getpwent*() function shall return a null pointer on end-of-file or error.

to:

On successful completion, *getpwent*() shall return a pointer to a **passwd** structure. On end-of-file, *getpwent*() shall return a null pointer and shall not change the setting of *errno*. On error, *getpwent*() shall return a null pointer and *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 493. See http://austingroupbugs.net/view.php?id=493. The changes are made for consistency.

**Change Number: XSH/TC2/D4/0093** [656]

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

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 *getpwuid*(), *getpwnam*(), or *getpwent*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0087 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P760, L25558-25561:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

Note that the deletion of text on L25278 in XSH/TC1/D5/0087 is not impacted by this change.

**Change Number: XSH/TC2/D4/0094** [493]

On Page: 755 Line: 25274 Section: endpwent()

In the ERRORS section, insert before the EIO error:

[EINTR] A signal was caught during the operation.

*Rationale*: Austin Group Defect Report(s) applied: 493. See http://austingroupbugs.net/view.php?id=493. The changes are made for consistency.

**Change Number: XSH/TC2/D4/0095** [656]

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

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 *getservbyname*(), *getservbyport*(), or *getservent*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0088 [75].

The change is to add the following text to the end of the paragraph at 2013 edition, P763, L25652-25655:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0096** [630]

On Page: 767 Line: 25598 Section: erfc()

In the APPLICATION USAGE section, delete:

Note for IEEE Std 754-1985 **double**, 26.55 < *x* implies *erfc*(*x*) has underflowed.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

**Change Number: XSH/TC2/D4/0097** [584]

On Page: 780 Line: 26023 Section: exec

In the RATIONALE section, change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0098** [898]

On Page: 780 Line: 26024 Section: exec
(2013 edition Page: 785 Line: 26351)

In the RATIONALE section, add a new paragraph:

Also, note that the *test* and *[* utilities require specific strings for the *argv[0]* argument to have deterministic behavior across all implementations.

*Rationale*: Austin Group Defect Report(s) applied: 898. See http://austingroupbugs.net/view.php?id=898.

**Change Number: XSH/TC2/D4/0099** [734]

On Page: 782 Line: 26142 Section: exec
(2013 edition Page: 788 Line: 26469)

In the RATIONALE section, change from:

Alternatively, an function like *openat*()

to:

Alternatively, a function like *openat*()

*Rationale*: Austin Group Defect Report(s) applied: 734. See http://austingroupbugs.net/view.php?id=734.

**Change Number: XSH/TC2/D4/0100** [594]

On Page: 785 Line: 26213 Section: exit()

In the DESCRIPTION section, change from:

the least significant 8 bits (that is, *status* & 0377) shall be available to a waiting parent process

to:

the least significant 8 bits (that is, *status* & 0377) shall be available from *wait*() and *waitpid*(); the full value shall be available from *waitid*() and in the **siginfo_t** passed to a signal handler for SIGCHLD.

*Rationale*: Austin Group Defect Report(s) applied: 594. See http://austingroupbugs.net/view.php?id=594. The standard is unclear about when si_status contains an exit status and when it contains a signal.

**Change Number: XSH/TC2/D4/0101** [630]

On Page: 787 Line: 26297 Section: exp()

In the APPLICATION USAGE section, delete:

Note that for IEEE Std 754-1985 **double**, 709.8 < *x* implies *exp*(*x*) has overflowed. The value *x*< -708.4 implies *exp*(*x*) has underflowed.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

**Change Number: XSH/TC2/D4/0102** [630]

On Page: 789 Line: 26366 Section: exp2()

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 **double**, 1024 <= *x* implies *exp2*(*x*) has overflowed. The value *x*< -1022 implies *exp*(*x*) has underflowed.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

**Change Number: XSH/TC2/D4/0103** [630]

On Page: 791 Line: 26426 Section: expm1()

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 **double**, 709.8 < *x* implies *expm1*(*x*) has overflowed.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

**Change Number: XSH/TC2/D4/0104** [555]

On Page: 806 Line: 26839 Section: fclose()

In the APPLICATION USAGE section, change from:

None.

to:

Since after the call to *fclose*() any use of *stream* results in undefined behavior, *fclose*() should not be used on *stdin*, *stdout*, or *stderr* except immediately before process termination (see XBD 3.297 on page 81), so as to avoid triggering undefined behavior in other standard interfaces that rely on these streams. If there are

any *atexit*() handlers registered by the application, such a call to *fclose*() should not occur until the last handler is finishing. Once *fclose*() has been used to close *stdin*, *stdout*, or *stderr*, there is no standard way to reopen any of these streams.

Use of *freopen*() to change *stdin*, *stdout*, or *stderr* instead of closing them avoids the danger of a file unexpectedly being opened as one of the special file descriptors STDIN_FILENO, STDOUT_FILENO, or STDERR_FILENO at a later time in the application.

*Rationale*: Austin Group Defect Report(s) applied: 555. See http://austingroupbugs.net/view.php?id=555.

**Change Number: XSH/TC2/D4/0105** [835]

On Page: 807 Line: 26877 Section: fcntl()
(2013 edition Page: 813 Line: 27229)

Change from:

Return a new file descriptor which shall be the lowest numbered available (that is, not already open) file descriptor greater than or equal to the third argument, *arg*, taken as an integer of type **int**.

to:

Return a new file descriptor which shall be allocated as described in [xref to new section 2.14] except that it shall be the lowest numbered available file descriptor greater than or equal to the third argument, *arg*, taken as an integer of type **int**.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0106** [677]

On Page: 807 Line: 26911-26920 Section: fcntl()

In the DESCRIPTION section, replace the F_GETOWN and F_SETOWN descriptions with:

F_GETOWN

If *fildes* refers to a socket, get the process ID or process group ID specified to receive SIGURG signals when out-of-band data is available. Positive values shall indicate a process ID; negative values, other than -1, shall indicate a process group ID; the value zero shall indicate that no SIGURG signals are to be sent. If *fildes* does not refer to a socket, the results are unspecified.

F_SETOWN

If *fildes* refers to a socket, set the process ID or process group ID specified to receive SIGURG signals when out-of-band data is available, using the value of the third argument, *arg*, taken as type **int**. Positive values shall indicate a process ID; negative values, other than -1, shall indicate a process group ID; the value zero shall indicate that no SIGURG signals are to be sent. Each time a SIGURG signal is sent to the specified process or process group, permission checks equivalent to those performed by *kill*() shall be performed, as if *kill*() were called by a process with the same real user ID, effective user ID, and privileges that the process calling *fcntl*() has at the time of the call; if the *kill*() call would fail, no signal shall be sent. These permission checks may also be performed by the *fcntl*() call. If the process specified by *arg* later terminates, or the process group specified by *arg* later becomes empty, while still being specified to receive SIGURG signals when out-of-band data is available from *fildes*, then no signals shall be sent to any subsequently created process that has the same process ID or process group ID, regardless of permission; it is unspecified whether this is

achieved by the equivalent of an `fcntl(`*`fildes`*`, F_SETOWN, 0)` call at the time the process terminates or is waited for or the process group becomes empty, or by other means. If *fildes* does not refer to a socket, the results are unspecified.

*Rationale*: Austin Group Defect Report(s) applied: 677. See http://austingroupbugs.net/view.php?id=677.

**Change Number: XSH/TC2/D4/0107** [484]

On Page: 808 Line: 26923 Section: fcntl()

In the DESCRIPTION section, for F_GETLK change from:

Get the first lock which blocks ...

to:

Get any lock which blocks ...

*Rationale*: Austin Group Defect Report(s) applied: 484. See http://austingroupbugs.net/view.php?id=484. It was unclear whether "first lock" means a lock with the lowest start offset or a first acquired lock in a chronological order or even something completely different.

**Change Number: XSH/TC2/D4/0108** [675]

On Page: 810 Line: 27040 Section: fcntl()
(2013 edition Page: 816 Line: 27394)

In the ERRORS section add after the [EOVERFLOW] error:

[ESRCH]
The *cmd* argument is F_SETOWN and no process or process group can be found corresponding to that specified by *arg*.

*Rationale*: Austin Group Defect Report(s) applied: 675. See http://austingroupbugs.net/view.php?id=675.

**Change Number: XSH/TC2/D4/0109** [675,677]

On Page: 811 Line: 27044 Section: fcntl()

In the ERRORS section, add to the "may fail" errors:

[EINVAL]
The *cmd* argument is F_SETOWN and the value of the argument is not valid as a process or process group identifier.

[EPERM]
The *cmd* argument is F_SETOWN and the calling process does not have permission to send a SIGURG signal to any process specified by *arg*.

On Page: 812 Line: 27107 Section: fcntl()

In the APPLICATION USAGE section, add a new paragraph:

On systems which do not perform permission checks at the time of an *fcntl*() call with F_SETOWN, if the permission checks performed at the time the signal is sent disallow sending the signal to any process, the process that called *fcntl*() has no way of discovering that this has happened. A call to *kill*() with signal 0 can be used as a prior check of permissions, although this is no guarantee that permission will be granted at the time a signal is sent, since the target process(es) could change user IDs or privileges in the meantime.

On Page: 814 Line: 27172 Section: fcntl()

In the SEE ALSO section, add *kill*().

*Rationale*: Austin Group Defect Report(s) applied: 675,677. See http://austingroupbugs.net/view.php?id=675.
See http://austingroupbugs.net/view.php?id=677.

**Change Number: XSH/TC2/D4/0110** [501]

On Page: 815 Line: 27225 Section: fdatasync()

In the ERRORS section, change from:

The *fildes* argument is not a valid file descriptor open for writing.

to:

The *fildes* argument is not a valid file descriptor.

At line 27232 in the APPLICATION USAGE section, change from:

None.

to:

Note that even if the file descriptor is not open for writing, if there are any pending write requests on the underlying file, then that I/O will be completed prior to the return of *fdatasync*().

*Rationale*: Austin Group Defect Report(s) applied: 501. See http://austingroupbugs.net/view.php?id=501.
The access mode of the file descriptor does not affect whether there are pending I/O operations on the underlying file.

**Change Number: XSH/TC2/D4/0111** [543]

On Page: 834 Line: 27835 Section: feraiseexcept()

In the DESCRIPTION section, change from:

The order in which these floating-point exceptions are raised is unspecified.

to:

The order in which these floating-point exceptions are raised is unspecified, [MX]except that if the *excepts* argument represents IEC 60559 valid coincident floating-point exceptions for atomic operations (namely overflow and inexact, or underflow and inexact), then overflow or underflow shall be raised before inexact.[/MX]

*Rationale*: Austin Group Defect Report(s) applied: 543. See http://austingroupbugs.net/view.php?id=543.

**Change Number: XSH/TC2/D4/0112** [816]

On Page: 844 Line: 28021 Section: fflush()

In the DESCRIPTION section, moves lines 28021-28022 to become the final paragraph:

If *stream* is a null pointer, *fflush*() shall perform this flushing action on all streams for which the behavior is defined above.

*Rationale*: Austin Group Defect Report(s) applied: 816. See http://austingroupbugs.net/view.php?id=816. This makes it clear that fflush(NULL) must affect underlying positions of seekable fds associated with read streams.

**Change Number: XSH/TC2/D4/0113** [626]

On Page: 844 Line: 28023 Section: fflush()

In the DESCRIPTION section, change from:

For a stream open for reading,

to:

For a stream open for reading with an underlying file description,

*Rationale*: Austin Group Defect Report(s) applied: 626. See http://austingroupbugs.net/view.php?id=626.

**Change Number: XSH/TC2/D4/0114** [468]

On Page: 852 Line: 28283-28285 Section: fgets()

In the DESCRIPTION section, change from:

The *fgets*() function shall read bytes from *stream* into the array pointed to by *s*, until *n*-1 bytes are read, or a <newline> is read and transferred to *s*, or an end-of-file condition is encountered. The string is then terminated with a null byte.

to:

The *fgets*() function shall read bytes from *stream* into the array pointed to by *s* until *n*-1 bytes are read, or a <newline> is read and transferred to *s*, or an end-of-file condition is encountered. A null byte shall be written immediately after the last byte read into the array. If the end-of-file condition is encountered before any bytes are read, the contents of the array pointed to by *s* shall not be changed.

*Rationale*: Austin Group Defect Report(s) applied: 468. See http://austingroupbugs.net/view.php?id=468. This is a conflict with the C99 standard. All current implementations are thought to implement this as C99 requires.

**Change Number: XSH/TC2/D4/0115** [589]

On Page: 858 Line: 28462-28464 Section: fileno()

In the ERRORS section, replace the entire section with:

The *fileno*() function shall fail if:

[EBADF]
The stream is not associated with a file.

The *fileno*() function may fail if:

[EBADF]
The file descriptor underlying *stream* is not a valid file descriptor.

*Rationale*: Austin Group Defect Report(s) applied: 589. See http://austingroupbugs.net/view.php?id=589.

**Change Number: XSH/TC2/D4/0116** [611]

On Page: 859 Line: 28523 Section: flockfile()

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

A call to *exit*() can block until locked streams are unlocked because a thread having ownership of a (**FILE** *) object blocks all function calls that reference that (**FILE** *) object (except those with names ending in _unlocked) from other threads, including calls to *exit*().

*Rationale*: Austin Group Defect Report(s) applied: 611. See http://austingroupbugs.net/view.php?id=611.

**Change Number: XSH/TC2/D4/0117** [587]

On Page: 866 Line: 28763 Section: fmemopen()

In the DESCRIPTION section, change from:

for modes *a* and *a*+ the initial size shall be either the position of the first null byte in the buffer or the value of the *size* argument if no null byte is found

to:

for modes *a* and *a*+ the initial size shall be:

- zero, if *buf* is a null pointer
- the position of the first null byte in the buffer, if one is found
- the value of the *size* argument, if *buf* is not a null pointer and no null byte is found

*Rationale*: Austin Group Defect Report(s) applied: 587. See http://austingroupbugs.net/view.php?id=587.
The standard does not speak to the null pointer case, but existing practice handles this case consistently.

**Change Number: XSH/TC2/D4/0118** [586,818]

On Page: 867 Line: 28786,28790 Section: fmemopen()

In the ERRORS section, on line 28786 change from:

[EINVAL]
The *size* argument specifies a buffer size of zero.

to:

[EMFILE]
{STREAM_MAX} streams are currently open in the calling process.

On line 28790, add a new paragraph before the "may fail" ENOMEM error:

[EINVAL]
The *size* argument specifies a buffer size of zero and the implementation does not support this.

*Rationale*: Austin Group Defect Report(s) applied: 586,818. See
http://austingroupbugs.net/view.php?id=586.
See http://austingroupbugs.net/view.php?id=818.

**Change Number: XSH/TC2/D4/0119** [818]

On Page: 868 Line: 28284 Section: fmemopen()

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future revision of this standard may require support of zero length buffer streams explicitly.

*Rationale*: Austin Group Defect Report(s) applied: 818. See http://austingroupbugs.net/view.php?id=818.

**Change Number: XSH/TC2/D4/0120** [605]

On Page: 870 Line: 28887 Section: fmod()

In the RETURN VALUE section, change from:

If *x* or *y* is NaN, a NaN shall be returned.

to:

If *x* or *y* is NaN, a NaN shall be returned, and none of the conditions below shall be considered.

*Rationale*: Austin Group Defect Report(s) applied: 605. See http://austingroupbugs.net/view.php?id=605.

**Change Number: XSH/TC2/D4/0121** [806]

On Page: 875 Line: 29060 Section: fnmatch()

In the DESCRIPTION section, change from:

In particular, "\\" shall match a <backslash> in *string*.

to:

In particular, "\\" shall match a <backslash> in *string*. If *pattern* ends with an unescaped <backslash>, *fnmatch*() shall return a non-zero value (indicating either no match or an error).

*Rationale*: Austin Group Defect Report(s) applied: 806. See http://austingroupbugs.net/view.php?id=806. Existing practice is for a pattern ending in an unescaped <backslash> not to match anything, but it would be useful to allow implementations to diagnose this as user error.

**Change Number: XSH/TC2/D4/0122** [822]

On Page: 878 Line: 29180 Section: fopen()

In the ERRORS section, change from:

[ENOENT]
A component of *filename* does not name an existing file or *filename* is an empty string.

to:

[ENOENT]
The *mode* string begins with `'r'` and a component of *pathname* does not name an existing file, or mode begins with `'w'` or `'a'` and a component of the path prefix of *pathname* does not name an existing file, or *pathname* is an empty string.

[ENOENT] or [ENOTDIR]
The *pathname* argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *pathname* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822. This change layers upon XSH/TC1/D5/0157 [146,433] changing the last sentence of the change.

**Change Number: XSH/TC2/D4/0123** [858]

On Page: 882 Line: 29311 Section: fork()
(2013 edition Page: 889 Line: 29741)

In the DESCRIPTION section, delete:

Fork handlers may be established by means of the *pthread_atfork*() function in order to maintain

application invariants across *fork*() calls.

*Rationale*: Austin Group Defect Report(s) applied: 858. See http://austingroupbugs.net/view.php?id=858.

**Change Number: XSH/TC2/D4/0124** [651]

On Page: 888 Line: 29550-29551 Section: fpathconf()

In the ERRORS section, move the "shall fail" ELOOP error to the "may fail" part of the ERRORS section before the other ELOOP error at line 29558.

*Rationale*: Austin Group Defect Report(s) applied: 651. See http://austingroupbugs.net/view.php?id=651. This was inconsistent with the requirements for other pathconf() error conditions.

**Change Number: XSH/TC2/D4/0125** [651]

On Page: 889 Line: 29612 Section: fpathconf()

In the RATIONALE section, change from:

... not required to detect any of the errors except the meaning of [EINVAL] that indicates that the value of *name* is not valid for that variable.

to:

... not required to detect any of the errors except the meaning of [EINVAL] that indicates that the value of *name* is not valid for that variable, and the [EOVERFLOW] error that indicates the value to be returned is larger than {LONG_MAX}.

*Rationale*: Austin Group Defect Report(s) applied: 651. See http://austingroupbugs.net/view.php?id=651.

**Change Number: XSH/TC2/D4/0126** [894]

On Page: 895 Line: 29808 Section: fprintf()
(2013 edition Page: 902 Line: 30242)

In the DESCRIPTION section, change from:

For o conversion, it increases the precision (if necessary) to force the first digit of the result to be zero.

to:

For o conversion, it shall increase the precision, if and only if necessary, to force the first digit of the result to be a zero (if the value and precision are both 0, a single 0 is printed).

*Rationale*: Austin Group Defect Report(s) applied: 894. See http://austingroupbugs.net/view.php?id=894.

**Change Number: XSH/TC2/D4/0127** [557]

On Page: 899 Line: 30005 Section: fprintf()

In the ERRORS section, delete:

In addition, all forms of *fprintf*() may fail if:

[EINVAL] There are insufficient arguments.

In the RATIONALE section, on page 904 line 30225 change from:

None

to:

If an implementation detects that there are insufficient arguments for the format, it is recommended that the function should fail and report an [EINVAL] error.

*Rationale*: Austin Group Defect Report(s) applied: 557. See http://austingroupbugs.net/view.php?id=557. The descriptions of fprintf() and fwprintf() say "The results are undefined if there are insufficient arguments for the format." However, the ERRORS section on each of those pages also specifies a "may fail" EINVAL error for this condition.

**Change Number: XSH/TC2/D4/0128** [936]

On Page: 904 Line: 30230 Section: fprintf()
(2013 edition Page: 912 Line: 30677)

In the SEE ALSO section, add **<inttypes.h>** to the list.

*Rationale*: Austin Group Defect Report(s) applied: 936. See http://austingroupbugs.net/view.php?id=936.

**Change Number: XSH/TC2/D4/0129** [926]

On Page: 913 Line: 30537 Section: fread()
(2013 edition Page: 921 Line: 31022)

In the EXAMPLES section, change from:

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

to:

The following example transfers a single 100-byte fixed length record from the *fp* stream into the array pointed to by *buf*.

*Rationale*: Austin Group Defect Report(s) applied: 926. See http://austingroupbugs.net/view.php?id=926.

**Change Number: XSH/TC2/D4/0130** [939]

On Page: 916 Line: 30649 Section: freeaddrinfo()
(2013 edition Page: 924 Line: 31149)

In the DESCRIPTION section, change from:

In this *hints* structure every member other than *ai_flags*, *ai_family*, *ai_socktype*, and *ai_protocol* shall be set to zero or a null pointer.

to:

The application shall ensure that each of the *ai_addrlen*, *ai_addr*, *ai_canonname*, and *ai_next* members, as well as each of the non-standard additional members, if any, of this *hints* structure is initialized. If any of these members has a value other than the value that would result from default initialization, the behavior is implementation-defined.

*Rationale*: Austin Group Defect Report(s) applied: 939. See http://austingroupbugs.net/view.php?id=939.

**Change Number: XSH/TC2/D4/0131** [979]

On Page: 917 Line: 30683 Section: freeaddrinfo()
(2013 edition Page: 925 Line: 31183)

In the DESCRIPTION section, change from:

If the AI_V4MAPPED flag is specified along with an *ai_family* of AF_INET6 then *getaddrinfo*() shall return IPv4-mapped IPv6 addresses on finding no matching IPv6 addresses (*ai_addrlen* shall be 16). The AI_V4MAPPED flag shall be ignored unless *ai_family* equals AF_INET6. If the AI_ALL flag is used with the AI_V4MAPPED flag, then *getaddrinfo*() shall return all matching IPv6 and IPv4 addresses. The AI_ALL flag without the AI_V4MAPPED flag is ignored.

to (keeping the IP6 shading):

By default, with an *ai_family* of AF_INET6, *getaddrinfo*() shall return only IPv6 addresses. If the AI_V4MAPPED flag is specified along with an *ai_family* of AF_INET6, then *getaddrinfo*() shall return IPv4-mapped IPv6 addresses on finding no matching IPv6 addresses. The AI_V4MAPPED flag shall be ignored unless *ai_family* equals AF_INET6. If the AI_ALL flag is used with the AI_V4MAPPED flag, then *getaddrinfo*() shall return all matching IPv6 and IPv4 addresses. The AI_ALL flag without the AI_V4MAPPED flag shall be ignored.

*Rationale*: Austin Group Defect Report(s) applied: 979. See http://austingroupbugs.net/view.php?id=979.

**Change Number: XSH/TC2/D4/0132** [918]

On Page: 919 Line: 30762 Section: freeaddrinfo()
(2013 edition Page: 927 Line: 31262)

In the EXAMPLES section, change from:

```
struct addrinfo hints = {};
```

to:

```
struct addrinfo hints = {0};
```

*Rationale*: Austin Group Defect Report(s) applied: 918. See http://austingroupbugs.net/view.php?id=918.

**Change Number: XSH/TC2/D4/0133** [934]

On Page: 920 Line: 30796 Section: freeaddrinfo()
(2013 edition Page: 928 Line: 31296)

In the APPLICATION USAGE section, add a new paragraph:

Although it is common practice to initialize the *hints* structure using:

```
struct addrinfo hints;
memset(&hints, 0, sizeof hints);
```

this method is not portable according to this standard, because the structure can contain pointer or floating point members that are not required to have an all-bits-zero representation after default initialization. Portable methods make use of default initialization, for example:

```
struct addrinfo hints = { 0 };
```

or

```
static struct addrinfo hints_init;
struct addrinfo hints = hints_init;
```

A future version of this standard may require that a pointer object with an all-bits-zero representation is a null pointer, and that **addrinfo** does not have any floating-point members if a floating-point object with an all-bits-zero representation does not have the value 0.0.

*Rationale*: Austin Group Defect Report(s) applied: 934. See http://austingroupbugs.net/view.php?id=934.

**Change Number: XSH/TC2/D4/0134** [822]

On Page: 924 Line: 30923 Section: freopen()

In the ERRORS section, change from:

[ENOENT]
A component of filename does not name an existing file or filename is an empty string.

to:

[ENOENT]
The *mode* string begins with `'r'` and a component of pathname does not name an existing file, or mode begins with `'w'` or `'a'` and a component of the path prefix of *pathname* does not name an existing file, or *pathname* is an empty string.

[ENOENT] or [ENOTDIR]

The pathname argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If pathname without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822. This change layers upon XSH/TC1/D5/0182 [146,433], changing the last sentence of the change.

**Change Number: XSH/TC2/D4/0135** [936]

On Page: 935 Line: 31352 Section: fscanf()
(2013 edition Page: 944 Line: 31867)

In the SEE ALSO section, add **<inttypes.h>** to the list.

*Rationale*: Austin Group Defect Report(s) applied: 936. See http://austingroupbugs.net/view.php?id=936.

**Change Number: XSH/TC2/D4/0136** [591]

On Page: 945 Line: 31644 Section: fstatat()

Before the fstatat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0137** [817]

On Page: 945 Line: 31686 Section: fstatat()
(2013 edition Page: 954 Line: 32215)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0138** [817]

On Page: 946 Line: 31717 Section: fstatat()
(2013 edition Page: 955 Line: 32247)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0139** [889]

On Page: 948 Line: 31811 Section: fstatat()
(2013 edition Page: 957 Line: 32341)

In the RATIONALE section, delete the paragraph:

The *lstat*() function is not required to update the time-related fields if the named file is not a symbolic link. While the *st_uid*, *st_gid*, *st_atim*, *st_mtim*, and *st_ctim* members of the **stat** structure may apply to a symbolic link, they are not required to do so. No functions in POSIX.1-2008 are required to maintain any of these time fields.

*Rationale*: Austin Group Defect Report(s) applied: 889. See http://austingroupbugs.net/view.php?id=889.

**Change Number: XSH/TC2/D4/0140** [591]

On Page: 968 Line: 32362 Section: futimens()

Before the utimensat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0141** [817]

On Page: 968 Line: 32386 Section: futimens()
(2013 edition Page: 976 Line: 32925)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0142** [485]

On Page: 969 Line: 32407-32408 Section: futimens()

In the DESCRIPTION section, change from:

Upon completion, *futimens*() and *utimensat*() shall mark the last file status change timestamp for update.

to:

Upon successful completion, *futimens*() and *utimensat*() shall mark the last file status change timestamp for update, with the exception that if both *tv_nsec* fields are set to UTIME_OMIT, the file status change timestamp need not be marked for update.

*Rationale*: Austin Group Defect Report(s) applied: 485. See http://austingroupbugs.net/view.php?id=485. The standard requires that timestamps be marked for update even when values being written are identical to what is already present. However, it is unclear whether utimensat is required to write anything when both arguments use UTIME_OMIT. The intent is that implementations may, but not must, optimize these two cases.

**Change Number: XSH/TC2/D4/0143** [817]

On Page: 969 Line: 32435 Section: futimens()
(2013 edition Page: 977 Line: 32973)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0144** [73]

On Page: 973 Line: 32586 Section: fwprintf()

In the DESCRIPTION section, change from:

In format wide-character strings containing the % form of conversion specification, each argument in the argument list shall be used exactly once.

to:

In format wide-character strings containing the % form of conversion specification, each argument in the argument list shall be used exactly once. It is unspecified whether an encoding error occurs if the format string contains **wchar_t** values that do not correspond to members of the character set of the current locale and the specified semantics do not require that value to be processed by *wcrtomb*().

*Rationale*: Austin Group Defect Report(s) applied: 73. See http://austingroupbugs.net/view.php?id=73.

A clarification has been made in the C11 standard.

**Change Number: XSH/TC2/D4/0145** [894]

On Page: 974 Line: 32634 Section: fwprintf()
(2013 edition Page: 983 Line: 33180)

In the DESCRIPTION section, change from:

For o conversion, it increases the precision (if necessary) to force the first digit of the result to be zero.

to:

For o conversion, it shall increase the precision, if and only if necessary, to force the first digit of the result to be a zero (if the value and precision are both 0, a single 0 is printed).

*Rationale*: Austin Group Defect Report(s) applied: 894. See http://austingroupbugs.net/view.php?id=894.

**Change Number: XSH/TC2/D4/0146** [557]

On Page: 979 Line: 32817,32838 Section: fwprintf()

In the ERRORS section, on line 32817 delete:

In addition, all forms of *fwprintf*() may fail if:

[EINVAL] There are insufficient arguments.

In the RATIONALE section, on line 32838 change from:

None

to:

If an implementation detects that there are insufficient arguments for the format, it is recommended that the function should fail and report an [EINVAL] error.

*Rationale*: Austin Group Defect Report(s) applied: 557. See http://austingroupbugs.net/view.php?id=557.
The descriptions of fprintf() and fwprintf() say "The results are undefined if there are insufficient arguments for the format." However, the ERRORS section on each of those pages also specifies a "may fail" EINVAL error for this condition.

**Change Number: XSH/TC2/D4/0147** [936]

On Page: 979 Line: 32843 Section: fwprintf()
(2013 edition Page: 988 Line: 33389)

In the SEE ALSO section, add **<inttypes.h>** to the list.

*Rationale*: Austin Group Defect Report(s) applied: 936. See http://austingroupbugs.net/view.php?id=936.

**Change Number: XSH/TC2/D4/0148** [73]

On Page: 983 Line: 32957 Section: fwscanf()

In the DESCRIPTION section, change from:

The format is a wide-character string composed of zero or more directives. Each directive is composed of one of the following: one or more white-space wide characters (<space>, <tab>, <newline>, <vertical-tab>, or <form-feed>); an ordinary wide character (neither '%' nor a white-space character); or a conversion specification.

to:

The format is a wide-character string composed of zero or more directives. Each directive is composed of one of the following: one or more white-space wide characters (<space>, <tab>, <newline>, <vertical-tab>, or <form-feed>); an ordinary wide character (neither '%' nor a white-space character); or a conversion specification. It is unspecified whether an encoding error occurs if the format string contains **wchar_t** values that do not correspond to members of the character set of the current locale and the specified semantics do not require that value to be processed by *wcrtomb*().

*Rationale*: Austin Group Defect Report(s) applied: 73. See http://austingroupbugs.net/view.php?id=73. A clarification has been made in the C11 standard.

**Change Number: XSH/TC2/D4/0149** [823]

On Page: 988 Line: 33162 Section: fwscanf()
(2013 edition Page: 997 Line: 33711)

In the RETURN VALUE section, change from:

If the input ends before the first matching failure or conversion, EOF shall be returned. If any error occurs, EOF shall be returned, [CX]and *errno* shall be set to indicate the error[/CX]. If a read error occurs, the error indicator for the stream shall be set.

to:

If the input ends before the first conversion (if any) has completed, and without a matching failure having occurred, EOF shall be returned. If an error occurs before the first conversion (if any) has completed, and without a matching failure having occurred, EOF shall be returned [CX]and *errno* shall be set to indicate the error[/CX]. If a read error occurs, the error indicator for the stream shall be set.

*Rationale*: Austin Group Defect Report(s) applied: 823. See http://austingroupbugs.net/view.php?id=823.

**Change Number: XSH/TC2/D4/0150** [936]

On Page: 989 Line: 33200 Section: fwscanf()
(2013 edition Page: 998 Line: 33749)

In the SEE ALSO section, add **<inttypes.h>** to the list.

*Rationale*: Austin Group Defect Report(s) applied: 936. See http://austingroupbugs.net/view.php?id=936.

**Change Number: XSH/TC2/D4/0151** [826]

On Page: 993 Line: 33318 Section: getc_unlocked()

In the DESCRIPTION section change from:

... in a thread-safe manner. They may only safely be used within a scope protected by *flockfile*() (or *ftrylockfile*()) and *funlockfile*(). These functions may safely be used ...

to:

... in a fully thread-safe manner. They shall be thread-safe when used within a scope protected by *flockfile*() (or *ftrylockfile*()) and *funlockfile*(). These functions can safely be used ...

*Rationale*: Austin Group Defect Report(s) applied: 826. See http://austingroupbugs.net/view.php?id=826. The thread safety of these functions needs to be specified clearly.

**Change Number: XSH/TC2/D4/0152** [796]

On Page: 1003 Line: 33674 Section: getdate()
(2013 edition Page: 1012 Line: 34234)

Change from:

is Mon Sep 22 12:19:47 EDT 1986 and the LC_TIME category is set to the default C locale:

to:

is Mon Sep 22 12:19:47 EDT 1986 and the LC_TIME category is set to the default C or POSIX locale:

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0153** [569]

On Page: 1005 Line: 33740,33751 Section: getdelim()

In the DESCRIPTION section, at line 33740 change from:

The size of the object pointed to by **lineptr* shall be increased to fit the incoming line, if it isn't already large enough, including room for the delimiter and a terminating NUL. The characters read, including any delimiter, shall be stored in the string pointed to by the *lineptr* argument, and a terminating NUL added when the delimiter or end of file is encountered.

to:

If **lineptr* is a null pointer or if the object pointed to by **lineptr* is of insufficient size, an object shall be allocated as if by *malloc*() or the object shall be reallocated as if by *realloc*(), respectively, such that the object is large enough to hold the characters to be written to it, including the terminating NUL, and **n* shall be set to the new size. If the object was allocated, or if the reallocation operation moved the object, **lineptr*

shall be updated to point to the new object or new location. The characters read, including any delimiter, shall be stored in the object, and a terminating NUL added when the delimiter or end of file is encountered.

In the RETURN VALUE section, at line 33751 change from:

shall return the number of characters written

to:

shall return the number of bytes written

*Rationale*: Austin Group Defect Report(s) applied: 569. See http://austingroupbugs.net/view.php?id=569. The changes described here are made to match existing practice and describe what was originally intended.

**Change Number: XSH/TC2/D4/0154** [571]

On Page: 1005 Line: 33753 Section: getdelim()

In the RETURN VALUE section, change from:

If no characters were read, and the end-of-file indicator for the stream is set, or if the stream is at end-of-file, the end-of-file indicator for the stream shall be set and the function shall return -1.

to:

If the end-of-file indicator for the stream is set, or if no characters were read and the stream is at end-of-file, the end-of-file indicator for the stream shall be set and the function shall return -1.

*Rationale*: Austin Group Defect Report(s) applied: 571. See http://austingroupbugs.net/view.php?id=571. The changes described here are made to match existing practice and describe what was originally intended.

**Change Number: XSH/TC2/D4/0155** [570]

On Page: 1005 Line: 33764 Section: getdelim()

In the ERRORS section, replace the EOVERFLOW description with:

The number of bytes to be written into the buffer, including the delimiter character (if encountered), would exceed {SSIZE_MAX}.

*Rationale*: Austin Group Defect Report(s) applied: 570. See http://austingroupbugs.net/view.php?id=570. The changes described here are made to match existing practice and describe what was originally intended.

**Change Number: XSH/TC2/D4/0156** [511]

On Page: 1007 Line: 33813,33824 Section: getegid()

In the DESCRIPTION section, on line 33813 add a sentence:

The *getegid*() function shall not modify *errno*.

In the RATIONALE section, on line 33824 change from:

None.

to:

In a conforming environment, *getegid*() will always succeed. It is possible for implementations to provide an extension where a process in a non-conforming environment will not be associated with a user or group ID. It is recommended that such implementations return (**gid_t**)-1 and set *errno* to indicate such an environment; doing so does not violate this standard, since such an environment is already an extension.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XSH/TC2/D4/0157** [656]

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.1-2008. [CX]The returned string pointer might also be invalidated if the calling thread is terminated.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0238 [75,428].

The change is to add the following text to the end of the paragraph at 2013 edition P1017, L34414-34416:

[CX]The returned string pointer might also be invalidated if the calling thread is terminated.[/CX]

**Change Number: XSH/TC2/D4/0158** [511]

On Page: 1011 Line: 33934,33945 Section: geteuid()

In the DESCRIPTION section, on line 33934 add a sentence:

The *geteuid*() function shall not modify *errno*.

In the RATIONALE section, on line 33945 change from:

None.

to:

In a conforming environment, *geteuid*() will always succeed. It is possible for implementations to provide an extension where a process in a non-conforming environment will not be associated with a user or group ID. It is recommended that such implementations return (**uid_t**)-1 and set *errno* to indicate such an environment; doing so does not violate this standard, since such an environment is already an extension.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XSH/TC2/D4/0159** [511]

On Page: 1012 Line: 33966,33977 Section: getgid()

In the DESCRIPTION section, on line 33966 add a sentence

The *getgid*() function shall not modify *errno*.

In the RATIONALE section, on line 33977 change from:

None.

to:

In a conforming environment, *getgid*() will always succeed. It is possible for implementations to provide an extension where a process in a non-conforming environment will not be associated with a user or group ID. It is recommended that such implementations return (**gid_t**)-1 and set *errno* to indicate such an environment; doing so does not violate this standard, since such an environment is already an extension.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XSH/TC2/D4/0160** [808]

On Page: 1014 Line: 34008 Section: getgrgid()

In the DESCRIPTION add a new paragraph after the second paragraph:

Applications wishing to check for error situations should set *errno* to 0 before calling *getgrgid*(). If *getgrgid*() returns a null pointer and *errno* is set to non-zero, an error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0161** [808]

On Page: 1014 Line: 34019 Section: getgrgid()

In the RETURN VALUE section, change from:

On error, *errno* shall be set to indicate the error.

to:

If the requested entry was not found, *errno* shall not be changed. On error, *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0162** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0241 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P1023, L34589-34592:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0163** [808]

On Page: 1016 Line: 34082 Section: getgrgid()

In the APPLICATION USAGE section, delete the first paragraph:

Applications wishing to check for error situations should set *errno* to 0 before calling *getgrgid*(). If *errno* is set on return, an error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0164** [808]

On Page: 1018 Line: 34136 Section: getgrnam()

In the DESCRIPTION section, add a new paragraph after the second paragraph:

Applications wishing to check for error situations should set *errno* to 0 before calling *getgrnam*(). If *getgrnam*() returns a null pointer and *errno* is set to non-zero, an error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0165** [808]

On Page: 1018 Line: 34147 Section: getgrnam()
(2013 edition Page: 1027 Line: 34718)

In the RETURN VALUE section, change from:

On error, *errno* shall be set to indicate the error.

to:

If the requested entry was not found, *errno* shall not be changed. On error, *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0166** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0242 [75].

The change is to add the following text to the end of the paragraph at 2013 edition, P1027, L34720-34723:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0167** [808]

On Page: 1019 Line: 34196 Section: getgrnam()

In the APPLICATION USAGE section, delete the first paragraph:

Applications wishing to check for error situations should set *errno* to 0 before calling *getgrnam*(). If *errno* is set on return, an error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

Application developers requiring to check for errors from these functions should set errno to zero before calling the function if they need to determine if an error occurred.

**Change Number: XSH/TC2/D4/0168** [656]

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*(). The returned pointer and the string content might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0244 [75].

The change is to add the following text to the end of the paragraph at 2013 edition, P1038, L35048-35049:

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0169** [608]

On Page: 1040 Line: 34848 Section: getopt()

In the DESCRIPTION section change from:

*getopt*() shall also print a diagnostic message to *stderr* in the format specified for the *getopts* utility.

to:

*getopt*() shall also print a diagnostic message to *stderr* in the format specified for the *getopts* utility, unless the *stderr* stream has wide orientation in which case the behavior is undefined.

In the APPLICATION USAGE section, add a new paragraph after page 1043 line 34980:

Applications which use wide character output functions with *stderr* should ensure that any calls to *getopt*() do not write to *stderr*, either by setting *opterr* to 0 or by ensuring the first character of *optstring* is always a <colon>.

*Rationale*: Austin Group Defect Report(s) applied: 608. See http://austingroupbugs.net/view.php?id=608.

**Change Number: XSH/TC2/D4/0170** [808]

On Page: 1057 Line: 35318 Section: getpwnam()
(2013 edition Page: 1065 Line: 35887)

In the RETURN VALUE section, change from:

On error, *errno* shall be set to indicate the error.

to:

If the requested entry was not found, *errno* shall not be changed. On error, *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0171** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0255 [75].

The change is to add the following text to the end of the paragraph at 2013 edition, P1065, L35888-35891:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0172** [808]

On Page: 1061 Line: 35451 Section: getpwuid()
(2013 edition Page: 1069 Line: 36023)

In the RETURN VALUE section, change from:

On error, *errno* shall be set to indicate the error.

to:

If the requested entry was not found, *errno* shall not be changed. On error, *errno* shall be set to indicate the error.

*Rationale*: Austin Group Defect Report(s) applied: 808. See http://austingroupbugs.net/view.php?id=808.

**Change Number: XSH/TC2/D4/0173** [656]

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*(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0256 [75].

The change is to add the following text to the end of the paragraph at 2013 edition, P1069, L36024-36027:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0174** [791]

On Page: 1078 Line: 35928 Section: getsubopt()

In the SYNOPSIS section, add a CX margin marker and shading to all the synopsis.

*Rationale*: Austin Group Defect Report(s) applied: 791. See http://austingroupbugs.net/view.php?id=791. This function in <stdlib.h> is an extension over the C standard contents for the header.

**Change Number: XSH/TC2/D4/0175** [511]

On Page: 1083 Line: 36116 Section: getuid()

In the DESCRIPTION section add a sentence:

The *getuid*() function shall not modify *errno*.

In the RATIONALE section on line 36134, change from:

None.

to:

In a conforming environment, *getuid*() will always succeed. It is possible for implementations to provide an extension where a process in a non-conforming environment will not be associated with a user or group ID. It is recommended that such implementations return (**uid_t**)-1 and set *errno* to indicate such an

environment; doing so does not violate this standard, since such an environment is already an extension.

*Rationale*: Austin Group Defect Report(s) applied: 511. See http://austingroupbugs.net/view.php?id=511.

**Change Number: XSH/TC2/D4/0176** [897]

On Page: 1083 Line: 36124 Section: getuid()
(2013 edition Page: 1091 Line: 36709)

In the EXAMPLES section, change from:

The following example sets the effective user ID and the real user ID of the current process to the real user ID of the caller.

```
#include <unistd.h>
#include <sys/types.h>
...
setreuid(getuid(), getuid());
```

to:

The following example sets the effective user ID of the calling process to the real user ID.

```
#include <unistd.h>
...
seteuid(getuid());
```

*Rationale*: Austin Group Defect Report(s) applied: 897. See http://austingroupbugs.net/view.php?id=897.

**Change Number: XSH/TC2/D4/0177** [506]

On Page: 1093 Line: 36468 Section: grantpt()

In the ERRORS section, move line 36472 prior to 36470 (sort [EACCES] before [EBADF]).

*Rationale*: Austin Group Defect Report(s) applied: 506. See http://austingroupbugs.net/view.php?id=506.
This is an editorial change.

**Change Number: XSH/TC2/D4/0178** [777]

On Page: 1116 Line: 37199 Section: inet_ntop()

In the DESCRIPTION section, change from:

The preferred form is "`x:x:x:x:x:x:x:x`", where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address. Leading zeros in individual fields can be omitted, but there shall be at least one numeral in every field.

to:

The preferred form is "`x:x:x:x:x:x:x:x`", where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address. Leading zeros in individual fields can be omitted, but there shall be one to four

hexadecimal digits in every field.

*Rationale*: Austin Group Defect Report(s) applied: 777. See http://austingroupbugs.net/view.php?id=777.

**Change Number: XSH/TC2/D4/0179** [743]

On Page: 1119 Line: 37298 Section: initstate()
(2013 edition Page: 1128 Line: 37891)

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

These functions should be avoided whenever non-trivial requirements (including safety) have to be fulfilled.

*Rationale*: Austin Group Defect Report(s) applied: 743. See http://austingroupbugs.net/view.php?id=743.

**Change Number: XSH/TC2/D4/0180** [685]

On Page: 1169 Line: 38878 Section: iswalnum()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0181** [685]

On Page: 1171 Line: 38926 Section: iswalpha()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0182** [685]

On Page: 1173 Line: 38974 Section: iswblank()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0183** [685]

On Page: 1174 Line: 39015 Section: iswcntrl()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0184** [799]

On Page: 1176 Line: 39070 Section: iswctype()
(2013 edition Page: 1188 Line: 39711)

In the RETURN VALUE section, change from:

If *charclass* is 0

to:

If *charclass* is (**wctype_t**)0

*Rationale*: Austin Group Defect Report(s) applied: 799. See http://austingroupbugs.net/view.php?id=799.

**Change Number: XSH/TC2/D4/0185** [799]

On Page: 1177 Line: 39130 Section: iswctype()
(2013 edition Page: 1189 Line: 39770)

In the CHANGE HISTORY section, change from:

The behavior of *n*=0 is now described.

to:

The behavior of *charclass* = (**wctype_t**)0 is now described.

*Rationale*: Austin Group Defect Report(s) applied: 799. See http://austingroupbugs.net/view.php?id=799.

**Change Number: XSH/TC2/D4/0186** [685]

On Page: 1178 Line: 39149 Section: iswdigit()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0187** [685]

On Page: 1180 Line: 39197 Section: iswgraph()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0188** [685]

On Page: 1182 Line: 39245 Section: iswlower()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0189** [685]

On Page: 1184 Line: 39293 Section: iswprint()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0190** [685]

On Page: 1186 Line: 39341 Section: iswpunct()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0191** [685]

On Page: 1188 Line: 39389 Section: iswspace()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0192** [685]

On Page: 1190 Line: 39437 Section: iswupper()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0193** [685]

On Page: 1192 Line: 39485 Section: iswxdigit()

In the DESCRIPTION section, change from:

is a wide-character code corresponding to a valid character in the current locale

to:

is a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0194** [765]

On Page: 1201 Line: 39708 Section: kill()
(2013 edition Page: 1213 Line: 40379)

In the RATIONALE section, change from:

Existing implementations vary on the result of a *kill*() with *pid* indicating an inactive process (a terminated process that has not been waited for by its parent). Some indicate success on such a call (subject to permission checking), while others give an error of [ESRCH]. Since the definition of process lifetime in this volume of POSIX.1-2008 covers inactive processes, the [ESRCH] error as described is inappropriate in this case. In particular, this means that an application cannot have a parent process check for termination of a particular child with *kill*(). (Usually this is done with the null signal; this can be done reliably with *waitpid*().)

to:

Historical implementations varied on the result of a *kill*() with *pid* indicating a zombie process. Some indicated success on such a call (subject to permission checking), while others gave an error of [ESRCH]. Since the definition of process lifetime in this standard covers zombie processes, the [ESRCH] error as described is inappropriate in this case and implementations that give this error do not conform. This means that an application cannot have a parent process check for termination of a particular child by sending it the null signal with *kill*(), but must instead use *waitpid*() or *waitid*().

*Rationale*: Austin Group Defect Report(s) applied: 765. See http://austingroupbugs.net/view.php?id=765.

**Change Number: XSH/TC2/D4/0195** [873]

On Page: 1216 Line: 40102 Section: link()
(2013 edition Page: 1228 Line: 40780)

In the NAME section, delete:

relative to two directory file descriptors

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0196** [591]

On Page: 1216 Line: 40106 Section: link()

Before the linkat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0197** [817]

On Page: 1216 Line: 40126 Section: link()
(2013 edition Page: 1228 Line: 40805)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0198** [822]

On Page: 1217 Line: 40156 Section: link()
(2013 edition Page: 1229 Line: 40836)

In the RETURN VALUE section, add after the [ENOENT] error:

[ENOENT] or [ENOTDIR]

> The *path1* argument names an existing non-directory file, and the *path2* argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *path2* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822.

**Change Number: XSH/TC2/D4/0199** [817]

On Page: 1217 Line: 40170 Section: link()
(2013 edition Page: 1229 Line: 40853)

In the ERRORS section, add:

[EACCES]

The access mode of the open file description associated with *fd1* or *fd2* is not O_SEARCH and the permissions of the directory underlying *fd1* or *fd2* respectively do not permit directory searches.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0200** [656]

On Page: 1235 Line: 40719 Section: localeconv()

In the RETURN VALUE section, change from:

The application shall not modify the structure pointed to by the return value which may be overwritten by a subsequent call to *localeconv*(). In addition, calls to *setlocale*() with the categories LC_ALL, LC_MONETARY, or LC_NUMERIC or calls to *uselocale*() which change the categories LC_MONETARY or LC_NUMERIC may overwrite the contents of the structure.

to:

The application shall not modify the structure to which the return value points, [CX]nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or[/CX] the structure [CX]or the storage areas[/CX] might be overwritten by a subsequent call to *localeconv*(). In addition, [CX]the returned pointer, and pointers within the structure, might be invalidated or[/CX] the structure [CX]or the storage areas[/CX] might be overwritten by subsequent calls to *setlocale*() with the categories LC_ALL, LC_MONETARY, or LC_NUMERIC, [CX]or by calls to *uselocale*() which change the categories LC_MONETARY or LC_NUMERIC. The returned pointer, pointers within the structure, the structure, and the storage areas, might also be invalidated if the calling thread is terminated.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0362 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P1247, L41402-41409:

The returned pointer, pointers within the structure, the structure, and the storage areas, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0201** [664]

On Page: 1238 Line: 40811 Section: localtime()

In the DESCRIPTION section, change from:

If *localtime_r*() does not set *tzname*, it shall not set *daylight* and shall not set *timezone*.

to:

If *localtime_r*() sets *tzname*, it shall also set *daylight* and *timezone*. If *localtime_r*() does not set *tzname*, it shall not set *daylight* and shall not set *timezone*.

*Rationale*: Austin Group Defect Report(s) applied: 664. See http://austingroupbugs.net/view.php?id=664. The change is made for consistent behavior.

**Change Number: XSH/TC2/D4/0202** [516]

On Page: 1256 Line: 41378,41395 Section: longjmp()

In the DESCRIPTION section, at line 41378 change from:

As it bypasses the usual function call and return mechanisms, *longjmp*() shall execute correctly in contexts of interrupts, signals, and any of their associated functions. However, if *longjmp*() is invoked from a nested signal handler (that is, from a function invoked as a result of a signal raised during the handling of another signal), the behavior is undefined.

to:

Although *longjmp*() is an async-signal-safe function, if it is invoked from a signal handler which interrupted a non-async-signal-safe function or equivalent (such as the processing equivalent to *exit*() performed after a return from the initial call to *main*()), the behavior of any subsequent call to a non-async-signal-safe function or equivalent is undefined.

In the APPLICATION USAGE section, at line 41395 add a new paragraph:

It is recommended that applications do not call *longjmp*() or *siglongjmp*() from signal handlers. To avoid undefined behavior when calling these functions from a signal handler, the application needs to ensure one of the following two things:

1. After the call to *longjmp*() or *siglongjmp*() the process only calls async-signal-safe functions and does not return from the initial call to *main*().

2. Any signal whose handler calls *longjmp*() or *siglongjmp*() is blocked during **every** call to a non-async-signal-safe function, and no such calls are made after returning from the initial call to *main*().

*Rationale*: Austin Group Defect Report(s) applied: 516. See http://austingroupbugs.net/view.php?id=516. The restrictions on using longjmp() and siglongjmp() are more restrictive than they need to be on POSIX systems. The loosened restrictions presented here do not break existing implementations and make it easier for application writers to create portable applications.

**Change Number: XSH/TC2/D4/0203** [526]

On Page: 1268 Line: 41713,41718 Section: malloc()

In the DESCRIPTION section, on line 41713 change from:

If the size of the space requested is 0, the behavior is implementation-defined: the value returned shall be either a null pointer or a unique pointer.

to:

If the size of the space requested is 0, the behavior is implementation-defined: either a null pointer shall be returned, or the behavior shall be as if the size were some nonzero value, except that the behavior is undefined if the returned pointer is used to access an object.

In the RETURN VALUE section, on line 41718 change from:

either a null pointer or a unique pointer that can be successfully passed to *free*() shall be returned.

to:

either:

- a null pointer shall be returned [CX]and *errno* may be set to an implementation-defined value[/CX], or
- a pointer to the allocated space shall be returned. The application shall ensure that the pointer is not used to access an object.

*Rationale*: Austin Group Defect Report(s) applied: 526. See http://austingroupbugs.net/view.php?id=526.

**Change Number: XSH/TC2/D4/0204** [663,674]

On Page: 1270 Line: 41775 Section: mblen()
(2013 edition Page: 1282 Line: 42472)

In the ERRORS section, change from:

[EILSEQ]
[XSI]An invalid character sequence is detected.[/XSI]

to:

[EILSEQ]
[CX]An invalid character sequence is detected. In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

On Page: 1272 Line: 41825 Section: mbrlen()
(2013 edition Page: 1284 Line: 42526)

In the ERRORS section, change from:

[EILSEQ]
An invalid character sequence is detected.

to:

[EILSEQ]
An invalid character sequence is detected. [CX]In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

On Page: 1275 Line: 41890 Section: mbrtowc()
(2013 edition Page: 1287 Line: 42594)

In the ERRORS section, change from:

[EILSEQ]
An invalid character sequence is detected.

to:

[EILSEQ]
An invalid character sequence is detected. [CX]In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663,674. See http://austingroupbugs.net/view.php?id=663.
See http://austingroupbugs.net/view.php?id=674.

**Change Number: XSH/TC2/D4/0205** [601]

On Page: 1277 Line: 41984 Section: mbsrtowcs()

In the DESCRIPTION section, change from:

except that the conversion of characters pointed to by *src* is limited to at most *nmc* bytes (the size of the input buffer).

to (all within the CX shading):

except that the conversion of characters indirectly pointed to by *src* is limited to at most *nmc* bytes (the size of the input buffer), and under conditions where *mbsrtowcs*() would assign the address just past the last character converted (if any) to the pointer object pointed to by *src*, *mbsnrtowcs*() shall instead assign the address just past the last byte processed (if any) to that pointer object. If the input buffer ends with an incomplete character, it is unspecified whether conversion stops at the end of the previous character (if any), or at the end of the input buffer. In the latter case, a subsequent call to *mbsnrtowcs*() with an input buffer that starts with the remainder of the incomplete character shall correctly complete the conversion of that character.

*Rationale*: Austin Group Defect Report(s) applied: 601. See http://austingroupbugs.net/view.php?id=601.

**Change Number: XSH/TC2/D4/0206** [663]

On Page: 1278 Line: 41998 Section: mbsrtowcs()
(2013 edition Page: 1290 Line: 42706)

In the ERRORS section, change from:

[EILSEQ]
An invalid character sequence is detected.

to:

[EILSEQ]
An invalid character sequence is detected. [CX]In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XSH/TC2/D4/0207** [601]

On Page: 1278 Line: 42008 Section: mbsrtowcs()

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version may require that when the input buffer ends with an incomplete character, conversion stops at the end of the input buffer.

*Rationale*: Austin Group Defect Report(s) applied: 601. See http://austingroupbugs.net/view.php?id=601.

**Change Number: XSH/TC2/D4/0208** [663,674]

On Page: 1279 Line: 42051 Section: mbstowcs()
(2013 edition Page: 1291 Line: 42760)

In the ERRORS section, change from:

[EILSEQ]
[XSI]An invalid byte sequence is detected.[/XSI]

to:

[EILSEQ]
[CX]An invalid character sequence is detected. In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663,674. See http://austingroupbugs.net/view.php?id=663.
See http://austingroupbugs.net/view.php?id=674.

**Change Number: XSH/TC2/D4/0209** [663,674]

On Page: 1281 Line: 42104 Section: mbtowc()
(2013 edition Page: 1293 Line: 42815)

In the ERRORS section, change from:

[EILSEQ]
[XSI]An invalid character sequence is detected.[/XSI]

to:

[EILSEQ]
[CX]An invalid character sequence is detected. In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 663,674. See http://austingroupbugs.net/view.php?id=663.

See http://austingroupbugs.net/view.php?id=674.

**Change Number: XSH/TC2/D4/0210** [873]

On Page: 1289 Line: 42299 Section: mkdir()
(2013 edition Page: 1301 Line: 43016)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0211** [591]

On Page: 1289 Line: 42301 Section: mkdir()

Before the mkdirat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0212** [817]

On Page: 1289 Line: 42323 Section: mkdir()
(2013 edition Page: 1301 Line: 43040)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0213** [817]

On Page: 1290 Line: 42348 Section: mkdir()
(2013 edition Page: 1302 Line: 43066)

In the ERRORS section, add:

[EACCES]
The access mode of the open file description associated with *fd* is not O_SEARCH and the

permissions of the directory underlying *fd* do not permit directory searches.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0214** [591]

On Page: 1291 Line: 42393 Section: mkdir()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0215** [567,669]

On Page: 1292 Line: 42419-42424 Section: mkdtemp()

Replace the entire DESCRIPTION section with:

DESCRIPTION

> The *mkdtemp*() function shall create a directory with a unique name derived from *template*. The application shall ensure that the string provided in *template* is a pathname ending with at least six trailing 'X' characters. The *mkdtemp*() function shall modify the contents of *template* by replacing six or more 'X' characters at the end of the pathname with the same number of characters from the portable filename character set. The characters shall be chosen such that the resulting pathname does not duplicate the name of an existing file at the time of the call to *mkdtemp*(). The *mkdtemp*() function shall use the resulting pathname to create the new directory as if by a call to:
>
> `mkdir(`*`pathname`*`, S_IRWXU)`
>
> The *mkstemp*() function shall create a regular file with a unique name derived from *template* and return a file descriptor for the file open for reading and writing. The application shall ensure that the string provided in *template* is a pathname ending with at least six trailing 'X' characters. The *mkstemp*() function shall modify the contents of *template* by replacing six or more 'X' characters at the end of the pathname with the same number of characters from the portable filename character set. The characters shall be chosen such that the resulting pathname does not duplicate the name of an existing file at the time of the call to *mkstemp*(). The *mkstemp*() function shall use the resulting pathname to create the file, and obtain a file descriptor for it, as if by a call to:
>
> `open(`*`filename`*`, O_RDWR|O_CREAT|O_EXCL, S_IRUSR|S_IWUSR)`
>
> By behaving as if the O_EXCL flag for *open*() is set, the function prevents any possible race condition between testing whether the file exists and opening it for use.

In the RETURN VALUE section, at line 42435 change from:

Upon successful completion, the *mkdtemp*() function shall return a pointer to the string containing the directory name if it was created.

to:

Upon successful completion, the *mkdtemp*() function shall return the value of *template*.

In the RETURN VALUE section, at line 42439 change from:

Otherwise, it shall return -1 if no suitable file could be created.

to:

Otherwise, it shall return -1 and shall set *errno* to indicate the error.

In the APPLICATION USAGE section, at line 42476 change from:

The *mkdtemp*() and *mkstemp*() functions need not check to determine whether the filename part of template exceeds the maximum allowable filename length.

to:

Portable applications should pass exactly six trailing 'X's in the template and no more; implementations may treat any additional trailing 'X's as either a fixed or replaceable part of the template. To be sure of only passing six, a fixed string of at least one non-'X' character should precede the six 'X's.

Since 'X' is in the portable filename character set, some of the replacement characters can be 'X's, leaving part (or even all) of the template effectively unchanged.

*Rationale*: Austin Group Defect Report(s) applied: 567,669. See http://austingroupbugs.net/view.php?id=567.
See http://austingroupbugs.net/view.php?id=669.

**Change Number: XSH/TC2/D4/0216** [873]

On Page: 1295 Line: 42496 Section: mkfifo()
(2013 edition Page: 1307 Line: 43219)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0217** [591]

On Page: 1295 Line: 42498 Section: mkfifo()

Before the mkfifoat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0218** [817]

On Page: 1295 Line: 42519 Section: mkfifo()
(2013 edition Page: 1307 Line: 43242)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0219** [822]

On Page: 1296 Line: 42537 Section: mkfifo()

In the ERRORS section, change from:

[ENOENT]
A component of the path prefix specified by *path* does not name an existing directory or *path* is an empty string.

to:

[ENOENT]
A component of the path prefix of *path* does not name an existing file or *path* is an empty string.

[ENOENT] or [ENOTDIR]
The *path* argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *path* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822.
This change is layered upon XSH/TC1/D5/0384 [146,435], changing the last sentence.

**Change Number: XSH/TC2/D4/0220** [817]

On Page: 1296 Line: 42544 Section: mkfifo()
(2013 edition Page: 1308 Line: 43272)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0221** [591]

On Page: 1297 Line: 42594 Section: mkfifo()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0222** [591]

On Page: 1298 Line: 42617 Section: mknod()

Before the mknodat SYNOPSIS line, insert a line with OH XSI shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0223** [817]

On Page: 1299 Line: 42669 Section: mknod()
(2013 edition Page: 1313 Line: 43434)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0224** [822]

On Page: 1299 Line: 42689 Section: mknod()

In the ERRORS section, change from:

[ENOENT]
A component of the path prefix specified by *path* does not name an existing directory or *path*

is an empty string.

to:

[ENOENT]
A component of the path prefix of *path* does not name an existing file or *path* is an empty string.

[ENOENT] or [ENOTDIR]
The path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *path* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822. This change is layered upon XSH/TC1/D5/0390 [146,435], changing the last sentence.

**Change Number: XSH/TC2/D4/0225** [817]

On Page: 1300 Line: 42699 Section: mknod()
(2013 edition Page: 1312 Line: 43399)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0226** [591]

On Page: 1301 Line: 42745 Section: mknod()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0227** [902]

On Page: 1302 Line: 42761 Section: mkstemp()
(2013 edition Page: 1315 Line: 43498)

In the NAME section, change from:

create a unique directory

to:

create a unique file

*Rationale*: Austin Group Defect Report(s) applied: 902. See http://austingroupbugs.net/view.php?id=902.

**Change Number: XSH/TC2/D4/0228** [724]

On Page: 1303 Line: 42778-42792 Section: mktime()

In the DESCRIPTION section, at line 42778 change from:

The original values of the *tm_wday* and *tm_yday* components of the structure are ignored, and the original values of the other components are not restricted to the ranges described in <**time.h**>.

to:

The original values of the *tm_wday* and *tm_yday* components of the structure shall be ignored, and the original values of the other components shall not be restricted to the ranges described in <**time.h**>.

At line 42789 change from:

where the names in the structure and in the expression correspond

to:

where the names other than *tm_yday* in the structure and in the expression correspond, and the *tm_yday* value used in the expression is the day of the year from 0 to 365 inclusive, calculated from the other **tm** structure members specified in <**time.h**> (excluding *tm_wday*).

At line 42792 change from:

and the other components are set to represent the specified time

to:

and the other components shall be set to represent the specified time

*Rationale*: Austin Group Defect Report(s) applied: 724. See http://austingroupbugs.net/view.php?id=724.

**Change Number: XSH/TC2/D4/0229** [852]

On Page: 1311 Line: 43068 Section: mmap()
(2013 edition Page: 1324 Line: 43807)

In the DESCRIPTION section, change from:

If a MAP_FIXED request is successful, the mapping established by *mmap*() replaces any previous mappings for the pages in the range [*pa*,*pa*+*len*) of the process.

to:

If a MAP_FIXED request is successful, then any previous mappings [ML|MLR]or memory locks[/ML|MLR] for those whole pages containing any part of the address range [*pa*,*pa*+*len*) shall be removed, as if by an appropriate call to *munmap*(), before the new mapping is established.

On Page: 1313 Line: 43189 Section: mmap()
(2013 edition Page: 1326 Line: 43928)

In the RATIONALE section, change from:

If an application requests a mapping that would overlay existing mappings in the process, it might be desirable that an implementation detect this and inform the application. However, the default, portable (not MAP_FIXED) operation does not overlay existing mappings. On the other hand, if the program specifies a fixed address mapping (which requires some implementation knowledge to determine a suitable address, if the function is supported at all), then the program is presumed to be successfully managing its own address space and should be trusted when it asks to map over existing data structures. Furthermore, it is also desirable to make as few system calls as possible, and it might be considered onerous to require an *munmap*() before an *mmap*() to the same address range. This volume of POSIX.1-2008 specifies that the new mappings replace any existing mappings, following existing practice in this regard.

to:

If an application requests a mapping that overlaps existing mappings in the process, it might be desirable that an implementation detect this and inform the application. However, if the program specifies a fixed address mapping (which requires some implementation knowledge to determine a suitable address, if the function is supported at all), then the program is presumed to be successfully managing its own address space and should be trusted when it asks to map over existing data structures. Furthermore, it is also desirable to make as few system calls as possible, and it might be considered onerous to require an *munmap*() before an *mmap*() to the same address range. This volume of POSIX.1-2008 specifies that the new mapping replaces any existing mappings (implying an automatic *munmap*() on the address range), following existing practice in this regard. The standard developers also considered whether there should be a way for new mappings to overlay existing mappings, but found no existing practice for this.

On Page: 1315 Line: 43244 Section: mmap()
(2013 edition Page: 1328 Line: 43983)

In the RATIONALE section, change from:

and the MEMLOCK_FUTURE argument

to:

and the MCL_FUTURE argument

*Rationale*: Austin Group Defect Report(s) applied: 852. See http://austingroupbugs.net/view.php?id=852.

**Change Number: XSH/TC2/D4/0230** [504]

On Page: 1339 Line: 44014 Section: mq_unlink()

In the ERRORS section, after line 44014 (EACCES) add:

[EINTR]
The call to *mq_unlink*() blocked waiting for all references to the named message queue to be closed and a signal interrupted the call.

*Rationale*: Austin Group Defect Report(s) applied: 504. See http://austingroupbugs.net/view.php?id=504.

**Change Number: XSH/TC2/D4/0231** [909]

On Page: 1360 Line: 44625 Section: nanosleep()
(2013 edition Page: 1373 Line: 45379)

In the RETURN VALUE section, change from:

The *rqtp* and *rmtp* arguments may point to the same object.

to:

The *rqtp* and *rmtp* arguments can point to the same object.

*Rationale*: Austin Group Defect Report(s) applied: 909. See http://austingroupbugs.net/view.php?id=909.

**Change Number: XSH/TC2/D4/0232** [781]

On Page: 1364 Line: 44728 Section: newlocale()
(2013 edition Page: 1376 Line: 45474)

In the DESCRIPTION section, change from:

any of the other implementation-defined *LC_*_MASK* values defined in **<locale.h>**

to:

any of the implementation-defined mask values defined in **<locale.h>**

*Rationale*: Austin Group Defect Report(s) applied: 781. See http://austingroupbugs.net/view.php?id=781.

**Change Number: XSH/TC2/D4/0233** [673]

On Page: 1365 Line: 44763 Section: newlocale()

In the EXAMPLES section, change from:

and the *LC_TIME* category data from a locale *tok2*

to:

and the *LC_TIME* category data from a locale *loc2*

*Rationale*: Austin Group Defect Report(s) applied: 673. See http://austingroupbugs.net/view.php?id=673.

**Change Number: XSH/TC2/D4/0234** [656]

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*(). The returned pointer and the string content might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0415 [75,402].

The change is to add the following text to the end of the paragraph at 2013 edition P1387, L45872-45880:

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0235** [873]

On Page: 1379 Line: 45158 Section: open()
(2013 edition Page: 1391 Line: 49540)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0236** [835]

On Page: 1379 Line: 45169 Section: open()
(2013 edition Page: 1391 Line: 45951)

In the DESCRIPTION section, change from:

... return a file descriptor for the named file that is the lowest file descriptor not currently open for that process.

to:

... return a file descriptor for the named file, allocated as described in [xref to new section 2.14].

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0237** [847]

On Page: 1379 Line: 45193 Section: open()
(2013 edition Page: 1391 Line: 45975)

In the DESCRIPTION section, change from:

Otherwise, the file shall be created; ...

to:

Otherwise, if O_DIRECTORY is not set the file shall be created as a regular file; ...

On Page: 1381 Line: 45268 Section: open()
(2013 edition Page: 1393 Line: 46052)

In the DESCRIPTION section, insert a new paragraph:

If O_CREAT and O_DIRECTORY are set and the requested access mode is neither O_WRONLY nor O_RDWR, the result is unspecified.

*Rationale*: Austin Group Defect Report(s) applied: 847. See http://austingroupbugs.net/view.php?id=847. The standard does not specify the behavior when open() is called with O_CREAT|O_RDONLY (or O_CREAT|O_SEARCH) on an existing directory. Additionally, some systems want to allow creation of directories using the open() function. This should be an allowed, but not required, extension.

**Change Number: XSH/TC2/D4/0238** [817]

On Page: 1381 Line: 45290 Section: open()
(2013 edition Page: 1394 Line: 46074)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0239** [835]

On Page: 1382 Line: 45300 Section: open()
(2013 edition Page: 1394 Line: 46084)

In the RETURN VALUE section, change from:

representing the lowest numbered unused file descriptor

to:

representing the file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0240** [847]

On Page: 1382 Line: 45314 Section: open()
(2013 edition Page: 1394 Line: 46098)

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

The named file is a directory and *oflag* includes O_WRONLY or O_RDWR.

to:

The named file is a directory and *oflag* includes O_WRONLY or O_RDWR, or includes O_CREAT without O_DIRECTORY.

*Rationale*: Austin Group Defect Report(s) applied: 847. See http://austingroupbugs.net/view.php?id=847. The standard does not specify the behavior when open() is called with O_CREAT|O_RDONLY (or O_CREAT|O_SEARCH) on an existing directory. Additionally, some systems want to allow creation of directories using the open() function. This should be an allowed, but not required, extension.

**Change Number: XSH/TC2/D4/0241** [822]

On Page: 1382 Line: 45322 Section: open()

In the ERRORS section, change from:

[ENOENT]
O_CREAT is not set and the named file does not exist; or O_CREAT is set and either the path prefix does not exist or the *path* argument points to an empty string.

to:

[ENOENT]
O_CREAT is not set and a component of path does not name an existing file, or O_CREAT is set and a component of the path prefix of *path* does not name an existing file, or *path* points to an empty string.

[ENOENT] or [ENOTDIR]
O_CREAT is set, and the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *path* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822. This change is layered upon XSH/TC1/D5/0422 [146], changing the last sentence.

**Change Number: XSH/TC2/D4/0242** [817]

On Page: 1383 Line: 45345 Section: open()
(2013 edition Page: 1395 Line: 46134)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0243** [943]

On Page: 1383 Line: 45361 Section: open()
(2013 edition Page: 1395 Line: 46152)

In the ERRORS section, before the [ETXTBSY] error, add:

[EOPNOTSUPP] The *path* argument names a socket.

*Rationale*: Austin Group Defect Report(s) applied: 943. See http://austingroupbugs.net/view.php?id=943.

**Change Number: XSH/TC2/D4/0244** [588]

On Page: 1388 Line: 45564 Section: open_memstream()

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

After a successful *fclose*(), the pointer referenced by *bufp* can be passed to *free*().

*Rationale*: Austin Group Defect Report(s) applied: 588. See http://austingroupbugs.net/view.php?id=588.

**Change Number: XSH/TC2/D4/0245** [586]

On Page: 1388 Line: 45569 Section: open_memstream()

At the start of ERRORS section, insert:

These functions shall fail if:

[EMFILE]
{STREAM_MAX} streams are currently open in the calling process.

*Rationale*: Austin Group Defect Report(s) applied: 586. See http://austingroupbugs.net/view.php?id=586.

**Change Number: XSH/TC2/D4/0246** [632]

On Page: 1396 Line: 45712 Section: pclose()

At the end of the DESCRIPTION section, add a new paragraph:

If a thread is canceled during execution of *pclose*(), the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 632. See http://austingroupbugs.net/view.php?id=632.

**Change Number: XSH/TC2/D4/0247** [835]

On Page: 1400 Line: 45837 Section: pipe()
(2013 edition Page: 1413 Line: 46643)

Change from:

Their integer values shall be the two lowest available at the time of the *pipe*() call.

to:

The file descriptors shall be allocated as described in [xref to new section 2.14].

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0248** [467,835]

On Page: 1400 Line: 45851 Section: pipe()
(2013 edition Page: 1413 Line: 46657)

Change from:

... otherwise, -1 shall be returned and *errno* set to indicate the error.

to:

... otherwise, -1 shall be returned and *errno* set to indicate the error, no file descriptors shall be allocated and the contents of *fildes* shall be left unmodified.

*Rationale*: Austin Group Defect Report(s) applied: 467,835. See http://austingroupbugs.net/view.php?id=467.
See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0249** [623]

On Page: 1404 Line: 45985 Section: poll()

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

Provided the application does not perform any action that results in unspecified or undefined behavior, the value of the *fd* and *events* members of each element of *fds*[] shall not be modified by *poll*().

*Rationale*: Austin Group Defect Report(s) applied: 623. See http://austingroupbugs.net/view.php?id=623.

**Change Number: XSH/TC2/D4/0250** [683]

On Page: 1404 Line: 45987 Section: poll()
(2013 edition Page: 1417 Line: 46793)

In the RETURN VALUE section, change from:

Upon successful completion, *poll*() shall return a non-negative value. A positive value indicates the total number of file descriptors that have been selected (that is, file descriptors for which the *revents* member is non-zero).

to:

Upon successful completion, *poll*() shall return a non-negative value. A positive value indicates the total number of **pollfd** structures that have selected events (that is, those for which the *revents* member is non-zero).

*Rationale*: Austin Group Defect Report(s) applied: 683. See http://austingroupbugs.net/view.php?id=683.

**Change Number: XSH/TC2/D4/0251** [526]

On Page: 1418 Line: 46423 Section: posix_memalign()

In the DESCRIPTION section change from:

If the size of the space requested is 0, the behavior is implementation-defined; the value returned in *memptr* shall be either a null pointer or a unique pointer.

to:

If the size of the space requested is 0, the behavior is implementation-defined: either a null pointer shall be returned in *memptr*, or the behavior shall be as if the size were some nonzero value, except that the behavior is undefined if the the value returned in *memptr* is used to access an object.

*Rationale*: Austin Group Defect Report(s) applied: 526. See http://austingroupbugs.net/view.php?id=526.

**Change Number: XSH/TC2/D4/0252** [520,526]

On Page: 1418 Line: 46428,46436 Section: posix_memalign()

In the RETURN VALUE section, on line 46428 change from:

otherwise, an error number shall be returned to indicate the error.

to:

otherwise, an error number shall be returned to indicate the error and the contents of *memptr* shall either be left unmodified or be set to a null pointer.

If *size* is 0, either:

- *posix_memalign*() shall not attempt to allocate any space, in which case either an implementation-defined error number shall be returned, or zero shall be returned with a null pointer returned in *memptr*
- *posix_memalign*() shall attempt to allocate some space and if the allocation succeeds, zero shall be returned and a pointer to the allocated space shall be returned in *memptr*. The application shall ensure that the pointer is not used to access an object.

In the EXAMPLES section, on line 46436 change from:

None

to:

The following example shows how applications can obtain consistent behavior on error by setting **memptr* to be a null pointer before calling *posix_memalign*().

```
void *ptr = NULL;
...
//do some work, which might goto error
if (posix_memalign(&ptr, align, size))
  goto error;

//do some more work, which might goto error
...
error:
  free(ptr);
  //more cleanup;
```

*Rationale*: Austin Group Defect Report(s) applied: 520,526. See http://austingroupbugs.net/view.php?id=520.
See http://austingroupbugs.net/view.php?id=526.

**Change Number: XSH/TC2/D4/0253** [835]

On Page: 1420 Line: 46464 Section: posix_openpt()
(2013 edition Page: 1433 Line: 47270)

Change from:

The file descriptor is used by other I/O functions that refer to that pseudo-terminal.

to:

The file descriptor shall be allocated as described in [xref to new section 2.14] and can be used by other I/O functions that refer to that pseudo-terminal.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0254** [835]

On Page: 1420 Line: 46474 Section: posix_openpt()
(2013 edition Page: 1433 Line: 47281)

Change from:

... shall open a master pseudo-terminal device and return a non-negative integer representing the lowest numbered unused file descriptor.

to:

... shall open a file descriptor for a master pseudo-terminal device and return a non-negative integer representing the file descriptor.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0255** [824]

On Page: 1423 Line: 46571,46578 Section: posix_spawn()

In the DESCRIPTION section, at line 46571 change from:

For those file descriptors that remain open, all attributes of the corresponding open file descriptions, including file locks (see *fcntl*()), shall remain unchanged.

to:

For those file descriptors that remain open, the child process shall not inherit any file locks, but all remaining attributes of the open file descriptions (see *fcntl*()) shall remain unchanged.

At line 46578 (the second sentence in item 1 in the list), change from:

All attributes of the corresponding open file descriptions, including file locks (see *fcntl*()), shall remain unchanged.

to:

The child process shall not inherit any file locks, but all remaining attributes of the corresponding open file descriptions (see *fcntl*()) shall remain unchanged.

*Rationale*: Austin Group Defect Report(s) applied: 824. See http://austingroupbugs.net/view.php?id=824. The behavior seen when creating a child using posix_spawn() and when using fork()/exec() should be the same.

**Change Number: XSH/TC2/D4/0256** [835]

On Page: 1515 Line: 48880 Section: posix_typed_mem_open()
(2013 edition Page: 1528 Line: 49729)

Change from:

The file descriptor is used by other functions to refer to that typed memory object.

to:

The file descriptor shall be allocated as described in [xref to new section 2.14] and can be used by other functions to refer to that typed memory object.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0257** [835]

On Page: 1516 Line 48924 Section: posix_typed_mem_open()
(2013 edition Page: 1529 Line: 49773)

Change from:

... return a file descriptor for the typed memory object that is the lowest numbered file descriptor not currently open for that process.

to:

... return a file descriptor for the typed memory object.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0258** [835]

On Page: 1516 Line: 48939 Section: posix_typed_mem_open()
(2013 edition Page: 1529 Line: 49788)

Change from:

... return a non-negative integer representing the lowest numbered unused file descriptor.

to:

... return a non-negative integer representing the file descriptor.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0259** [680]

On Page: 1526 Line: 49239 Section: pselect()
(2013 edition Page: 1539 Line: 50094)

In the ERRORS section, change from:

[EINTR]
The function was interrupted before any of the selected events occurred and before the timeout interval expired.

to:

[EINTR]
The function was interrupted while blocked waiting for any of the selected descriptors to become ready and before the timeout interval expired.

*Rationale*: Austin Group Defect Report(s) applied: 680. See http://austingroupbugs.net/view.php?id=680.

**Change Number: XSH/TC2/D4/0260** [629]

On Page: 1528 Line: 49313,49319 Section: psiginfo()

In the NAME section, change from:

print signal information to standard error

to:

write signal information to standard error

In the DESCRIPTION section, at line 49319 change from:

The *psiginfo*() and *psignal*() functions shall print a message out on *stderr* associated with a signal number. If message is not null and is not the empty string, then the string pointed to by the *message* argument shall be printed first, followed by a <colon>, a <space>, and the signal description string indicated by *signum*, or by the signal associated with *pinfo*. If the *message* argument is null or points to an empty string, then only the signal description shall be printed. For *psiginfo*(), the argument *pinfo* references a valid **siginfo_t** structure. For *psignal*(), if *signum* is not a valid signal number, the behavior is implementation-defined.

to:

The *psiginfo*() and *psignal*() functions shall write a language-dependent message associated with a signal number to the standard error stream as follows:

- First, if *message* is not a null pointer and is not the empty string, the string pointed to by the *message* argument shall be written, followed by a <colon> and a <space>.

- Then the signal description string associated with *signum* or with the signal indicated by *pinfo* shall be written, followed by a <newline>.

For *psiginfo*(), the application shall ensure that the argument *pinfo* references a valid **siginfo_t** structure. For *psignal*(), if *signum* is not a valid signal number, the behavior is implementation-defined.

*Rationale*: Austin Group Defect Report(s) applied: 629. See http://austingroupbugs.net/view.php?id=629.

**Change Number: XSH/TC2/D4/0261** [858]

On Page: 1529 Line: 49363 Section: pthread_atfork()
(2013 edition Page: 1543 Line: 50230)

In the DESCRIPTION section, add a new paragraph:

If a *fork*() call in a multi-threaded process leads to a *child* fork handler calling any function that is not async-signal-safe, the behavior is undefined.

On Page: 1529 Line: 49377 Section: pthread_atfork()
(2013 edition Page: 1543 Line: 50244)

In the APPLICATION USAGE section, change from:

None.

to:

The original usage pattern envisaged for *pthread_atfork*() was for the *prepare* fork handler to lock mutexes and other locks, and for the *parent* and *child* handlers to unlock them. However, since all of the relevant unlocking functions, except *sem_post*(), are not async-signal-safe this usage results in undefined behavior in the child process unless the only such unlocking function it calls is *sem_post*().

On Page: 1531 Line: 49441 Section: pthread_atfork()
(2013 edition Page: 1545 Line: 50308)

In the FUTURE DIRECTIONS section, change from:

None.

to:

The *pthread_atfork*() function may be formally deprecated (for example by shading it OB) in a future revision of this standard.

*Rationale*: Austin Group Defect Report(s) applied: 858. See http://austingroupbugs.net/view.php?id=858.

**Change Number: XSH/TC2/D4/0262** [757]

On Page: 1541 Line: 49831 Section: pthread_attr_getinheritsched()
(2013 edition Page: 1555 Line: 50699)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_attr_getinheritsched*() and *pthread_attr_setinheritsched*() functions are moved from the Threads option.

to:

The *pthread_attr_getinheritsched*() and *pthread_attr_setinheritsched*() functions are marked only as part of the Thread Execution Scheduling option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0263** [757]

On Page: 1545 Line: 49961 Section: pthread_attr_getschedpolicy()
(2013 edition Page: 1559 Line: 50833)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_attr_getschedpolicy*() and *pthread_attr_setschedpolicy*() functions are moved from the Threads option.

to:

The *pthread_attr_getschedpolicy*() and *pthread_attr_setschedpolicy*() functions are marked only as part of the Thread Execution Scheduling option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0264** [757]

On Page: 1547 Line: 50023 Section: pthread_attr_getscope()
(2013 edition Page: 1561 Line: 50897)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_attr_getscope*() and *pthread_attr_setscope*() functions are moved from the Threads option.

to:

The *pthread_attr_getscope*() and *pthread_attr_setscope*() functions are marked only as part of the Thread Execution Scheduling option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0265** [757]

On Page: 1552 Line: 50161 Section: pthread_attr_getstacksize()
(2013 edition Page: 1566 Line: 51036)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_attr_getstacksize*() and *pthread_attr_setstacksize*() functions are moved from the Threads option.

to:

The *pthread_attr_getstacksize*() and *pthread_attr_setstacksize*() functions are marked only as part of the Thread Stack Size Attribute option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0266** [972]

On Page: 1562 Line: 50259-50261 Section: pthread_barrier_destroy()
(2013 edition Page: 1576 Line: 51134-51136)

In the DESCRIPTION section, change from:

Only the object referenced by *barrier* may be used for performing synchronization. The result of referring to copies of that object in calls to *pthread_barrier_destroy*() or *pthread_barrier_wait*() is undefined.

to:

See [xref to section 2.9.9] for further requirements.

On Page: 1568 Line: 50418-50421 Section: pthread_barrierattr_getpshared()
(2013 edition Page: 1582 Line: 51293-51296)

In the DESCRIPTION section, change from:

If the *process-shared* attribute is PTHREAD_PROCESS_PRIVATE, the barrier shall only be operated upon by threads created within the same process as the thread that initialized the barrier; if threads of different processes attempt to operate on such a barrier, the behavior is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0267** [757]

On Page: 1569 Line: 50460 Section: pthread_barrierattr_getpshared()
(2013 edition Page: 1583 Line: 51335)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_barrierattr_getpshared*() and *pthread_barrierattr_setpshared*() functions are moved from the Barriers option.

to:

The *pthread_barrierattr_getpshared*() and *pthread_barrierattr_setpshared*() functions are marked only as part of the Thread Process-Shared Synchronization option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0268** [624]

On Page: 1574 Line: 50552 Section: pthread_cleanup_pop()

In the DESCRIPTION section, change from:

These functions may be implemented as macros.

to:

It is unspecified whether *pthread_cleanup_push* and *pthread_cleanup_pop* are macros or functions. If a macro definition is suppressed in order to access an actual function, or a program defines an external identifier with any of these names the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 624. See http://austingroupbugs.net/view.php?id=624.

**Change Number: XSH/TC2/D4/0269** [972]

On Page: 1582 Line: 50872-50874 Section: pthread_cond_destroy()
(2013 edition Page: 1596 Line: 51751-51753)

In the DESCRIPTION section, change from:

Only *cond* itself may be used for performing synchronization. The result of referring to copies of *cond* in calls to *pthread_cond_wait*(), *pthread_cond_timedwait*(), *pthread_cond_signal*(), *pthread_cond_broadcast*(), and *pthread_cond_destroy*() is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0270** [910]

On Page: 1583 Line: 50927 Section: pthread_cond_destroy()
(2013 edition Page: 1597 Line: 51806)

In the EXAMPLES section, change from:

```
rp->notbusy
```

to:

```
ep->notbusy
```

*Rationale*: Austin Group Defect Report(s) applied: 910. See http://austingroupbugs.net/view.php?id=910.

**Change Number: XSH/TC2/D4/0271** [749]

On Page: 1587 Line: 51042 Section: pthread_cond_timedwait()

In the RETURN VALUE section, change from:

Except in the case of [ETIMEDOUT],

to:

Except for [ETIMEDOUT], [ENOTRECOVERABLE], and [EOWNERDEAD],

*Rationale*: Austin Group Defect Report(s) applied: 749. See http://austingroupbugs.net/view.php?id=749.

**Change Number: XSH/TC2/D4/0272** [972]

On Page: 1596 Line: 51368-51371 Section: pthread_condattr_getpshared()
(2013 edition Page: 1611 Line: 52252-52255)

In the DESCRIPTION section, change from:

If the *process-shared* attribute is PTHREAD_PROCESS_PRIVATE, the condition variable shall only be operated upon by threads created within the same process as the thread that initialized the condition variable; if threads of differing processes attempt to operate on such a condition variable, the behavior is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0273** [757]

On Page: 1597 Line: 51409 Section: pthread_condattr_getpshared()
(2013 edition Page: 1612 Line: 52293)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_condattr_getpshared*() and *pthread_condattr_setpshared*() functions are moved from the Threads option.

to:

The *pthread_condattr_getpshared*() and *pthread_condattr_setpshared*() functions are marked only as part of the Thread Process-Shared Synchronization option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0274** [849]

On Page: 1602 Line: 51480 Section: pthread_create()

In the APPLICATION USAGE section, change from:

through the return value of the *pthread_create*() function

to:

through the *thread* argument of the *pthread_create*() function

*Rationale*: Austin Group Defect Report(s) applied: 849. See http://austingroupbugs.net/view.php?id=849.

**Change Number: XSH/TC2/D4/0275** [757]

On Page: 1611 Line: 51797 Section: pthread_getcpuclockid()
(2013 edition Page: 1626 Line: 52682)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_getcpuclockid*() function is moved from the Threads option.

to:

The *pthread_getcpuclockid*() function is marked only as part of the Thread CPU-Time Clocks option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0276** [757]

On Page: 1614 Line: 51885 Section: pthread_getschedparam()
(2013 edition Page: 1629 Line: 52771)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_getschedparam*() and *pthread_setschedparam*() functions are moved from the Threads option.

to:

The *pthread_getschedparam*() and *pthread_setschedparam*() functions are marked only as part of the Thread Execution Scheduling option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0277** [765]

On Page: 1625 Line: 52287 Section: pthread_kill()
(2013 edition Page: 1640 Line: 53174)

In the RATIONALE section, add a paragraph:

Existing implementations vary on the result of a *pthread_kill*() with a thread ID indicating an inactive thread (a terminated thread that has not been detached or joined). Some indicate success on such a call, while others give an error of [ESRCH]. Since the definition of thread lifetime in this volume of POSIX.1-2008 covers inactive threads, the [ESRCH] error as described is inappropriate in this case. In particular, this means that an application cannot have one thread check for termination of another with *pthread_kill*().

On Page: 1625 Line: 52289 Section: pthread_kill()
(2013 edition Page: 1640 Line: 53176)

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of this standard may require that *pthread_kill*() not fail with ESRCH in the case of sending signals to an inactive thread (a terminated thread not yet detached or joined), even though no signal will be delivered because the thread is no longer running.

*Rationale*: Austin Group Defect Report(s) applied: 765. See http://austingroupbugs.net/view.php?id=765.

**Change Number: XSH/TC2/D4/0278** [811]

On Page: 1628 Line: 52362 Section: pthread_mutex_destroy()
(2013 edition Page: 1643 Line: 53249)

In the DESCRIPTION section, change from:

Attempting to destroy a locked mutex or a mutex that is referenced (for example, while being used in a *pthread_cond_timedwait*() or *pthread_cond_wait*()) by another thread results in undefined behavior.

to:

Attempting to destroy a locked mutex, or a mutex that another thread is attempting to lock, or a mutex that is being used in a *pthread_cond_timedwait*() or *pthread_cond_wait*() call by another thread, results in undefined behavior.

*Rationale*: Austin Group Defect Report(s) applied: 811. See http://austingroupbugs.net/view.php?id=811.

**Change Number: XSH/TC2/D4/0279** [972]

On Page: 1628 Line: 52369-52371 Section: pthread_mutex_destroy()
(2013 edition Page: 1643 Line: 53256-53258)

In the DESCRIPTION section, change from:

Only *mutex* itself may be used for performing synchronization. The result of referring to copies of *mutex* in calls to *pthread_mutex_lock*(), *pthread_mutex_trylock*(), *pthread_mutex_unlock*(), and *pthread_mutex_destroy*() is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0280** [811]

On Page: 1632 Line: 52537-52557 Section: pthread_mutex_destroy()
(2013 edition Page: 1647 Line: 53424-53444)

In the RATIONALE section, change from:

A mutex can be destroyed immediately after it is unlocked. For example, consider the following code:

```
struct obj {
pthread_mutex_t om;
    int refcnt;
    ...
};
obj_done(struct obj *op)
{
    pthread_mutex_lock(&op->om);
    if (--op->refcnt == 0) {
        pthread_mutex_unlock(&op->om);
(A)     pthread_mutex_destroy(&op->om);
(B)     free(op);
    } else
(C)     pthread_mutex_unlock(&op->om);
}
```

In this case *obj* is reference counted and *obj_done*() is called whenever a reference to the object is dropped. Implementations are required to allow an object to be destroyed and freed and potentially unmapped (for example, lines A and B) immediately after the object is unlocked (line C).

to:

A mutex can be destroyed immediately after it is unlocked. However, since attempting to destroy a locked mutex, or a mutex that another thread is attempting to lock, or a mutex that is being used in a *pthread_cond_timedwait*() or *pthread_cond_wait*() call by another thread, results in undefined behavior, care must be taken to ensure that no other thread may be referencing the mutex.

*Rationale*: Austin Group Defect Report(s) applied: 811. See http://austingroupbugs.net/view.php?id=811.

**Change Number: XSH/TC2/D4/0281** [972]

On Page: 1657 Line: 53355-53358 Section: pthread_mutexattr_getpshared()
(2013 edition Page: 1672 Line: 54249-54252)

In the DESCRIPTION section, change from:

If the *process-shared* attribute is PTHREAD_PROCESS_PRIVATE, the mutex shall only be operated upon by threads created within the same process as the thread that initialized the mutex; if threads of differing processes attempt to operate on such a mutex, the behavior is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0282** [757]

On Page: 1658 Line: 53396 Section: pthread_mutexattr_getpshared()
(2013 edition Page: 1673 Line: 54290)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_mutexattr_getpshared*() and *pthread_mutexattr_setpshared*() functions are moved from the Threads option.

to:

The *pthread_mutexattr_getpshared*() and *pthread_mutexattr_setpshared*() functions are marked only as part of the Thread Process-Shared Synchronization option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0283** [748]

On Page: 1659 Line: 53421 Section: pthread_mutexattr_getrobust()
(2013 edition Page: 1744 Line: 54315)

In the DESCRIPTION section, change from:

... the next thread that acquires the mutex may be notified about the termination by the return value [EOWNERDEAD]. The notified thread can then attempt to mark the state protected by the mutex as consistent again by a call to *pthread_mutex_consistent*().

to:

... the next thread that attempts to acquire the mutex may be notified about the termination by the return value [EOWNERDEAD]. The notified thread can then attempt to make the state protected by the mutex consistent again, and if successful can mark the mutex state as consistent by calling *pthread_mutex_consistent*().

*Rationale*: Austin Group Defect Report(s) applied: 748. See http://austingroupbugs.net/view.php?id=748.

**Change Number: XSH/TC2/D4/0284** [863]

On Page: 1669 Line: 53609 Section: pthread_once()
(2013 edition Page: 1684 Line: 54487)

In the DESCRIPTION section, add new paragraph:

If the call to *init_routine* is terminated by a call to *longjmp*(), *_longjmp*(), or *siglongjmp*(), the behavior is undefined.

On Page: 1669 Line: 53621 Section: pthread_once()
(2013 edition Page: 1684 Line: 54499)

In the APPLICATION USAGE section, change from:

None.

to:

If *init_routine* recursively calls *pthread_once*() with the same *once_control*, the recursive call will not call the specified *init_routine*, and thus the specified *init_routine* will not complete, and thus the recursive call

to *pthread_once*() will not return. Use of *longjmp*(), *_longjmp*(), or *siglongjmp*() within an *init_routine* to jump to a point outside of *init_routine* prevents *init_routine* from returning.

*Rationale*: Austin Group Defect Report(s) applied: 863. See http://austingroupbugs.net/view.php?id=863.

**Change Number: XSH/TC2/D4/0285** [874]

On Page: 1669 Line: 53628 Section: pthread_once()
(2013 edition Page: 1684 Line: 54506)

In the RATIONALE section, change from:

```
extern int initialize_random();
```

to:

```
extern void initialize_random(void);
```

*Rationale*: Austin Group Defect Report(s) applied: 874. See http://austingroupbugs.net/view.php?id=874.

**Change Number: XSH/TC2/D4/0286** [874]

On Page: 1670 Line: 53640 Section: pthread_once()
(2013 edition Page: 1685 Line: 54518)

In the RATIONALE section, change from:

For dynamic library initialization in a multi-threaded process, a simple initialization flag is not sufficient; the flag needs to be protected against modification by multiple threads simultaneously calling into the library. Protecting the flag requires the use of a mutex; however, mutexes have to be initialized before they are used. Ensuring that the mutex is only initialized once requires a recursive solution to this problem.

The use of *pthread_once*() not only supplies an implementation-guaranteed means of dynamic initialization, it provides an aid to the reliable construction of multi-threaded and realtime systems. The preceding example then becomes:

to:

For dynamic library initialization in a multi-threaded process, if an initialization flag is used the flag needs to be protected against modification by multiple threads simultaneously calling into the library. This can be done by using a mutex (initialized by assigning PTHREAD_MUTEX_INITIALIZER). However, the better solution is to use *pthread_once*(), which is designed for exactly this purpose, as follows:

On Page: 1670 Line: 53650 Section: pthread_once()
(2013 edition Page: 1685 Line: 54528)

In the RATIONALE section, change from:

```
extern int initialize_random();
```

to:

```
extern void initialize_random(void);
```

*Rationale*: Austin Group Defect Report(s) applied: 874. See http://austingroupbugs.net/view.php?id=874.

**Change Number: XSH/TC2/D4/0287** [747]

On Page: 1670 Line: 53656 Section: pthread_once()

In the RATIONALE section, remove the text:

Note that a **pthread_once_t** cannot be an array because some compilers do not accept the construct &<**array_name**>.

*Rationale*: Austin Group Defect Report(s) applied: 747. See http://austingroupbugs.net/view.php?id=747.

**Change Number: XSH/TC2/D4/0288** [972]

On Page: 1671 Line: 53698-53701 Section: pthread_rwlock_destroy()
(2013 edition Page: 1686 Line: 54576-54579)

In the DESCRIPTION section, change from:

Only the object referenced by *rwlock* may be used for performing synchronization. The result of referring to copies of that object in calls to *pthread_rwlock_destroy*(), *pthread_rwlock_rdlock*(), *pthread_rwlock_timedrdlock*(), *pthread_rwlock_timedwrlock*(), *pthread_rwlock_tryrdlock*(), *pthread_rwlock_trywrlock*(), *pthread_rwlock_unlock*(), or *pthread_rwlock_wrlock*() is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0289** [758]

On Page: 1672 Line: 53728 Section: pthread_rwlock_destroy()
(2013 edition Page: 1687 Line: 54606)

In the RATIONALE section, change from:

*pthread_rwlockr_init*()

to:

*pthread_rwlock_init*()

*Rationale*: Austin Group Defect Report(s) applied: 758. See http://austingroupbugs.net/view.php?id=758.

**Change Number: XSH/TC2/D4/0290** [720]

On Page: 1682 Line: 54002 Section: pthread_rwlock_trywrlock()

In the DESCRIPTION section, change from:

The calling thread acquires the write lock if no other thread (reader or writer) holds the read-write lock *rwlock*. Otherwise, the thread shall block until it can acquire the lock. The calling thread may deadlock if at the time the call is made it holds the read-write lock (whether a read or write lock).

to:

The calling thread shall acquire the write lock if no thread (reader or writer) holds the read-write lock *rwlock*. Otherwise, if another thread holds the read-write lock *rwlock*, the calling thread shall block until it can acquire the lock. If a deadlock condition occurs or the calling thread already owns the read-write lock for writing or reading, the call shall either deadlock or return EDEADLK.

*Rationale*: Austin Group Defect Report(s) applied: 720. See http://austingroupbugs.net/view.php?id=720. It is clear from the description of pthread_rwlock_[try]rdlock() and pthread_rwlock_unlock() that read locks are recursive. It is clear from the description of pthread_rwlock_trywrlock() and pthread_rwlock_unlock() that write locks are not recursive. The intention is that pthread_rwlock_wrlock() either deadlocks or returns EDEADLK if the caller already holds a write lock, but this is written as two separate "may" clauses (one in the DESCRIPTION and one in the ERRORS section), which creates a loophole that allows pthread_rwlock_wrlock() to succeed (as a no-op). However, this behavior would be of no use to applications. In particular, if an application tried to make matching pthread_rwlock_unlock() calls for two successful pthread_rwlock_wrlock() calls, the first unlock call would set the state to unlocked and the second unlock call would result in undefined behavior.

**Change Number: XSH/TC2/D4/0291** [722]

On Page: 1682 Line: 54006 Section: pthread_rwlock_trywrlock()
(2013 edition Page: 1697 Line: 54885)

In the DESCRIPTION section, remove the line:

Implementations may favor writers over readers to avoid writer starvation.

*Rationale*: Austin Group Defect Report(s) applied: 722. See http://austingroupbugs.net/view.php?id=722. Changes were made to the read locks in Issue 6, that should also be represented in the write lock specification regarding read/write precedence.

**Change Number: XSH/TC2/D4/0292** [972]

On Page: 1689 Line: 54196-54199 Section: pthread_rwlockattr_getpshared()
(2013 edition Page: 1704 Line: 55075-55078)

In the DESCRIPTION section, change from:

If the *process-shared* attribute is PTHREAD_PROCESS_PRIVATE, the read-write lock shall only be operated upon by threads created within the same process as the thread that initialized the read-write lock; if threads of differing processes attempt to operate on such a read-write lock, the behavior is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0293** [757]

On Page: 1690 Line: 54240 Section: pthread_rwlockattr_getpshared()
(2013 edition Page: 1705 Line: 55119)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_rwlockattr_getpshared*() and *pthread_rwlockattr_setpshared*() functions are moved from the Threads option.

to:

The *pthread_rwlockattr_getpshared*() and *pthread_rwlockattr_setpshared*() functions are marked only as part of the Thread Process-Shared Synchronization option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0294** [622]

On Page: 1694 Line: 54327 Section: pthread_setcancelstate()

In the APPLICATION USAGE section, change from:

None.

to:

In order to write a signal handler for an asynchronous signal which can run safely in a cancellable thread, *pthread_setcancelstate*() must be used to disable cancellation for the duration of any calls that the signal handler makes which are cancellation points. However, the standard does not permit strictly conforming applications to call *pthread_setcancelstate*() from a signal handler since it is not currently required to be async-signal-safe. On implementations where *pthread_setcancelstate*() is not async-signal-safe, alternatives are to ensure either that the corresponding signals are blocked during execution of functions that are not async-cancel-safe or that cancellation is disabled during times when those signals could be delivered. Implementations are strongly encouraged to make *pthread_setcancelstate*() async-signal-safe.

*Rationale*: Austin Group Defect Report(s) applied: 622. See http://austingroupbugs.net/view.php?id=622.

**Change Number: XSH/TC2/D4/0295** [615]

On Page: 1695 Line: 54349 Section: pthread_setcancelstate()

In the FUTURE DIRECTIONS section, change from:

None.

to:

The *pthread_setcancelstate*() function may be added to the table of async-signal-safe functions in section 2.4.3 on page 489.

*Rationale*: Austin Group Defect Report(s) applied: 615. See http://austingroupbugs.net/view.php?id=615. In order to write a signal handler for an asynchronous signal which can run safely in a cancellable thread, pthread_setcancelstate() must be used to disable cancellation for the duration of any calls that the signal handler makes which are cancellation points. However, the standard does not currently permit strictly conforming applications to do this since pthread_setcancelstate() is not required to be async-signal-safe.

**Change Number: XSH/TC2/D4/0296** [757]

On Page: 1699 Line: 54427 Section: pthread_setschedprio()
(2013 edition Page: 1714 Line: 55305)

In the CHANGE HISTORY section for Issue 7, change from:

The *pthread_setschedprio*() function is moved from the Threads option.

to:

The *pthread_setschedprio*() function is marked only as part of the Thread Execution Scheduling option as the Threads option is now part of the Base.

*Rationale*: Austin Group Defect Report(s) applied: 757. See http://austingroupbugs.net/view.php?id=757.

**Change Number: XSH/TC2/D4/0297** [972]

On Page: 1705 Line: 54589-54593 Section: pthread_spin_destroy()
(2013 edition Page: 1720 Line: 55469-55473)

In the DESCRIPTION section, change from:

[TSH]If the Thread Process-Shared Synchronization option is supported and the value of *pshared* is PTHREAD_PROCESS_PRIVATE,[/TSH] or if the option is not supported, the spin lock shall only be operated upon by threads created within the same process as the thread that initialized the spin lock. If threads of differing processes attempt to operate on such a spin lock, the behavior is undefined.

to:

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0298** [503]

On Page: 1712 Line: 54750 Section: ptsname()

In the RETURN VALUE section, change from:

Upon failure *ptsname*() shall return a null pointer.

to:

Upon failure *ptsname*() shall return a null pointer and may set *errno*.

*Rationale*: Austin Group Defect Report(s) applied: 503. See http://austingroupbugs.net/view.php?id=503.

**Change Number: XSH/TC2/D4/0299** [656]

On Page: 1712 Line: 54752 Section: ptsname()

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

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 *ptsname*(). The returned pointer and the string content might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0468 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P1727, L55633-55634:

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

Note that the other change in XSH/TC1/D5/0468 [75] to the APPLICATION USAGE section is not impacted.

**Change Number: XSH/TC2/D4/0300** [503]

On Page: 1712 Line: 54754 Section: ptsname()

In the ERRORS section, change from:

No errors are defined.

to:

The *ptsname*() function may fail if:

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

[ENOTTY]
The file associated with the *fildes* argument is not a master pseudo-terminal device.

*Rationale*: Austin Group Defect Report(s) applied: 503. See http://austingroupbugs.net/view.php?id=503.

**Change Number: XSH/TC2/D4/0301** [743]

On Page: 1734 Line: 55339 Section: rand()
(2013 edition Page: 1749 Line: 56250)

In the APPLICATION USAGE section, change from:

The *drand48*() function provides a much more elaborate random number generator.

to:

The *drand48*() and *random*() functions provide much more elaborate pseudo-random number generators.

On Page: 1734 Line: 55342-55343 Section: rand()
(2013 edition Page: 1749 Line: 56253-56254)

In the APPLICATION USAGE section, change from:

Therefore this function should be avoided whenever non-trivial requirements (including safety) have to be fulfilled.

to a new paragraph:

These functions should be avoided whenever non-trivial requirements (including safety) have to be fulfilled.

On Page: 1735 Line: 55362 Section: rand()
(2013 edition Page: 1750 Line: 56273)

In the SEE ALSO section, add *initstate*().

*Rationale*: Austin Group Defect Report(s) applied: 743. See http://austingroupbugs.net/view.php?id=743.

**Change Number: XSH/TC2/D4/0302** [710]

On Page: 1739 Line: 55477 Section: read()

In the DESCRIPTION section, change from:

The *pread*() function shall be equivalent to *read*(), except that it shall read from a given position in the file without changing the file pointer.

to:

The *pread*() function shall be equivalent to *read*(), except that it shall read from a given position in the file without changing the file offset.

*Rationale*: Austin Group Defect Report(s) applied: 710. See http://austingroupbugs.net/view.php?id=710.

**Change Number: XSH/TC2/D4/0303** [676,710]

On Page: 1739 Line: 55506 Section: read()
(2013 edition Page: 1754 Line: 56422)

In the ERRORS section, before "The *read*() function shall fail if" insert:

The *pread*() function shall fail if:

[EINVAL]
The file is a regular file or block special file, and the offset argument is negative. The file offset shall remain unchanged.

[ESPIPE]
The file is incapable of seeking.

On Page: 1740 Line: 55520 Section: read()

In the ERRORS section, delete lines 55520-55525 (the pread() "shall fail" errors).

*Rationale*: Austin Group Defect Report(s) applied: 676,710. See
http://austingroupbugs.net/view.php?id=676.
See http://austingroupbugs.net/view.php?id=710.
This is a layered change on XSH/TC1/D5/0484 [218] and XSH/TC1/D5/0482 [218].

**Change Number: XSH/TC2/D4/0304** [656]

On Page: 1744 Line: 55685 Section: readdir()

In the DESCRIPTION section, change from:

The pointer returned by *readdir*() points to data which may be overwritten by another call to *readdir*() on the same directory stream. This data is not overwritten by another call to *readdir*() on a different directory stream.

to:

The application shall not modify the structure to which the return value of *readdir*() 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 *readdir*() on the same directory stream. They shall not be affected by a call to *readdir*() on a different directory stream. The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0486 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P1759, L56601-56605:

The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0305** [591]

On Page: 1749 Line: 55846 Section: readlink()

Before the readlinkat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0306** [817]

On Page: 1749 Line: 55858 Section: readlink()
(2013 edition Page: 1764 Line: 56779)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0307** [817]

On Page: 1750 Line: 55885 Section: readlink()
(2013 edition Page: 1765 Line: 56805)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0308** [591]

On Page: 1751 Line: 55930 Section: readlink()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0309** [526]

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 behavior is undefined if the returned pointer is used to access an object.

*Rationale*: Austin Group Defect Report(s) applied: 526. See http://austingroupbugs.net/view.php?id=526.

This change is layered on XSH/TC1/D5/0495 applied in TC1, the difference being that bug report 526 changes from:
except that the returned pointer shall not be used to access an object.
to:
except that the behavior is undefined if the returned pointer is used to access an object.

**Change Number: XSH/TC2/D4/0310** [526,688]

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, if *ptr* is not a null pointer, *errno* shall be set to an implementation defined value[/CX].
- A pointer to the allocated space 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: 526,688. See http://austingroupbugs.net/view.php?id=526.
See http://austingroupbugs.net/view.php?id=688.
This change is layered on XSH/TC1/D5/0496 [400] applied in TC1, the difference being that bug report 526 changes from:
A unique pointer that can be successfully passed to *free*() shall be returned
to:
A pointer to the allocated space shall be returned

This is also a layered change on XSH/TC1/D5/0496 [400] as a result of bug report 688. The difference being that the wording is changed from:
A null pointer shall be returned [CX]and *errno* set to an implementation-defined value[/CX].
to:
A null pointer shall be returned [CX]and, if *ptr* is not a null pointer, *errno* shall set to an implementation-defined value[/CX].

**Change Number: XSH/TC2/D4/0311** [873]

On Page: 1781 Line: 56918 Section: rename()
(2013 edition Page: 1797 Line: 57881)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0312** [591]

On Page: 1781 Line: 56922 Section: rename()

Before the renameat SYNOPSIS line, insert a line with OH CX shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0313** [716]

On Page: 1781 Line: 56939 Section: rename()

In the DESCRIPTION section, change from:

In this case, a link named *new* shall remain visible to other processes throughout the renaming operation

to:

In this case, a link named *new* shall remain visible to other threads throughout the renaming operation

*Rationale*: Austin Group Defect Report(s) applied: 716. See http://austingroupbugs.net/view.php?id=716.

**Change Number: XSH/TC2/D4/0314** [817]

On Page: 1782 Line: 56969 Section: rename()
(2013 edition Page: 1798 Line: 57932)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0315** [817]

On Page: 1783 Line: 57023 Section: rename()
(2013 edition Page: 1799 Line: 57990)

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

*oldfd* or *newfd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *oldfd* or *newfd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0316** [591]

On Page: 1784 Line: 57083 Section: rename()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0317** [870]

On Page: 1812 Line: 57894 Section: sem_close()
(2013 edition Page: 1827 Line: 58865)

In the DESCRIPTION section, change from:

The effect of subsequent use of the semaphore indicated by *sem* by this process is undefined.

to:

The effect of subsequent use of the semaphore indicated by *sem* by this process is undefined. If any threads in the calling process are currently blocked on the semaphore, the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 870. See http://austingroupbugs.net/view.php?id=870. There are current implementations which return *mmap*()ed memory and therefore a *sem_close*() in this situation on such implementations will result in undefined behavior.

**Change Number: XSH/TC2/D4/0318** [972]

On Page: 1818 Line: 58029-58031 Section: sem_init()
(2013 edition Page: 1832 Line: 59000-59002)

In the DESCRIPTION section, delete:

Only sem itself may be used for performing synchronization. The result of referring to copies of sem in calls to *sem_wait*(), *sem_timedwait*(), *sem_trywait*(), *sem_post*(), and *sem_destroy*() is undefined.

On Page: 1818 Line: 58034-58035 Section: sem_init()
(2013 edition Page: 1832 Line: 59005-59006)

In the DESCRIPTION section, change from:

The use of the semaphore by threads other than those created in the same process is undefined.

to (as a new paragraph):

See [xref to section 2.9.9] for further requirements.

*Rationale*: Austin Group Defect Report(s) applied: 972. See http://austingroupbugs.net/view.php?id=972.

**Change Number: XSH/TC2/D4/0319** [532]

On Page: 1833 Line: 58506 Section: semctl()

In the DESCRIPTION section add the following new paragraph after the union semun definition:

Each operation shall be performed atomically.

*Rationale*: Austin Group Defect Report(s) applied: 532. See http://austingroupbugs.net/view.php?id=532.

**Change Number: XSH/TC2/D4/0320** [899]

On Page: 1867 Line: 59571 Section: setkey()
(2013 edition Page: 1882 Line: 60535)

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of the standard may mark this interface as obsolete or remove it altogether.

*Rationale*: Austin Group Defect Report(s) applied: 899. See http://austingroupbugs.net/view.php?id=899.

**Change Number: XSH/TC2/D4/0321** [826]

On Page: 1869 Line: 59629 Section: setlocale()
(2013 edition Page: 1884 Line: 60595)

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

[CX]The *setlocale*() function need not be thread-safe.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 826. See http://austingroupbugs.net/view.php?id=826.

**Change Number: XSH/TC2/D4/0322** [826]

On Page: 1869 Line: 59637 Section: setlocale()
(2013 edition Page: 1884 Line: 60604)

In the RETURN VALUE section, change from:

The application shall not modify the string returned which may be overwritten by a subsequent call to setlocale().

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 setlocale(). [CX]The returned pointer might also be invalidated if the calling thread is terminated.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 826. See http://austingroupbugs.net/view.php?id=826.
This is a layered change, building upon XSH/TC1/D5/0567 [288], the difference being the addition of:
[CX]The returned pointer might also be invalidated if the calling thread is terminated.[/CX]

**Change Number: XSH/TC2/D4/0323** [596]

On Page: 1871 Line: 59720 Section: setlocale()

In the SEE ALSO section, remove the setlocale() entry.

*Rationale*: Austin Group Defect Report(s) applied: 596. See http://austingroupbugs.net/view.php?id=596.

**Change Number: XSH/TC2/D4/0324** [835]

On Page: 1898 Line: 60324 Section: shm_open()
(2013 edition Page: 1913 Line: 61308)

Change from:

The file descriptor is used by other functions to refer to that shared memory object.

to:

The file descriptor shall be allocated as described in [xref to new section 2.14] and can be used by other functions to refer to that shared memory object.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0325** [835]

On Page: 1898 Line: 60335 Section: shm_open()
(2013 edition Page: 1913 Line: 61319)

Change from:

... return a file descriptor for the shared memory object that is the lowest numbered file descriptor not currently open for that process.

to:

... return a file descriptor for the shared memory object.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0326** [835]

On Page: 1899 Line: 60375 Section: shm_open()
(2013 edition Page: 1914 Line: 61358)

Change from:

... return a non-negative integer representing the lowest numbered unused file descriptor.

to:

... return a non-negative integer representing the file descriptor.

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0327** [522]

On Page: 1905 Line: 60591 Section: shmat()

In the RETURN VALUE section, change from:

*shmat*() shall return -1

to:

*shmat*() shall return (**void***)-1

*Rationale*: Austin Group Defect Report(s) applied: 522. See http://austingroupbugs.net/view.php?id=522.

**Change Number: XSH/TC2/D4/0328** [640]

On Page: 1912 Line: 60779-60780 Section: shmget()

In the ERRORS section, change from:

A shared memory identifier and associated shared memory segment shall be created, but the amount of available physical memory is not sufficient to fill the request.

to:

A shared memory identifier and associated shared memory segment are to be created, but the amount of available physical memory is not sufficient to fill the request.

*Rationale*: Austin Group Defect Report(s) applied: 640. See http://austingroupbugs.net/view.php?id=640.

**Change Number: XSH/TC2/D4/0329** [690]

On Page: 1917 Line: 60935 Section: sigaction()
(2013 edition Page: 1932 Line: 61931)

In the DESCRIPTION section change from:

SA_NOCLDWAIT: If set, and sig equals SIGCHLD, child processes of the calling processes shall not be transformed into zombie processes when they terminate. If the calling process subsequently waits for its children, and the process has no unwaited-for children that were transformed into zombie processes, it shall block until all of its children terminate, and *wait*(), *waitid*(), and *waitpid*() shall fail and set *errno* to [ECHILD]. Otherwise, terminating child processes shall be transformed into zombie processes, unless SIGCHLD is set to SIG_IGN.

to:

[XSI]SA_NOCLDWAIT: If sig does not equal SIGCHLD, the behavior is unspecified. Otherwise, the behavior of the SA_NOCLDWAIT flag is as specified under "Consequences of Process Termination" in the description of the *_Exit*() function on page 549.[/XSI]

(Note the addition of XSI shading.)

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0330** [491]

On Page: 1918 Line: 60981 Section: sigaction()

In the ERRORS section, delete the [ENOTSUP] error:

[ENOTSUP]

The SA_SIGINFO bit flag is set in the *sa_flags* field of the **sigaction** structure.

On Page: 1918 Line: 60985 Section: sigaction()

In the ERRORS section, change from:

In addition, the *sigaction*() function may fail if the SA_SIGINFO flag is set ...

to:

In addition, on systems that do not support the XSI option, the *sigaction*() function may fail if the SA_SIGINFO flag is set ...

*Rationale*: Austin Group Defect Report(s) applied: 491. See http://austingroupbugs.net/view.php?id=491.

**Change Number: XSH/TC2/D4/0331** [785]

On Page: 1937 Line: 61615 Section: signal()

In the DESCRIPTION section, delete the paragraph:

Use of this function is unspecified in a multi-threaded process.

*Rationale*: Austin Group Defect Report(s) applied: 785. See http://austingroupbugs.net/view.php?id=785.

**Change Number: XSH/TC2/D4/0332** [844]

On Page: 1944 Line: 61771 Section: sigqueue()
(2013 edition Page: 1959 Line: 62775)

In the SYNOPSIS section, change from:

```
const union sigval value;
```

to:

```
union sigval value;
```

*Rationale*: Austin Group Defect Report(s) applied: 844. See http://austingroupbugs.net/view.php?id=844.

**Change Number: XSH/TC2/D4/0333** [815]

On Page: 1951 Line: 61997 Section: sigtimedwait()

In the DESCRIPTION section, change from:

The *sigwaitinfo*() function shall be equivalent to the *sigwait*() function if the *info* argument is NULL. If the *info* argument is non-NULL, the *sigwaitinfo*() function shall be equivalent to *sigwait*(), except that the selected signal number ...

to:

The *sigwaitinfo*() function shall be equivalent to the *sigwait*() function, except that the return value and the error reporting method are different (see RETURN VALUE), and that if the *info* argument is non-NULL, the selected signal number ...

*Rationale*: Austin Group Defect Report(s) applied: 815. See http://austingroupbugs.net/view.php?id=815.

**Change Number: XSH/TC2/D4/0334** [625]

On Page: 1963 Line: 62339 Section: sleep()

In the DESCRIPTION section, add a new paragraph after the first paragraph:

In single-threaded programs, *sleep*() may make use of SIGALRM. In multi-threaded programs, *sleep*() shall not make use of SIGALRM and the remainder of this DESCRIPTION does not apply.

In the RATIONALE section on page 1963 line 62369 change from:

This volume of POSIX.1-2008 permits either approach.

to:

This volume of POSIX.1-2008 permits either approach in single-threaded programs, but the simple alarm/suspend implementation is not appropriate for multi-threaded programs.

On Page: 1964 Line: 62406 Section: sleep()

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of this standard may require that *sleep*() does not make use of SIGALRM in all programs, not just multi-threaded programs.

*Rationale*: Austin Group Defect Report(s) applied: 625. See http://austingroupbugs.net/view.php?id=625.

**Change Number: XSH/TC2/D4/0335** [835]

On Page: 1968 Line: 62491 Section: socket()
(2013 edition Page: 1983 Line: 63518)

After the text:

... return a file descriptor that can be used in later function calls that operate on sockets.

add:

The file descriptor shall be allocated as described in [xref to new section 2.14].

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0336** [835]

On Page: 1970 Line: 62565 Section: socketpair()
(2013 edition Page: 1985 Line: 63592)

After the text:

The file descriptors used in referencing the created sockets shall be returned in *socket_vector*[0] and *socket_vector*[1].

add:

The file descriptors shall be allocated as described in [xref to new section 2.14].

*Rationale*: Austin Group Defect Report(s) applied: 835. See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0337** [483,835]

On Page: 1970 Line: 62593 Section: socketpair()
(2013 edition Page: 1985 Line: 63619)

Change from:

... otherwise, -1 shall be returned and *errno* set to indicate the error.

to:

... otherwise, -1 shall be returned and *errno* set to indicate the error, no file descriptors shall be allocated and the contents of *socket_vector* shall be left unmodified.

*Rationale*: Austin Group Defect Report(s) applied: 483,835. See http://austingroupbugs.net/view.php?id=483.
See http://austingroupbugs.net/view.php?id=835.

**Change Number: XSH/TC2/D4/0338** [738]

On Page: 1997 Line: 63203 Section: strdup()
(2013 edition Page: 2012 Line: 64235)

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

Implementations are free to *malloc*() a buffer containing either (*size* + 1) bytes or (*strnlen*(s, *size*) + 1) bytes. Applications should not assume that *strndup*() will allocate (*size* + 1) bytes when *strlen*(s) is smaller than *size*.

*Rationale*: Austin Group Defect Report(s) applied: 738. See http://austingroupbugs.net/view.php?id=738.

**Change Number: XSH/TC2/D4/0339** [656]

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. The returned pointer and the string content might also be invalidated if the calling thread is terminated.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0595 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P2014, L64269-64271:

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0340** [584]

On Page: 2007 Line: 63525 Section: strftime()

In the DESCRIPTION section, change from:

minus-sign character ('-')

to:

<hyphen-minus> character ('-')

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0341** [796]

On Page: 2012 Line: 63745 Section: strftime()
(2013 edition Page: 2027 Line: 64791)

In the APPLICATION USAGE section, change from:

In the C locale, the E and O modifiers are ignored and the replacement strings for the following

to:

In the C or POSIX locale, the E and O modifiers are ignored and the replacement strings for the following

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0342** [584]

On Page: 2014 Line: 63807 Section: strftime()

In the RATIONALE section, change from:

leading minus-sign for %F,

to:

leading <hyphen-minus> for %F,

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0343** [584]

On Page: 2014 Line: 63815 Section: strftime()

In the RATIONALE section, change from:

leading minus-sign ('-') when using %Y

to:

leading <hyphen-minus> ('-') when using %Y

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0344** [560]

On Page: 2016 Line: 63886 Section: strlen()

In the DESCRIPTION section, change from:

the terminating NUL character

to:

any terminating NUL character

On Page: 2016 Line: 63891 Section: strlen()

In the RETURN VALUE section, change from:

The *strnlen*() function shall return an integer containing the smaller of either the length of the string pointed to by *s* or *maxlen*.

to:

The *strnlen*() function shall return the number of bytes preceding the first null byte in the array to which *s* points, if *s* contains a null byte within the first *maxlen* bytes; otherwise, it shall return *maxlen*.

*Rationale*: Austin Group Defect Report(s) applied: 560. See http://austingroupbugs.net/view.php?id=560.

**Change Number: XSH/TC2/D4/0345** [920]

On Page: 2026 Line: 64133 Section: strptime()
(2013 edition Page: 2041 Line: 65181)

In the DESCRIPTION section, change from:

... conversion specifier other than C, F, or Y.

to:

... conversion specifier other than C or Y.

*Rationale*: Austin Group Defect Report(s) applied: 920. See http://austingroupbugs.net/view.php?id=920.
The standard states that the behavior of field widths is unspecified for conversions other than C, F, or Y, but does not specify an F conversion.

**Change Number: XSH/TC2/D4/0346** [919]

On Page: 2029 Line: 64233 Section: strptime()
(2013 edition Page: 2044 Line: 65281)

In the EXAMPLES section, change from:

**Data-Plus-Time**

to:

**Date-Plus-Time**

*Rationale*: Austin Group Defect Report(s) applied: 919. See http://austingroupbugs.net/view.php?id=919.

**Change Number: XSH/TC2/D4/0347** [656]

On Page: 2032 Line: 64344 Section: strsignal()

In the DESCRIPTION section, change from:

The string pointed to shall not be modified by the application, but may be overwritten by a subsequent call to *strsignal*() or *setlocale*().

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 *strsignal*() or *setlocale*(). The returned pointer might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656.
This item is a layered change on XSH/TC1/D5/0609 [75].

The change is to add the following text to the end of the paragraph at 2013 edition P2047, L65392-65393:

The returned pointer might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0348** [584]

On Page: 2036 Line: 64482 Section: strtod()

In the DESCRIPTION section, change from:

If the subject sequence begins with a minus-sign,

to:

If the subject sequence begins with a <hyphen-minus>,

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0349** [796]

On Page: 2036 Line: 64495 Section: strtod()
(2013 edition Page: 2051 Line: 65544)

Change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0350** [878]

On Page: 2040 Line: 64631-64633 Section: strtok()
(2013 edition Page: 2056 Line: 65689-65691)

In the SYNOPSIS section, change from:

```
char *strtok(char *restrict s1, const char *restrict s2);
char *strtok_r(char *restrict s, const char *restrict sep,
    char **restrict lasts);
```

to:

```
char *strtok(char *restrict s, const char *restrict sep);
char *strtok_r(char *restrict s, const char *restrict sep,
    char **restrict state);
```

On Page: 2040 Line: 64638-64650 Section: strtok()
(2013 edition Page: 2056 Line: 65696-65708)

In the second, third and fourth paragraphs of the DESCRIPTION section, replace all instances of *s1* with *s* and *s2* with *sep*.

On Page: 2040 Line: 64656-64667 Section: strtok()
(2013 edition Page: 2056 Line: 65714-65725)

In the DESCRIPTION section, change from:

The *strtok_r*() function considers the null-terminated string *s* as a sequence of zero or more text tokens separated by spans of one or more characters from the separator string *sep*. The argument *lasts* points to a user-provided pointer which points to stored information necessary for *strtok_r*() to continue scanning the same string.

In the first call to *strtok_r*(), *s* points to a null-terminated string, *sep* to a null-terminated string of separator characters, and the value pointed to by *lasts* is ignored. The *strtok_r*() function shall return a pointer to the first character of the first token, write a null character into *s* immediately following the returned token, and update the pointer to which *lasts* points.

In subsequent calls, *s* is a null pointer and *lasts* shall be unchanged from the previous call so that subsequent calls shall move through the string *s*, returning successive tokens until no tokens remain. The separator string *sep* may be different from call to call. When no token remains in *s*, a null pointer shall be returned.

to:

The *strtok_r*() function shall be equivalent to *strtok*(), except that *strtok_r*() shall be thread-safe and the argument *state* points to a user-provided pointer that allows *strtok_r*() to maintain state between calls which scan the same string. The application shall ensure that the pointer pointed to by *state* is unique for each string (*s*) being processed concurrently by *strtok_r*() calls. The application need not initialize the pointer pointed to by *state* to any particular value. The implementation shall not update the pointer pointed to by *state* to point (directly or indirectly) to resources, other than within the string *s*, that need to be freed or released by the caller.

On Page: 2041 Line: 64704 Section: strtok()
(2013 edition Page: 2057 Line: 65766)

In the APPLICATION USAGE section, insert a new paragraph:

Note that if *sep* is the empty string, *strtok*() and *strtok_r*() return a pointer to the remainder of the string being tokenized.

*Rationale*: Austin Group Defect Report(s) applied: 878. See http://austingroupbugs.net/view.php?id=878. The behavior of *strtok_r*() with an empty *sep* is unclear.

**Change Number: XSH/TC2/D4/0351** [892]

On Page: 2043 Line: 64737 Section: strtol()
(2013 edition Page: 2059 Line: 65800)

In the SYNOPSIS section, change from:

```
long strtol(const char *restrict str, char **restrict endptr, int base);
long long strtoll(const char *restrict str, char **restrict endptr,
    int base)
```

to:

```
long strtol(const char *restrict nptr, char **restrict endptr, int base);
long long strtoll(const char *restrict nptr, char **restrict endptr,
    int base)
```

On Page: 2043 Line: 64744 Section: strtol()
(2013 edition Page: 2059 Line: 65807)

In the DESCRIPTION section, change from:

These functions shall convert the initial portion of the string pointed to by *str* ...

to:

These functions shall convert the initial portion of the string pointed to by *nptr* ...

*Rationale*: Austin Group Defect Report(s) applied: 892. See http://austingroupbugs.net/view.php?id=892.

**Change Number: XSH/TC2/D4/0352** [584]

On Page: 2043 Line: 64772 Section: strtol()

In the DESCRIPTION section, change from:

begins with a minus-sign,

to:

begins with a <hyphen-minus>,

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0353** [796]

On Page: 2043 Line: 64775 Section: strtol()
(2013 edition Page: 2059 Line: 65838)

In the DESCRIPTION section, change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0354** [892]

On Page: 2044 Line: 64778 Section: strtol()
(2013 edition Page: 2060 Line: 65841)

In the DESCRIPTION section, change from:

... the value of *str* is stored ...

to:

... the value of *nptr* shall be stored ...

*Rationale*: Austin Group Defect Report(s) applied: 892. See http://austingroupbugs.net/view.php?id=892.
This item is a layered change on XSH/TC1/D5/0616 [453].

**Change Number: XSH/TC2/D4/0355** [584]

On Page: 2048 Line: 64876 Section: strtoul()

In the DESCRIPTION section, change from:

begins with a minus-sign,

to:

begins with a <hyphen-minus>,

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0356** [796]

On Page: 2049 Line: 64879 Section: strtoul()
(2013 edition Page: 2065 Line: 65950)

In the DESCRIPTION section, change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0357** [873]

On Page: 2057 Line: 65044 Section: symlink()
(2013 edition Page: 2073 Line: 66122)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0358** [591]

On Page: 2057 Line: 65048 Section: symlink()

Before the symlinkat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0359** [641]

On Page: 2057 Line: 65053 Section: symlink()

In the DESCRIPTION section, change from:

The string pointed to by *path1* shall be treated only as a character string and shall not be validated as a pathname.

to:

The string pointed to by *path1* shall be treated only as a string and shall not be validated as a pathname.

*Rationale*: Austin Group Defect Report(s) applied: 641. See http://austingroupbugs.net/view.php?id=641.

**Change Number: XSH/TC2/D4/0360** [817]

On Page: 2057 Line: 65073 Section: symlink()
(2013 edition Page: 2073 Line: 66152)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0361** [822]

On Page: 2058 Line: 65096 Section: symlink()

In the RETURN VALUE section, add after the [ENOENT] error:

[ENOENT] or [ENOTDIR]

The *path2* argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters. If *path2* without the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.

*Rationale*: Austin Group Defect Report(s) applied: 822. See http://austingroupbugs.net/view.php?id=822.

**Change Number: XSH/TC2/D4/0362** [817]

On Page: 2058 Line: 65105 Section: symlink()
(2013 edition Page: 2074 Line: 66185)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0363** [591]

On Page: 2059 Line: 65140 Section: symlink()

Add a reference to <fcntl.h> to the SEE ALSO list.

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0364** [752]

On Page: 2063 Line: 65302-65317 Section: sysconf()

In the DESCRIPTION section, move lines 65302-65317 (_SC_PAGE_SIZE to _SC_TZNAME_MAX) to before line 65224 (_SC_ADVISORY_INFO).

*Rationale*: Austin Group Defect Report(s) applied: 752. See http://austingroupbugs.net/view.php?id=752. By moving these lines the table will have a block of variables with no underscore, a block of _POSIX_* variables, a block of _POSIX2_* variables, and a block of _XOPEN_* variables, with each block sorted by the _SC_* names.

**Change Number: XSH/TC2/D4/0365** [627]

On Page: 2072 Line: 65661 Section: system()

In the RATIONALE section, add a new paragraph:

Note also that the above example implementation is not thread-safe. Implementations can provide a thread-safe *system*() function, but doing so involves complications such as how to restore the signal dispositions for SIGINT and SIGQUIT correctly if there are overlapping calls, and how to deal with cancellation. The example above would not restore the signal dispositions and would leak a process ID if cancelled. This

does not matter for a non-thread-safe implementation since cancelling a non-thread-safe function results in undefined behavior (see [xref to 2.9.5.2]). To avoid leaking a process ID, a thread-safe implementation would need to terminate the child process when acting on a cancellation.

*Rationale*: Austin Group Defect Report(s) applied: 627. See http://austingroupbugs.net/view.php?id=627. The system() function is not required to be thread-safe, yet the standard requires a cancellation point to occur when executing it.

**Change Number: XSH/TC2/D4/0366** [551]

On Page: 2097 Line: 66353 Section: tdelete()

In the DESCRIPTION section, after the sentence:

The variable pointed to by *rootp* shall be changed if the deleted node was the root of the tree.

add a new sentence:

If the deleted node was the root of the tree and had no children, the variable pointed to by *rootp* shall be set to a null pointer.

On Page: 2098 Line: 66392-66446 Section: tdelete()

In the EXAMPLES section, replace the example program with:

```
#include <limits.h>
#include <search.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

struct element {      /* Pointers to these are stored in the tree. */
    int     count;
    char    string[];
};

void  *root = NULL;         /* This points to the root. */

int main(void)
{
    char   str[_POSIX2_LINE_MAX+1];
    int    length = 0;
    struct element *elementptr;
    void   *node;
    void   print_node(const void *, VISIT, int);
    int    node_compare(const void *, const void *),
           delete_root(const void *, const void *);

    while (fgets(str, sizeof(str), stdin))  {
        /* Set element. */
        length = strlen(str);
        if (str[length-1] == '\n')
            str[--length] = '\0';
        elementptr = malloc(sizeof(struct element) + length + 1);
        strcpy(elementptr->string, str);
        elementptr->count = 1;
        /* Put element into the tree. */
```

```
5397            node = tsearch((void *)elementptr, &root, node_compare);
5398            if (node == NULL) {
5399                fprintf(stderr,
5400                        "tsearch: Not enough space available\n");
5401                exit(EXIT_FAILURE);
5402            }
5403            else if (*(struct element **)node != elementptr) {
5404                /* A node containing the element already exists */
5405                (*(struct element **)node)->count++;
5406                free(elementptr);
5407            }
5408        }
5409        twalk(root, print_node);
5410
5411        /* Delete all nodes in the tree */
5412        while (root != NULL) {
5413            elementptr = *(struct element **)root;
5414            printf("deleting node: string = %s,  count = %d\n",
5415                   elementptr->string,
5416                   elementptr->count);
5417            tdelete((void *)elementptr, &root, delete_root);
5418            free(elementptr);
5419        }
5420
5421        return 0;
5422    }
5423
5424    /*
5425     *  This routine compares two nodes, based on an
5426     *  alphabetical ordering of the string field.
5427     */
5428    int
5429    node_compare(const void *node1, const void *node2)
5430    {
5431        return strcmp(((const struct element *) node1)->string,
5432            ((const struct element *) node2)->string);
5433    }
5434
5435    /*
5436     *  This comparison routine can be used with tdelete()
5437     *  when explicitly deleting a root node, as no comparison
5438     *  is necessary.
5439     */
5440    int
5441    delete_root(const void *node1, const void *node2)
5442    {
5443        return 0;
5444    }
5445
5446    /*
5447     *  This routine prints out a node, the second time
5448     *  twalk encounters it or if it is a leaf.
5449     */
5450    void
5451    print_node(const void *ptr, VISIT order, int level)
5452    {
5453        const struct element *p = *(const struct element **) ptr;
5454
5455        if (order == postorder || order == leaf)  {
5456            (void) printf("string = %s,  count = %d\n",
5457                p->string, p->count);
5458        }
```

}

On Page: 2099 Line: 66450 Section: tdelete()

In the APPLICATION USAGE section, change from:

The *tsearch*() function uses

to:

The *twalk*() function uses

*Rationale*: Austin Group Defect Report(s) applied: 551. See http://austingroupbugs.net/view.php?id=551.

**Change Number: XSH/TC2/D4/0367** [604]

On Page: 2105 Line: 66587-66604 Section: tgamma()

In the DESCRIPTION section, at line 66587 change from:

These functions shall compute the *gamma*() function of *x*.

to:

These functions shall compute $\Gamma(x)$ where $\Gamma(x)$ is defined as $\int_0^\infty e^{-t}t^{x-1}dt$.

In the RETURN VALUE section, at line 66593 change from:

shall return *Gamma*(*x*).

to:

shall return the gamma of *x*.

Before line 66604 insert a new paragraph, shaded MXX:

If *x* is subnormal and 1/*x* is representable, 1/*x* should be returned.

*Rationale*: Austin Group Defect Report(s) applied: 604. See http://austingroupbugs.net/view.php?id=604.

**Change Number: XSH/TC2/D4/0368** [630]

On Page: 2106 Line: 66639 Section: tgamma()

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 **double**, overflow happens when 0 < *x* < 1/DBL_MAX, and 171.7 < *x*.

*Rationale*: Austin Group Defect Report(s) applied: 630. See http://austingroupbugs.net/view.php?id=630.

5484 **Change Number: XSH/TC2/D4/0369** [659]

5485 On Page: 2113 Line: 66876 Section: timer_delete()

5486 In the DESCRIPTION section, add a new paragraph to the end of the section:

5487 The behavior is undefined if the value specified by the *timerid* argument does not correspond to a timer ID
5488 returned by *timer_create*() but not yet deleted by *timer_delete*().

5489 On Page: 2113 Line: 66881-66882 Section: timer_delete()

5490 In the ERRORS section, replace the entire section with:

5491 No errors are defined.

5492 On Page: 2113 Line: 66888 Section: timer_delete()

5493 In the RATIONALE section, change from:

5494 None.

5495 to:

5496 If an implementation detects that the value specified by the *timerid* argument to *timer_delete*() does not
5497 correspond to a timer ID returned by *timer_create*() but not yet deleted by *timer_delete*(), it is
5498 recommended that the function should fail and report an [EINVAL] error.

5499 *Rationale*: Austin Group Defect Report(s) applied: 659. See http://austingroupbugs.net/view.php?id=659.

5500 **Change Number: XSH/TC2/D4/0370** [659]

5501 On Page: 2115 Line: 66956 Section: timer_getoverrun()

5502 In the DESCRIPTION section, add a new paragraph to the end of the section:

5503 The behavior is undefined if the value specified by the *timerid* argument to *timer_getoverrun*(),
5504 *timer_gettime*(), or *timer_settime*() does not correspond to a timer ID returned by *timer_create*() but not yet
5505 deleted by *timer_delete*().

5506 On Page: 2115 Line: 66968-66970 Section: timer_getoverrun()

5507 In the ERRORS section, delete:

5508 These functions may fail if:

5509 [EINVAL] The *timerid* argument does not correspond to an ID returned by *timer_create*() but not yet
5510 deleted by *timer_delete*().

5511 On Page: 2116 Line: 66997 Section: timer_getoverrun()

5512 In the RATIONALE section, add a new paragraph to the end of the section:

If an implementation detects that the value specified by the *timerid* argument to *timer_getoverrun*(), *timer_gettime*(), or *timer_settime*() does not correspond to a timer ID returned by *timer_create*() but not yet deleted by *timer_delete*(), it is recommended that the function should fail and report an [EINVAL] error.

*Rationale*: Austin Group Defect Report(s) applied: 659. See http://austingroupbugs.net/view.php?id=659.

**Change Number: XSH/TC2/D4/0371** [644]

On Page: 2117 Line: 67056 Section: times()

In the ERRORS section, change from:

No errors are defined.

to:

The *times*() function shall fail if:

[EOVERFLOW] The return value would overflow the range of **clock_t**.

*Rationale*: Austin Group Defect Report(s) applied: 644. See http://austingroupbugs.net/view.php?id=644.

**Change Number: XSH/TC2/D4/0372** [678]

On Page: 2121 Line: 67134 Section: tmpfile()

In the DESCRIPTION section, change from:

The *tmpfile*() function shall create a temporary file and open a corresponding stream. The file shall be automatically deleted when all references to the file are closed. The file is opened as in *fopen*() for update (*w+*), except that implementations may restrict the permissions, either by clearing the file mode bits or setting them to the value S_IRUSR | S_IWUSR.

to:

The *tmpfile*() function shall create a temporary file and open a corresponding stream. The file shall be automatically deleted when all references to the file are closed. The file shall be opened as in *fopen*() for update (*wb+*), except that implementations may restrict the permissions, either by clearing the file mode bits or setting them to the value S_IRUSR | S_IWUSR.

*Rationale*: Austin Group Defect Report(s) applied: 678. See http://austingroupbugs.net/view.php?id=678.

**Change Number: XSH/TC2/D4/0373** [685]

On Page: 2131 Line: 67438 Section: towlower()

In the DESCRIPTION section, change from:

a wide-character code corresponding to a valid character in the current locale

to:

a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0374** [685]

On Page: 2133 Line: 67487 Section: towupper()

In the DESCRIPTION section, change from:

a wide-character code corresponding to a valid character in the current locale

to:

a wide-character code corresponding to a valid character in the locale used by the function

*Rationale*: Austin Group Defect Report(s) applied: 685. See http://austingroupbugs.net/view.php?id=685.

**Change Number: XSH/TC2/D4/0375** [489]

On Page: 2136 Line: 67570 Section: truncate()

In the DESCRIPTION, change from:

Upon successful completion, if the file size is changed, *truncate*() shall mark for update the last data modification and last file status change timestamps of the file, and the S_ISUID and S_ISGID bits of the file mode may be cleared.

to:

Upon successful completion, *truncate*() shall mark for update the last data modification and last file status change timestamps of the file, and the S_ISUID and S_ISGID bits of the file mode may be cleared.

*Rationale*: Austin Group Defect Report(s) applied: 489. See http://austingroupbugs.net/view.php?id=489.

**Change Number: XSH/TC2/D4/0376** [656]

On Page: 2140 Line: 67654 Section: ttyname()

In the DESCRIPTION section, change from:

The return value 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 *ttyname*(). The returned pointer and the string content might also be invalidated if the calling thread is terminated.

*Rationale*: Austin Group Defect Report(s) applied: 656. See http://austingroupbugs.net/view.php?id=656. This item is a layered change on XSH/TC1/D5/0065 [75,428].

The change is to add the following text to the end of the paragraph at 2013 edition P2159, L68808-68810:

The returned pointer and the string content might also be invalidated if the calling thread is terminated.

**Change Number: XSH/TC2/D4/0377** [880]

On Page: 2143 Line: 67732 Section: tzset()
(2013 edition Page: 2162 Line: 68888)

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

If a thread accesses *tzname*, [XSI]*daylight*, or *timezone*[/XSI] directly while another thread is in a call to *tzset*(), or to any function that is required or allowed to set timezone information as if by calling *tzset*(), the behavior is undefined.

On Page: 2143 Line: 67747 Section: tzset()
(2013 edition Page: 2162 Line: 68903)

In the APPLICATION USAGE section, change from:

None.

to:

Since the *ctime*(), *localtime*(), *mktime*(), *strftime*() and *strftime_l*() functions are required to set timezone information as if by calling *tzset*(), there is no need for an explicit *tzset*() call before using these functions. However, portable applications should call *tzset*() explicitly before using *ctime_r*() or *localtime_r*() because setting timezone information is optional for those functions.

*Rationale*: Austin Group Defect Report(s) applied: 880. See http://austingroupbugs.net/view.php?id=880.

**Change Number: XSH/TC2/D4/0378** [873]

On Page: 2154 Line: 68001 Section: unlink()
(2013 edition Page: 2174 Line: 69164)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0379** [591]

On Page: 2154 Line: 68005 Section: unlink()

Before the unlinkat SYNOPSIS line, insert a line with OH shading:

```
#include <fcntl.h>
```

*Rationale*: Austin Group Defect Report(s) applied: 591. See http://austingroupbugs.net/view.php?id=591.

**Change Number: XSH/TC2/D4/0380** [817]

On Page: 2154 Line: 68023 Section: unlink()
(2013 edition Page: 2174 Line: 69186)

In the DESCRIPTION section, change from:

If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the file descriptor was opened with O_SEARCH, the function shall not perform the check.

to:

If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the function shall check whether directory searches are permitted using the current permissions of the directory underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0381** [817]

On Page: 2155 Line: 68063 Section: unlink()
(2013 edition Page: 2175 Line: 69227)

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

*fd* was not opened with O_SEARCH and ...

to:

The access mode of the open file description associated with *fd* is not O_SEARCH and ...

*Rationale*: Austin Group Defect Report(s) applied: 817. See http://austingroupbugs.net/view.php?id=817.

**Change Number: XSH/TC2/D4/0382** [582]

On Page: 2162 Line: 68286 Section: uselocale()

Replace the DESCRIPTION, RETURN VALUE, and ERRORS sections with:

**DESCRIPTION**

The *uselocale*() function shall set or query the current locale for the calling thread.

The value for the *newloc* argument shall be one of the following:

1. A value returned by the *newlocale*() or *duplocale*() functions
2. The special locale object descriptor LC_GLOBAL_LOCALE
3. (**locale_t**)0

If the *newloc* argument is (**locale_t**)0, the current locale shall not be changed; this value can

be used to query the current locale setting. If the *newloc* argument is LC_GLOBAL_LOCALE, any thread- local locale for the calling thread shall be uninstalled; the thread shall again use the global locale as the current locale, and changes to the global locale shall affect the thread. Otherwise, the locale represented by *newloc* shall be installed as a thread-local locale to be used as the current locale for the calling thread.

Once the *uselocale*() function has been called to install a thread-local locale, the behavior of every interface using data from the current locale shall be affected for the calling thread. The current locale for other threads shall remain unchanged.

**RETURN VALUE**

Upon successful completion, the *uselocale*() function shall return a handle for the thread-local locale that was in use as the current locale for the calling thread on entry to the function, or LC_GLOBAL_LOCALE if no thread-local locale was in use. Otherwise, *uselocale*() shall return (**locale_t**)0 and set *errno* to indicate the error.

**ERRORS**

The *uselocale*() function may fail if:

[EINVAL]
*newloc* is not a valid locale object and is not (**locale_t**)0.

*Rationale*: Austin Group Defect Report(s) applied: 582. See http://austingroupbugs.net/view.php?id=582.

**Change Number: XSH/TC2/D4/0383** [873]

On Page: 2166 Line: 68405 Section: utimensat()
(2013 edition Page: 2187 Line: 69579)

In the NAME section, delete:

relative to directory file descriptor

*Rationale*: Austin Group Defect Report(s) applied: 873. See http://austingroupbugs.net/view.php?id=873.

**Change Number: XSH/TC2/D4/0384** [690]

On Page: 2181 Line: 68666 Section: wait()
(2013 edition Page: 2203 Line: 69847)

In the DESCRIPTION section change from:

The *wait*() and *waitpid*() functions shall obtain status information pertaining to one of the caller's child processes. Various options permit status information to be obtained for child processes that have terminated or stopped. If status information is available for two or more child processes, the order in which their status is reported is unspecified.

The *wait*() function shall suspend execution of the calling thread until status information for one of the terminated child processes of the calling process is available, or until delivery of a signal whose action is either to execute a signal-catching function or to terminate the process. If more than one thread is

suspended in *wait*() or *waitpid*() awaiting termination of the same process, exactly one thread shall return the process status at the time of the target process termination. If status information is available prior to the call to *wait*(), return shall be immediate.

to:

The *wait*() and *waitpid*() functions shall obtain status information (see XSH Section 2.13) pertaining to one of the caller's child processes. The *wait*() function obtains status information for process termination from any child process. The *waitpid*() function obtains status information for process termination, and optionally process stop and/or continue, from a specified subset of the child processes.

The *wait*() function shall cause the calling thread to become blocked until status information generated by child process termination is made available to the thread, or until delivery of a signal whose action is either to execute a signal-catching function or to terminate the process, or an error occurs. If termination status information is available prior to the call to *wait*(), return shall be immediate. If termination status information is available for two or more child processes, the order in which their status is reported is unspecified.

As described in XSH Section 2.13, the *wait*() and *waitpid*() functions consume the status information they obtain.

The behavior when multiple threads are blocked in *wait*(), *waitid*(), or *waitpid*() is described in XSH Section 2.13.

On Page: 2181 Line: 68700-68703 Section: wait()
(2013 edition Page: 2203 Line: 69881-69884)

In the DESCRIPTION section, delete:

[XSI]If the calling process has SA_NOCLDWAIT set or has SIGCHLD set to SIG_IGN, and the process has no unwaited-for children that were transformed into zombie processes, the calling thread shall block until all of the children of the process containing the calling thread terminate, and *wait*() and *waitpid*() shall fail and set *errno* to [ECHILD].[/XSI]

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0385** [691]

On Page: 2183 Line: 68748-68751,68755-68758 Section: wait()
(2013 Edition Page: 2205 Line: 69929-69932,69936-69939)

In the DESCRIPTION section, change the XSI shading on the first and third paragraphs on the page to completely enclose those paragraphs.

Remove the reference to wait() in the third paragraph, changing it to:

If the information pointed to by *stat_loc* was stored by a call to *waitpid*() that did not specify the WUNTRACED flag and specified the WCONTINUED flag, exactly one of the macros WIFEXITED(**stat_loc*), WIFSIGNALED(**stat_loc*), and WIFCONTINUED(**stat_loc*) shall evaluate to a non-zero value.

*Rationale*: Austin Group Defect Report(s) applied: 691. See http://austingroupbugs.net/view.php?id=691.

**Change Number: XSH/TC2/D4/0386** [690]

On Page: 2186 Line: 68926 Section: wait()

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

[XSI]As specified under "Consequences of Process Termination" in the description of the *_Exit*() function on page 549, if the calling process has SA_NOCLDWAIT set or has SIGCHLD set to SIG_IGN, then the termination of a child process will not cause status information to become available to a thread blocked in *wait*(), *waitid*(), or *waitpid*(). Thus, a thread blocked in one of the wait functions will remain blocked unless some other condition causes the thread to resume execution (such as an [ECHILD] failure due to no remaining children in the set of waited-for children).[/XSI]

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0387** [690]

On Page: 2190 Line: 69058 Section: waitid()
(2013 edition Page: 2212 Line: 70244)

In the DESCRIPTION section, change from:

The *waitid*() function shall suspend the calling thread until one child of the process containing the calling thread changes state. It records the current state of a child in the structure pointed to by infop. The fields of the structure pointed to by infop are filled in as described for the SIGCHLD signal in <signal.h>. If a child process changed state prior to the call to *waitid*(), *waitid*() shall return immediately. If more than one thread is suspended in *wait*(), *waitid*(), or *waitpid*() waiting for termination of the same process, exactly one thread shall return the process status at the time of the target process termination.

to:

The *waitid*() function shall obtain status information (see XSH Section 2.13) pertaining to termination, stop, and/or continue events in one of the caller's child processes.

The *waitid*() function shall cause the calling thread to become blocked until an error occurs or status information becomes available to the calling thread that satisfies all of the following properties ("matching status information"):

- The status information is from one of the child processes in the set of child processes specified by the idtype and id arguments.
- The state change in the status information matches one of the state change flags set in the options argument.

If matching status information is available prior to the call to *waitid*(), return shall be immediate. If matching status information is available for two or more child processes, the order in which their status is reported is unspecified.

As described in XSH Section 2.13, the *waitid*() function consumes the status information it obtains unless the WNOWAIT flag is set in the options argument.

The behavior when multiple threads are blocked in *wait*(), *waitid*(), or *waitpid*() is described in XSH Section 2.13.

The *waitid*() function shall record the obtained status information in the structure pointed to by infop. The

fields of the structure pointed to by infop shall be filled in as described under "Pointer to a Function" in Section 2.4.3 on page 492.

On Page: 2191 Line: 69101 Section: waitid()

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

[XSI]As specified under "Consequences of Process Termination" in the description of the *_Exit*() function on page 549, if the calling process has SA_NOCLDWAIT set or has SIGCHLD set to SIG_IGN, then the termination of a child process will not cause status information to become available to a thread blocked in *wait*(), *waitid*(), or *waitpid*(). Thus, a thread blocked in one of the wait functions will remain blocked unless some other condition causes the thread to resume execution (such as an [ECHILD] failure due to no remaining children in the set of waited-for children).[/XSI]

*Rationale*: Austin Group Defect Report(s) applied: 690. See http://austingroupbugs.net/view.php?id=690.

**Change Number: XSH/TC2/D4/0388** [73]

On Page: 2207 Line: 69521 Section: wcsftime()

In the DESCRIPTION section, change from:

- The argument format is a wide-character string and the conversion specifications are replaced by corresponding sequences of wide characters.

to:

- The argument format is a wide-character string and the conversion specifications are replaced by corresponding sequences of wide characters. It is unspecified whether an encoding error occurs if the format string contains **wchar_t** values that do not correspond to members of the character set of the current locale.

*Rationale*: Austin Group Defect Report(s) applied: 73. See http://austingroupbugs.net/view.php?id=73.

A clarification has been made in the C11 standard.

**Change Number: XSH/TC2/D4/0389** [740]

On Page: 2207 Line: 69521 Section: wcsftime()
(2013 edition Page: 2229 Line: 70726)

In the DESCRIPTION section, add a new bullet item before the final bullet item:

[CX]Field widths specify the number of wide characters instead of the number of bytes.[/CX]

*Rationale*: Austin Group Defect Report(s) applied: 740. See http://austingroupbugs.net/view.php?id=740.

**Change Number: XSH/TC2/D4/0390** [560]

On Page: 2209 Line: 69562-69569 Section: wcslen()

In the DESCRIPTION section, on lines 69562 and 69565 change from:

string

to:

array

On line 69563 change from:

the terminating null wide-character code

to:

any terminating null wide-character code.

In the RETURN VALUE section, on line 69568 change from:

The *wcsnlen*() function shall return an integer containing the smaller of either the length of the wide-character string pointed to by *ws* or *maxlen*.

to:

The *wcsnlen*() function shall return the number of wide characters preceding the first null wide character code in the array to which *ws* points, if *ws* contains a null wide character code within the first *maxlen* wide characters; otherwise, it shall return *maxlen*.

*Rationale*: Austin Group Defect Report(s) applied: 560. See http://austingroupbugs.net/view.php?id=560.

**Change Number: XSH/TC2/D4/0391** [584]

On Page: 2224 Line: 69979 Section: wcstod()

In the DESCRIPTION section, change from:

minus-sign, the sequence shall be interpreted as negated.

to:

<hyphen-minus>, the sequence shall be interpreted as negated.

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0392** [796]

On Page: 2224 Line: 69991 Section: wcstod()
(2013 edition Page: 2246 Line: 71199)

Change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0393** [584]

On Page: 2230 Line: 70196 Section: wcstol()

In the DESCRIPTION section, change from:

minus-sign, the sequence shall be interpreted as negated.

to:

<hyphen-minus>, the sequence shall be interpreted as negated.

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0394** [796]

On Page: 2231 Line: 70199 Section: wcstol()
(2013 edition Page: 2253 Line: 71409)

Change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0395** [584]

On Page: 2237 Line: 70357 Section: wcstoul()

In the DESCRIPTION section, change from:

minus-sign, the sequence shall be interpreted as negated.

to:

<hyphen-minus>, the sequence shall be interpreted as negated.

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XSH/TC2/D4/0396** [796]

On Page: 2238 Line: 70360 Section: wcstoul()
(2013 edition Page: 2260 Line: 71571)

Change from:

In other than the C [CX]or POSIX[/CX] locale[CX]s[/CX], other implementation-defined subject sequences may be

to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

*Rationale*: Austin Group Defect Report(s) applied: 796. See http://austingroupbugs.net/view.php?id=796.

**Change Number: XSH/TC2/D4/0397** [608]

On Page: 2258 Line: 71006 Section: wordexp()

In the DESCRIPTION section, change from:

*wordexp*() may write messages to *stderr* if syntax errors are detected while expanding words.

to:

*wordexp*() may write messages to *stderr* if syntax errors are detected while expanding words, unless the *stderr* stream has wide orientation in which case the behavior is undefined.

*Rationale*: Austin Group Defect Report(s) applied: 608. See http://austingroupbugs.net/view.php?id=608.

**Change Number: XSH/TC2/D4/0398** [704]

On Page: 2258 Line: 71009 Section: wordexp()
(2013 edition Page: 2280 Line: 72230)

In the DESCRIPTION section, add a new paragraph:

The results are unspecified if WRDE_APPEND and WRDE_REUSE are both specified.

*Rationale*: Austin Group Defect Report(s) applied: 704. See http://austingroupbugs.net/view.php?id=704. The standard is clear that wordfree() can be called after any wordexp() call, not just successful ones (the text in the fifth paragraph of the DESCRIPTION does not contain the word "successful"). However, the last sentence of the first paragraph of the RETURN VALUES section states: "In other cases, they shall not be modified.", but it is not clear whether "other cases" here refers to all cases other than when wordexp() returns WRDE_NOSPACE (including success), or just to error cases other than WRDE_NOSPACE. And, neither of these possibilities matches the intended behavior.

**Change Number: XSH/TC2/D4/0399** [704]

On Page: 2259 Line: 71022 Section: wordexp()
(2013 edition Page: 2281 Line: 72243)

In the RETURN VALUE section, change from:

In other cases, they shall not be modified.

to:

In other error cases, if the WRDE_APPEND flag was specified, *pwordexp->we_wordc* and *pwordexp->we_wordv* shall not be modified.

*Rationale*: Austin Group Defect Report(s) applied: 704. See http://austingroupbugs.net/view.php?id=704.

**Change Number: XSH/TC2/D4/0400** [608]

On Page: 2259 Line: 71038 Section: wordexp()

In the APPLICATION USAGE section, add a new paragraph:

Applications which use wide character output functions with *stderr* should ensure that any calls to *wordexp*() do not write to *stderr*, by avoiding use of the WRDE_SHOWERR flag.

*Rationale*: Austin Group Defect Report(s) applied: 608. See http://austingroupbugs.net/view.php?id=608.

**Change Number: XSH/TC2/D4/0401** [676,710]

On Page: 2266 Line: 71250 Section: write()
(2013 edition Page: 2288 Line: 72480)

In the ERRORS section, before "The *write*() function shall fail if" insert:

The *pwrite*() function shall fail if:

[EINVAL]
The file is a regular file or block special file, and the offset argument is negative. The file offset shall remain unchanged.

[ESPIPE]
The file is incapable of seeking.

On Page: 2266 Line: 71275 Section: write()

In the ERRORS section, delete lines 71275-71277 (the pwrite() "shall fail" errors).

*Rationale*: Austin Group Defect Report(s) applied: 676,710. See
http://austingroupbugs.net/view.php?id=676.
See http://austingroupbugs.net/view.php?id=710.
This is a layered change on XSH/TC1/D5/0743 [215].

**Change Number: XSH/TC2/D4/0402** [966]

On Page: 2269 Line: 71387 Section: write()
(2013 edition Page: 2291 Line: 72619)

In the RATIONALE section, change from:

This volume of POSIX.1-2008 does not specify behavior of concurrent writes to a file from multiple processes.

to:

This volume of POSIX.1-2008 does not specify the behavior of concurrent writes to a regular file from multiple threads, except that each write is atomic (see [xref to 2.9.7]).

*Rationale*: Austin Group Defect Report(s) applied: 966. See http://austingroupbugs.net/view.php?id=966.

# 4. 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/TC2/D4/0001** [666]

On Page: 2285 Line: 71853 Section: 1.2 Utility Limits

In the Name column of Table 1-3, change from:

{_POSIX2_RE_DUP_MAX}

to:

{_POSIX_RE_DUP_MAX}

In the Description column of Table 1-3, change from:

The maximum number of repeated occurrences of a BRE permitted when using the interval notation \{*m*,*n*\}; see XBD Section 9.3.6 (on page 186).

to:

Maximum number of repeated occurrences of a BRE or ERE interval expression; see XBD Section 9.3.6 (on page 186) and XBD Section 9.4.6 (on page 189).

On Page: 2287 Line: 71911 Section: 1.2 Utility Limits

In the Description column of Table 1-4, change from:

The maximum number of repeated occurrences of a BRE permitted when using the interval notation \{*m*,*n*\}; see XBD Section 9.3.6 (on page 186).

to:

Maximum number of repeated occurrences of a BRE or ERE interval expression; see XBD Section 9.3.6 (on page 186) and XBD Section 9.4.6 (on page 189).

In the Minimum Value column of Table 1-4, change from:

{_POSIX2_RE_DUP_MAX}

to:

{_POSIX_RE_DUP_MAX}

*Rationale*: Austin Group Defect Report(s) applied: 666. See http://austingroupbugs.net/view.php?id=666.

**Change Number: XCU/TC2/D4/0002** [584]

On Page: 2289 Line: 71989 Section: 1.4 Utility Description Defaults

Change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0003** [913]

On Page: 2293 Line: 72201 Section: 1.4 Utility Description Defaults
(2013 edition Page: 2315 Line: 73439)

Change from:

Utilities may terminate prematurely if they encounter: invalid usage of options, arguments, or environment variables; invalid usage of the complex syntaxes expressed in EXTENDED DESCRIPTION sections; difficulties accessing, creating, reading, or writing files; or difficulties associated with the privileges of the process.

to:

Utilities may terminate prematurely if they encounter: invalid usage of options, arguments, or environment variables; invalid usage of the complex syntaxes expressed in EXTENDED DESCRIPTION sections; resource exhaustion; difficulties accessing, creating, reading, or writing files; or difficulties associated with the privileges of the process.

*Rationale*: Austin Group Defect Report(s) applied: 913. See http://austingroupbugs.net/view.php?id=913.

**Change Number: XCU/TC2/D4/0004** [705]

On Page: 2296 Line: 72310 Section: 1.6 Built-In Utilities
(2013 edition Page: 2318 Line: 73457)

In Table 1-5 Regular Built-In Utilities, add *hash*, *type* and *ulimit* to the table.

*Rationale*: Austin Group Defect Report(s) applied: 705. See http://austingroupbugs.net/view.php?id=705.

**Change Number: XCU/TC2/D4/0005** [718]

On Page: 2299 Line: 72403 Section: 2.3 Token Recognition

Change from:

The shell shall read its input in terms of lines from a file, from a terminal in the case of an interactive shell,

or from a string in the case of *sh* **-c** or *system*().

to:

The shell shall read its input in terms of lines. (For details about how the shell reads its input, see the description of the *sh* utility on page xxx.)

*Rationale*: Austin Group Defect Report(s) applied: 718. See http://austingroupbugs.net/view.php?id=718.

**Change Number: XCU/TC2/D4/0006** [647]

On Page: 2299 Line: 72417 Section: 2.3 Token Recognition

In rule 1, change from:

1. If the end of input is recognized, the current token shall be delimited. If there is no current token, the end-of-input indicator shall be returned as the token.

to:

1. If the end of input is recognized, the current token (if any) shall be delimited.

*Rationale*: Austin Group Defect Report(s) applied: 647. See http://austingroupbugs.net/view.php?id=647.

**Change Number: XCU/TC2/D4/0007** [568]

On Page: 2299 Line: 72419-72424 Section: 2.3 Token Recognition

In rule 2, on line 72420 change from:

current characters

to:

previous characters

In rule 3, on line 72423 change from:

current characters

to:

previous characters

*Rationale*: Austin Group Defect Report(s) applied: 568. See http://austingroupbugs.net/view.php?id=568.

**Change Number: XCU/TC2/D4/0008** [648]

On Page: 2300 Line: 72448 Section: 2.3 Token Recognition

Delete rule 7:

7. If the current character is an unquoted <newline>, the current token shall be delimited.

*Rationale*: Austin Group Defect Report(s) applied: 648. See http://austingroupbugs.net/view.php?id=648.

**Change Number: XCU/TC2/D4/0009** [888]

On Page: 2302 Line: 72511-72523 Section: 2.5.2 Special Parameters
(2013 edition Page: 2324 Line: 73742-73754)

Replace the descriptions of the '@' and '*' special parameters with:

@

Expands to the positional parameters, starting from one, initially producing one field for each positional parameter that is set. When the expansion occurs in a context where field splitting will be performed, any empty fields may be discarded and each of the non-empty fields shall be further split as described in [xref to 2.6.5 Field Splitting]. When the expansion occurs within double-quotes, the behavior is unspecified unless one of the following is true:

- Field splitting as described in [xref to 2.6.5 Field Splitting] would be performed if the expansion were not within double-quotes (regardless of whether field splitting would have any effect, for example if IFS is null).

- The double-quotes are within the *word* of a ${*parameter*:-*word*} or a ${*parameter*:+*word*} expansion (with or without the <colon>; see [xref to 2.6.2 Parameter Expansion]) which would have been subject to field splitting if *parameter* had been expanded instead of *word*.

If one of these conditions is true, the initial fields shall be retained as separate fields except that, if the parameter being expanded was embedded within a word, the first field shall be joined with the beginning part of the original word and the last field shall be joined with the end part of the original word. In all other contexts the results of the expansion are unspecified. If there are no positional parameters, the expansion of '@' shall generate zero fields, even when '@' is within double-quotes; however, if the expansion is embedded within a word which contains one or more other parts that expand to a quoted null string, these null string(s) shall still produce an empty field, except that if the other parts are all within the same double-quotes as the '@', it is unspecified whether the result is zero fields or one empty field.

*

Expands to the positional parameters, starting from one, initially producing one field for each positional parameter that is set. When the expansion occurs in a context where field splitting will be performed, any empty fields may be discarded and each of the non-empty fields shall be further split as described in [xref to 2.6.5 Field Splitting]. When the expansion occurs in a context where field splitting will not be performed, the initial fields shall be joined to form a single field with the value of each parameter separated by the first character of the IFS variable if IFS contains at least one character, or separated by a <space> if IFS is unset, or with no separation if IFS is set to a null string.

*Rationale*: Austin Group Defect Report(s) applied: 888. See http://austingroupbugs.net/view.php?id=888.

**Change Number: XCU/TC2/D4/0010** [888]

On Page: 2303 Line: 72562 Section: 2.5.3 Shell Variables
(2013 edition Page: 2325 Line: 73791)

In the description of IFS, change from:

A string treated as a list of characters that is used for field splitting and to split lines into fields with the *read* command.

If *IFS* is not set, it shall behave as normal for an unset variable, except that field splitting by the shell and line splitting by the *read* command shall ...

to:

A string treated as a list of characters that is used for field splitting, expansion of the '*' special parameter, and to split lines into fields with the *read* utility. If the value of *IFS* includes any bytes that do not form part of a valid character, the results of field splitting, expansion of '*', and use of the *read* utility are unspecified.

If *IFS* is not set, it shall behave as normal for an unset variable, except that field splitting by the shell and line splitting by the *read* utility shall ...

*Rationale*: Austin Group Defect Report(s) applied: 888. See http://austingroupbugs.net/view.php?id=888.

**Change Number: XCU/TC2/D4/0011** [884]

On Page: 2303 Line: 72568 Section: 2.5.3 Shell Variables
(2013 edition Page: 2325 Line: 73796)

In the description of IFS, change from:

Implementations may ignore the value of *IFS* in the environment, or the absence of *IFS* from the environment, at the time the shell is invoked, in which case the shell shall set *IFS* to <space><tab><newline> when it is invoked.

to:

The shell shall set *IFS* to <space><tab><newline> when it is invoked.

*Rationale*: Austin Group Defect Report(s) applied: 884. See http://austingroupbugs.net/view.php?id=884. Some old shells did inherit IFS from the environment, but since most shell scripts do not set IFS as one of the steps in their initialization, this creates a security hole. Most, if not all, recent shells initialize IFS when the shell is invoked and do not change IFS in a subshell environment. This is the desired behavior.

**Change Number: XCU/TC2/D4/0012** [494]

On Page: 2304 Line: 72602 Section: 2.5.3 Shell Variables

Delete the following from the description of the PPID shell variable:

This volume of POSIX.1-2008 specifies the effects of the variable only for systems supporting the User

Portability Utilities option.

*Rationale*: Austin Group Defect Report(s) applied: 494. See http://austingroupbugs.net/view.php?id=494.

**Change Number: XCU/TC2/D4/0013** [888]

On Page: 2306 Line: 72709 Section: 2.6.2 Parameter Expansion
(2013 edition Page: 2328 Line: 73940)

Change from:

Field splitting shall not be performed on the results of the expansion, with the exception of '@'; see [xref to 2.5.2].

to:

Field splitting shall not be performed on the results of the expansion.

*Rationale*: Austin Group Defect Report(s) applied: 888. See http://austingroupbugs.net/view.php?id=888.

**Change Number: XCU/TC2/D4/0014** [867]

On Page: 2306 Line: 72716 Section: 2.6.2 Parameter Expansion
(2013 edition Page: 2328 Line: 73947)

Delete the sentence:

(For example, ${**foo-bar**}**xyz**} would result in the expansion of **foo** followed by the string **xyz}** if **foo** is set, else the string "barxyz}").

On Page: 2308 Line: 72769 Section: 2.6.2 Parameter Expansion
(2013 edition Page: 2330 Line: 74014)

Insert the following before the ${*parameter*:-*word*} example:

${*parameter*-*word*}

This example demonstrates the difference between unset and set to the empty string, as well as the rules for finding the delimiting close brace.

```
foo=asdf
echo ${foo-bar}xyz}
asdfxyz}
foo=
echo ${foo-bar}xyz}
xyz}
unset foo
echo ${foo-bar}xyz}
barxyz}
```

*Rationale*: Austin Group Defect Report(s) applied: 867. See http://austingroupbugs.net/view.php?id=867.

**Change Number: XCU/TC2/D4/0015** [584]

On Page: 2310 Line: 72861 Section: Section 2.6.4 Arithmetic Expansion
(2013 edition Page: 2332 Line: 74115)

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-sign> or <hyphen-minus>, then ...

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.
This item is a layered change on XCU/TC1/D5/0020 [50] applied in TC1, the difference being a change from:
leading plus or minus sign
to:
leading <plus-sign> or <hyphen-minus>

**Change Number: XCU/TC2/D4/0016** [832]

On Page: 2311 Line: 72882 Section: Section 2.6.5 Field Splitting
(2013 edition Page: 2333 Line: 74136)

Change from:

The shell shall treat each character of the *IFS* as a delimiter and use the delimiters as field terminators to split the results of parameter expansion and command substitution into fields.

to:

The shell shall treat each character of the *IFS* as a delimiter and use the delimiters as field terminators to split the results of parameter expansion, command substitution, and arithmetic expansion into fields.

*Rationale*: Austin Group Defect Report(s) applied: 832. See http://austingroupbugs.net/view.php?id=832.
Section 2.6.5 doesn't mention arithmetic expansion (which specifies when field splitting is done), but section 2.6 (where it talks about the order of word expansions) does require field splitting after arithmetic expansions.

**Change Number: XCU/TC2/D4/0017** [890]

On Page: 2313 Line: 72971 Section: 2.7.4 Here-Document
(2013 edition Page: 2335 Line: 74224)

Change from:

The redirection operators "<<" and "<<-" both allow redirection of lines contained in a shell input file, known as a "here-document", to the input of a command.

to:

The redirection operators "<<" and "<<-" both allow redirection of subsequent lines read by the shell to the input of a command. The redirected lines are known as a "here-document".

On Page: 2313 Line: 72981 Section: 2.7.4 Here-Document
(2013 edition Page: 2335 Line: 74234)

After:

where the optional *n* represents the file descriptor number. If the number is omitted, the here-document refers to standard input (file descriptor 0).

add a new sentence:

It is unspecified whether the file descriptor is opened as a regular file, a special file, or a pipe. Portable applications cannot rely on the file descriptor being seekable (see [xref to XSH lseek()]).

*Rationale*: Austin Group Defect Report(s) applied: 890. See http://austingroupbugs.net/view.php?id=890.

**Change Number: XCU/TC2/D4/0018** [583]

On Page: 2313 Line: 72982,72985 Section: 2.7.4 Here-Document

On line 72982 change from:

If any character in *word*

to:

If any part of *word*

On line 72985 change from:

If no characters in *word* are

to:

If no part of *word* is

*Rationale*: Austin Group Defect Report(s) applied: 583. See http://austingroupbugs.net/view.php?id=583.

**Change Number: XCU/TC2/D4/0019** [580]

On Page: 2314 Line: 72990 Section: 2.7.4 Here-Document

Change from:

redirection symbol

to:

redirection operator

*Rationale*: Austin Group Defect Report(s) applied: 580. See http://austingroupbugs.net/view.php?id=580.

**Change Number: XCU/TC2/D4/0020** [882]

On Page: 2315 Line: 73032-73042 Section: 2.8.1 Consequences of Shell Errors
(2013 edition Page: 2337 Line: 74285-74295)

Change from:

For a non-interactive shell, an error condition encountered by a special built-in (see Section 2.14) or other type of utility shall cause the shell to write a diagnostic message to standard error and exit as shown in the following table:

| **Error** | **Special Built-In** | **Other Utilities** |
|---|---|---|
| Shell language syntax error | Shall exit | Shall exit |
| Utility syntax error (option or operand error) | Shall exit | Shall not exit |
| Redirection error | Shall exit | Shall not exit |
| Variable assignment error | Shall exit | Shall not exit |
| Expansion error | Shall exit | Shall exit |
| Command not found | N/A | May exit |
| Dot script not found | Shall exit | N/A |

to:

Certain errors shall cause the shell to write a diagnostic message to standard error and exit as shown in the following table:

| **Error** | **Non-Interactive Shell** | **Interactive Shell** | **Shell Diagnostic Message Required** |
|---|---|---|---|
| Shell language syntax error | shall exit | shall not exit | yes |
| Special built-in utility error | shall exit | shall not exit | no[1] |
| Other utility (not a special built-in) error | shall not exit | shall not exit | no[2] |

| | | | |
|---|---|---|---|
| Redirection error with special built-in utilities | shall exit | shall not exit | yes |
| Redirection error with compound commands | may exit[3] | shall not exit | yes |
| Redirection error with function execution | may exit[3] | shall not exit | yes |
| Redirection error with other utilities (not special built-ins) | shall not exit | shall not exit | yes |
| Variable assignment error | shall exit | shall not exit | yes |
| Expansion error | shall exit | shall not exit | yes |
| Command not found | may exit | shall not exit | yes |

Notes:

1. Although special built-ins are part of the shell, a diagnostic message written by a special built-in is not considered to be a shell diagnostic message, and can be redirected like any other utility.
2. The shell is not required to write a diagnostic message, but the utility itself shall write a diagnostic message if required to do so.
3. A future version of this standard may require the shell to not exit in this condition.

*Rationale*: Austin Group Defect Report(s) applied: 882. See http://austingroupbugs.net/view.php?id=882. The standard is unclear on what constitutes a "Shell Error".

**Change Number: XCU/TC2/D4/0021** [717,882]

On Page: 2315 Line: 73047-73051 Section: 2.8.1 Consequences of Shell Errors
(2013 edition Page: 2337 Line: 74300-74304)

Change from:

If any of the errors shown as "shall exit" or "(may) exit" occur in a subshell, the subshell shall (respectively may) exit with a non-zero status, but the script containing the subshell shall not exit because of the error.

In all of the cases shown in the table, an interactive shell shall write a diagnostic message to standard error without exiting.

to:

If any of the errors shown as "shall exit" or "may exit" occur in a subshell environment, the shell shall (respectively may) exit from the subshell environment with a non-zero status and continue in the environment from which that subshell environment was invoked.

In all of the cases shown in the table where an interactive shell is required not to exit, the shell shall not perform any further processing of the command in which the error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 717,882. See http://austingroupbugs.net/view.php?id=717.
See http://austingroupbugs.net/view.php?id=882.
The changes made here make the standard reflect existing practice.

**Change Number: XCU/TC2/D4/0022** [717]

On Page: 2315 Line: 73059 Section: 2.8.2 Exit Status for Commands

Change from:

If a command fails during word expansion or redirection, its exit status shall be greater than zero.

to:

If a command fails during word expansion or redirection, its exit status shall be between 1 and 125 inclusive.

*Rationale*: Austin Group Defect Report(s) applied: 717. See http://austingroupbugs.net/view.php?id=717.
The changes made here make the standard reflect existing practice.

**Change Number: XCU/TC2/D4/0023** [473]

On Page: 2316 Line: 73070 Section: 2.9 Shell Commands

Change from:

... represent the syntax.

to:

... represent the syntax. In particular the representations include spacing between tokens in some places where blanks would not be necessary (when one of the tokens is an operator).

*Rationale*: Austin Group Defect Report(s) applied: 473. See http://austingroupbugs.net/view.php?id=473.

**Change Number: XCU/TC2/D4/0024** [654]

On Page: 2317 Line: 73101 Section: 2.9.1 Simple Commands

Change from:

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.

to:

Variable assignments shall be performed as follows:

- If no command name results, variable assignments shall affect the current execution environment.
- If the command name is not a special built-in utility or function, 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:
  - — Whether or not the assignments are visible for subsequent expansions in step 4
  - — Whether variable assignments made as side-effects of these expansions are visible for subsequent expansions in step 4, or in the current shell execution environment, or both
- If the command name is a standard utility implemented as a function (see XBD Section 4.21), the effect of variable assignments shall be as if the utility was not implemented as a function.
- If the command name is a special built-in utility, variable assignments shall affect the current execution environment. Unless the *set* **-a** option is on (see the *set* special built-in utility in Section 2.14), it is unspecified:
  - — Whether or not the variables gain the *export* attribute during the execution of the special built-in utility
  - — Whether or not *export* attributes gained as a result of the variable assignments persist after the completion of the special built-in utility
- If the command name is a function that is not a standard utility implemented as a function, variable assignments shall affect the current execution environment during the execution of the function. It is unspecified:
  - — Whether or not the variable assignments persist after the completion of the function
  - — Whether or not the variables gain the *export* attribute during the execution of the function
  - — Whether or not *export* attributes gained as a result of the variable assignments persist after the completion of the function (if variable assignments persist after the completion of the function)

If any of the variable assignments attempt to assign a value to a variable for which the *readonly* attribute is set in the current shell environment (regardless of whether the assignment is made in that environment), a variable assignment error shall occur. See [xref to 2.8.1] for the consequences of these errors.

*Rationale*: Austin Group Defect Report(s) applied: 654. See http://austingroupbugs.net/view.php?id=654. This change is layered on Change Number XCU/TC1/D5/0021 [255] in TC1.

**Change Number: XCU/TC2/D4/0025** [935]

On Page: 2317 Line: 73124 Section: 2.9.1.1 Command Search and Execution
(2013 edition Page: 2339 Line: 74383)

In list item 1. insert a new nested item:

b. If the command name matches the name of a utility listed in the following table, the results are unspecified:

*alloc*
*autoload*
*bind*
*bindkey*
*builtin*
*bye*
*caller*
*cap*
*chdir*
*clone*
*comparguments*
*compcall*
*compctl*
*compdescribe*
*compfiles*
*compgen*
*compgroups*
*complete*
*compquote*
*comptags*
*comptry*
*compvalues*
*declare*
*dirs*
*disable*
*disown*
*dosh*
*echotc*
*echoti*
*help*
*history*
*hist*
*let*
*local*
*login*
*logout*
*map*
*mapfile*
*popd*
*print*
*pushd*
*readarray*
*repeat*
*savehistory*
*source*
*shopt*
*stop*
*suspend*
*typeset*
*whence*

and renumber 1.b, c, and d to 1.c, d, and e.

*Rationale*: Austin Group Defect Report(s) applied: 935. See http://austingroupbugs.net/view.php?id=935.

**Change Number: XCU/TC2/D4/0026** [705]

On Page: 2317 Line: 73130 Section: 2.9.1.1 Command Search and Execution
(2013 edition Page: 2339 Line: 74388)

In item 1.c, change from:

If the command name matches the name of a utility listed in the following table, that utility shall be invoked.

to:

If the command name matches the name [XSI]of the *type* or *ulimit* utility, or[/XSI] of a utility listed in the following table, that utility shall be invoked.

and add *hash* to the table.

*Rationale*: Austin Group Defect Report(s) applied: 705. See http://austingroupbugs.net/view.php?id=705.

**Change Number: XCU/TC2/D4/0027** [521]

On Page: 2318 Line: 73179 Section: 2.9.2 Pipelines

After:

**[!]** *command1* **[** | *command2* **...]**

add a new paragraph:

If the pipeline begins with the reserved word ! and *command1* is a subshell command, the application shall ensure that the ( operator at the beginning of *command1* is separated from the ! by one or more <blank> characters. The behavior of the reserved word ! immediately followed by the ( operator is unspecified.

*Rationale*: Austin Group Defect Report(s) applied: 521. See http://austingroupbugs.net/view.php?id=521. Requiring a negated subshell conflicts with shells that provide an extension of negated pathname expansion.

**Change Number: XCU/TC2/D4/0028** [760]

On Page: 2320 Line: 73226 Section: 2.9.3.1 Asynchronous Lists

Change from:

The standard input for an asynchronous list, before any explicit redirections are performed, shall be considered to be assigned to a file that has the same properties as **/dev/null**. If it is an interactive shell, this need not happen.

to:

If job control is disabled (see *set* **-m**), the standard input for an asynchronous list, before any explicit redirections are performed, shall be considered to be assigned to a file that has the same properties as **/dev/null**. This shall not happen if job control is enabled.

*Rationale*: Austin Group Defect Report(s) applied: 760. See http://austingroupbugs.net/view.php?id=760.

**Change Number: XCU/TC2/D4/0029** [473]

On Page: 2321 Line: 73274 Section: 2.9.1.4 Grouping Commands

Change from:

(*compound-list*)

to:

( *compound-list* )

On Page: 2321 Line: 73277 Section: 2.9.4.1 Grouping Commands

Change from:

{ *compound-list*;}

to:

{ *compound-list* ; }

On Page: 2322 Line: 73312 Section: 2.9.4.3 Case Conditional Construct

Change from:

case *word* in
**[(]***pattern1*) *compound-list*;;
**[[(]***pattern***[** | *pattern***]** ... ) *compound-list*;;**]** ...
**[[(]***pattern***[** | *pattern***]** ... ) *compound-list***]**
esac

to:

case *word* in
**[(]** *pattern1* ) *compound-list* ;;
**[[(]** *pattern***[** | *pattern***]** ... ) *compound-list* ;;**]** ...
**[[(]** *pattern***[** | *pattern***]** ... ) *compound-list***]**
esac

On Page: 2324 Line: 73379 Section: 2.9.5 Function Definition Command

Change from:

*fname*() *compound-command***[***io-redirect* ...**]**

to:

*fname* ( ) *compound-command* **[***io-redirect* ...**]**

*Rationale*: Austin Group Defect Report(s) applied: 473. See http://austingroupbugs.net/view.php?id=473.

**Change Number: XCU/TC2/D4/0030** [654]

On Page: 2324 Line: 73391 Section: 2.9.5 Function Definition Command
(2013 edition Page: 2347 Line: 74675-74676)

Change from:

When a function is executed, it shall have the syntax-error and variable-assignment properties described for special built-in utilities in the enumerated list at the beginning of Section 2.14.

to:

When a function is executed, it shall have the syntax-error properties described for special built-in utilities in the first item in the enumerated list at the beginning of Section 2.14.

*Rationale*: Austin Group Defect Report(s) applied: 654. See http://austingroupbugs.net/view.php?id=654.

**Change Number: XCU/TC2/D4/0031** [648]

On Page: 2325 Line: 73415 Section: 2.10.1 Shell Grammar Lexical Conventions

Delete rule 1:

1. A <newline> shall be returned as the token identifier NEWLINE.

*Rationale*: Austin Group Defect Report(s) applied: 648. See http://austingroupbugs.net/view.php?id=648.

**Change Number: XCU/TC2/D4/0032** [574,646]

On Page: 2325 Line: 73421 Section: 2.10.1 Shell Grammar Lexical Conventions

Change from:

**ASSIGNMENT**

to:

**ASSIGNMENT_WORD**

*Rationale*: Austin Group Defect Report(s) applied: 574,646. See
http://austingroupbugs.net/view.php?id=574.
See http://austingroupbugs.net/view.php?id=646.

**Change Number: XCU/TC2/D4/0033** [643,839]

On Page: 2326 Line: 73475-73481 Section: 2.10.2 Shell Grammar Rules
(2013 edition Page: 2349 Line: 74755-74761)

Change rule 7.b from:

If the **TOKEN** contains the <equals-sign> character:

- If it begins with '=', the token **WORD** shall be returned.
- If all the characters preceding '=' form a valid name (see XBD Section 3.230, on page 70), the token **ASSIGNMENT_WORD** shall be returned. (Quoted characters cannot participate in forming a valid name.)
- Otherwise, it is unspecified whether it is **ASSIGNMENT_WORD** or **WORD** that is returned.

to:

If the **TOKEN** contains an unquoted (as determined while applying rule #4 from section 2.3) <equals-sign> character that is not part of an embedded parameter expansion, command substitution, or arithmetic expansion construct (as determined while applying rule #5 from section 2.3):

- If the **TOKEN** begins with '=', then rule 1 shall be applied.
- If all the characters in the **TOKEN** preceding the first such <equals-sign> form a valid name (see XBD Section 3.230, on page 70), the token **ASSIGNMENT_WORD** shall be returned.
- Otherwise, it is unspecified whether rule 1 is applied or **ASSIGNMENT_WORD** is returned.

Otherwise, rule 1 shall be applied.

*Rationale*: Austin Group Defect Report(s) applied: 643,839. See http://austingroupbugs.net/view.php?id=643.
See http://austingroupbugs.net/view.php?id=839.

**Change Number: XCU/TC2/D4/0034** [643]

On Page: 2326 Line: 73482 Section: 2.10.2 Shell Grammar Rules

In rule 7, change from:

Assignment to the **NAME** shall occur ...

to:

Assignment to the name within a returned **ASSIGNMENT_WORD** token shall occur ...

*Rationale*: Austin Group Defect Report(s) applied: 643. See http://austingroupbugs.net/view.php?id=643.

**Change Number: XCU/TC2/D4/0035** [648]

On Page: 2327 Line: 73500 Section: 2.10.2 Shell Grammar Rules

In the shell grammar, change from:

```
/* The following are the operators mentioned above */
```

to:

```
/* The following are the operators (see XBD 3.255) containing more
   than one character */
```

*Rationale*: Austin Group Defect Report(s) applied: 648. See http://austingroupbugs.net/view.php?id=648.

**Change Number: XCU/TC2/D4/0036** [736]

On Page: 2327 Line: 73521 Section: 2.10.2 Shell Grammar Rules
(2013 edition Page: 2350 Line: 74801-74808)

In the shell grammar, change from:

```
%start  complete_command
%%
complete_command : list separator
                 | list
                 ;
```

to:

```
%start program
%%
program          : linebreak complete_commands linebreak
                 | linebreak
                 ;
complete_commands: complete_commands newline_list complete_command
                 |                                complete_command
                 ;
complete_command : list separator_op
                 | list
                 ;
```

*Rationale*: Austin Group Defect Report(s) applied: 736. See http://austingroupbugs.net/view.php?id=736.

**Change Number: XCU/TC2/D4/0037** [737]

On Page: 2328 Line: 73554 Section: 2.10.2 Shell Grammar Rules
(2013 edition Page: 2350 Line: 74834-74838)

In the shell grammar, change from:

```
compound_list    :              term
                 | newline_list term
                 |              term separator
```

```
                    | newline_list term separator
                    ;
```

to:

```
compound_list       : linebreak term
                    | linebreak term separator
                    ;
```

*Rationale*: Austin Group Defect Report(s) applied: 737. See http://austingroupbugs.net/view.php?id=737.

**Change Number: XCU/TC2/D4/0038** [581]

On Page: 2328 Line: 73562-73564 Section: 2.10.2 Shell Grammar Rules

In the shell grammar, change from:

```
for_clause      : For name linebreak                                do_group
                | For name linebreak in sequential_sep            do_group
                | For name linebreak in wordlist sequential_sep do_group
                ;
```

to:

```
for_clause      : For name                                        do_group
                | For name sequential_sep                         do_group
                | For name linebreak in sequential_sep            do_group
                | For name linebreak in wordlist sequential_sep do_group
                ;
```

*Rationale*: Austin Group Defect Report(s) applied: 581. See http://austingroupbugs.net/view.php?id=581.

**Change Number: XCU/TC2/D4/0039** [735]

On Page: 2328 Line: 73583 Section: 2.10.2 Shell Grammar Rules
(2013 edition Page: 2351 Line: 74863-74866)

In the shell grammar, change from:

```
case_item_ns    :     pattern ')'               linebreak
                |     pattern ')' compound_list linebreak
                | '(' pattern ')'               linebreak
                | '(' pattern ')' compound_list linebreak
                ;
```

to:

```
case_item_ns    :     pattern ')' linebreak
                |     pattern ')' compound_list
                | '(' pattern ')' linebreak
                | '(' pattern ')' compound_list
                ;
```

*Rationale*: Austin Group Defect Report(s) applied: 735. See http://austingroupbugs.net/view.php?id=735.

**Change Number: XCU/TC2/D4/0040** [750]

On Page: 2330 Line: 73676 Section: 2.11 Signals and Error Handling
(2013 edition Page: 2353 Line: 74957)

Change from:

When a command is in an asynchronous list, it shall inherit from the shell a signal action of ignored (SIG_IGN) for the SIGQUIT and SIGINT signals, and may inherit a signal mask in which SIGQUIT and SIGINT are blocked. Otherwise, the signal actions and signal mask inherited by the command shall be the same as ...

to:

If job control is disabled (see the description of *set* **-m**) when the shell executes an asynchronous list, the commands in the list shall inherit from the shell a signal action of ignored (SIG_IGN) for the SIGINT and SIGQUIT signals. In all other cases, commands executed by the shell shall inherit the same signal actions as ...

*Rationale*: Austin Group Defect Report(s) applied: 750. See http://austingroupbugs.net/view.php?id=750.

**Change Number: XCU/TC2/D4/0041** [706]

On Page: 2331 Line: 73693 Section: 2.12 Shell Execution Environment
(2013 edition Page: 2353 Line: 74974)

In the shell execution environment list, add a new entry with XSI shading:

- File size limit as set by *ulimit*.

*Rationale*: Austin Group Defect Report(s) applied: 706. See http://austingroupbugs.net/view.php?id=706.

**Change Number: XCU/TC2/D4/0042** [806]

On Page: 2332 Line: 73736 Section: 2.13.1 Patterns Matching a Single Character
(2013 edition Page: 2354 Line: 75017)

Change from:

The escaping <backslash> shall be discarded.

to:

The escaping <backslash> shall be discarded. If a pattern ends with an unescaped <backslash>, it is unspecified whether the pattern does not match anything or the pattern is treated as invalid.

*Rationale*: Austin Group Defect Report(s) applied: 806. See http://austingroupbugs.net/view.php?id=806.
Existing practice is for a pattern ending in an unescaped <backslash> not to match anything, but it would be useful to allow implementations to diagnose this as user error.

**Change Number: XCU/TC2/D4/0043** [963]

On Page: 2333 Line: 73803 Section: 2.13.3 Patterns Used for Filename Expansion
(2013 edition Page: 2356 Line: 75082)

Change from:

... sorted according to the collating sequence in effect in the current locale.

to:

... sorted according to the collating sequence in effect in the current locale. If this collating sequence does not have a total ordering of all characters (see [xref to XBD 7.3.2]), any filenames or pathnames that collate equally should be further compared byte-by-byte using the collating sequence for the POSIX locale.

<small>Note: a future version of this standard may require the byte-by-byte further comparison described above.</small>

*Rationale*: Austin Group Defect Report(s) applied: 963. See http://austingroupbugs.net/view.php?id=963.

**Change Number: XCU/TC2/D4/0044** [882]

On Page: 2334 Line: 73821 Section: 2.14 Special Built-In Utilities
(2013 edition Page: 2356 Line: 75100)

In list item 1, change from:

A syntax error in a special built-in utility may cause a shell executing that utility to abort, while a syntax error in a regular built-in utility shall not cause a shell executing that utility to abort. (See Section 2.8.1 for the consequences of errors on interactive and non-interactive shells.) If a special built-in utility encountering a syntax error does not abort the shell, its exit value shall be non-zero.

to:

An error in a special built-in utility may cause a shell executing that utility to abort, while an error in a regular built-in utility shall not cause a shell executing that utility to abort. (See Section 2.8.1 for the consequences of errors on interactive and non-interactive shells.) If a special built-in utility encountering an error does not abort the shell, its exit value shall be non-zero.

*Rationale*: Austin Group Defect Report(s) applied: 882. See http://austingroupbugs.net/view.php?id=882.

**Change Number: XCU/TC2/D4/0045** [654]

On Page: 2334 Line: 73826 Section: 2.14 Special Built-In Utilities
(2013 edition Page: 2356 Line: 75105)

In list item 2, change from:

Variable assignments specified with special built-in utilities remain in effect after the built-in completes; this shall not be the case with a regular built-in or other utility.

to:

As described in Section 2.9.1, variable assignments preceding the invocation of a special built-in utility remain in effect after the built-in completes; this shall not be the case with a regular built-in or other utility.

*Rationale*: Austin Group Defect Report(s) applied: 654. See http://austingroupbugs.net/view.php?id=654.

**Change Number: XCU/TC2/D4/0046** [842]

On Page: 2335 Line: 73839 Section: 2.14 break
(2013 edition Page: 2358 Line: 75117)

Change the DESCRIPTION section from:

The *break* utility shall exit from the smallest enclosing **for**, **while**, or **until** loop, if any; or from the *n*th enclosing loop if *n* is specified. The value of *n* is an unsigned decimal integer greater than or equal to 1. The default shall be equivalent to *n*=1. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be exited. Execution shall continue with the command immediately following the loop.

to:

If *n* is specified, the *break* utility shall exit from the *n*th enclosing **for**, **while**, or **until** loop. If *n* is not specified, *break* shall behave as if *n* was specified as 1. Execution shall continue with the command immediately following the exited loop. The value of *n* is a positive decimal integer. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be exited. If there is no enclosing loop, the behavior is unspecified.

A loop shall enclose a *break* or *continue* command if the loop lexically encloses the command. A loop lexically encloses a *break* or *continue* command if the command is:

- executing in the same execution environment (see section 2.12) as the compound-list of the loop's do-group (see section 2.10.2), and
- contained in a compound-list associated with the loop (either in the compound-list of the loop's do-group or, if the loop is a **while** or **until** loop, in the compound-list following the **while** or **until** reserved word), and
- not in the body of a function whose function definition command (see section 2.9.5) is contained in a compound-list associated with the loop.

If *n* is greater than the number of lexically enclosing loops and there is a non-lexically enclosing loop in progress in the same execution environment as the *break* or *continue* command, it is unspecified whether that loop encloses the command.

On Page: 2336 Line: 73877 Section: 2.14 break
(2013 edition Page: 2359 Line: 75155)

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

The results of running the following example are unspecified: There are two loops in progress when the *break* command is executed, and they are in the same execution environment, but neither loop is lexically enclosing the *break* command. (There are no loops lexically enclosing the *continue* commands, either.)

```
foo() {
```

```
6605        for j in 1 2; do
6606            echo 'break 2' >/tmp/do_break
6607            echo "  sourcing /tmp/do_break ($j)..."
6608            # the behavior of the break from running the following command
6609            # results in unspecified behavior:
6610            . /tmp/do_break
6611
6612            do_continue() { continue 2; }
6613            echo "  running do_continue ($j)..."
6614            # the behavior of the continue in the following function call
6615            # results in unspecified behavior (if execution reaches this
6616            # point):
6617            do_continue
6618
6619            trap 'continue 2' USR1
6620            echo "  sending SIGUSR1 to self ($j)..."
6621            # the behavior of the continue in the trap invoked from the
6622            # following signal results in unspecified behavior (if
6623            # execution reaches this point):
6624            kill -USR1 $$
6625            sleep 1
6626        done
6627    }
6628    for i in 1 2; do
6629        echo "running foo ($i)..."
6630        foo
6631    done
```

6632 On Page: 2339 Line: 73966 Section: 2.14 continue
6633 (2013 edition Page: 2362 Line: 75244)

6634 Change the DESCRIPTION section from:

6635 The *continue* utility shall return to the top of the smallest enclosing **for**, **while**, or **until** loop, or to the top
6636 of the *n*th enclosing loop, if *n* is specified. This involves repeating the condition list of a **while** or **until** loop
6637 or performing the next assignment of a **for** loop, and re-executing the loop if appropriate.

6638 The value of *n* is a decimal integer greater than or equal to 1. The default shall be equivalent to *n*=1. If *n* is
6639 greater than the number of enclosing loops, the outermost enclosing loop shall be used.

6640 to:

6641 If *n* is specified, the *continue* utility shall return to the top of the *n*th enclosing **for**, **while**, or **until** loop. If *n*
6642 is not specified, *continue* shall behave as if *n* was specified as 1. Returning to the top of the loop involves
6643 repeating the condition list of a **while** or **until** loop or performing the next assignment of a **for** loop, and re-
6644 executing the loop if appropriate. The value of *n* is a positive decimal integer. If *n* is greater than the
6645 number of enclosing loops, the outermost enclosing loop shall be used. If there is no enclosing loop, the
6646 behavior is unspecified.

6647 The meaning of "enclosing" shall be as specified in the description of the *break* utility.

6648 *Rationale*: Austin Group Defect Report(s) applied: 842. See http://austingroupbugs.net/view.php?id=842.

6649

**Change Number: XCU/TC2/D4/0047** [717]

On Page: 2347 Line: 74214 Section: 2.14 exit
(2013 edition Page: 2370 Line: 75508)

In the DESCRIPTION section, change from:

The *exit* utility shall cause the shell to exit with the exit status specified by the unsigned decimal integer *n*.

to:

The *exit* utility shall cause the shell to exit from its current execution environment with the exit status specified by the unsigned decimal integer *n*. If the current execution environment is a subshell environment, the shell shall exit from the subshell environment with the specified exit status and continue in the environment from which that subshell environment was invoked, otherwise the shell utility shall terminate with the specified exit status.

*Rationale*: Austin Group Defect Report(s) applied: 717. See http://austingroupbugs.net/view.php?id=717. The changes made here make the standard reflect existing practice.

**Change Number: XCU/TC2/D4/0048** [960]

On Page: 2347 Line: 74240 Section: 2.14 exit
(2013 edition Page: 2370 Line: 75534)

In the EXIT STATUS section, change from:

The exit status shall be *n*, if specified.

to:

The exit status shall be *n*, if specified, except that the behavior is unspecified if *n* is not an unsigned decimal integer or is greater than 255.

*Rationale*: Austin Group Defect Report(s) applied: 960. See http://austingroupbugs.net/view.php?id=960.

**Change Number: XCU/TC2/D4/0049** [717]

On Page: 2348 Line: 74252 Section: 2.14 exit
(2013 edition Page: 2371 Line: 75546)

In the EXAMPLES section, add at the end of the section:

Propagate error handling from within a subshell:

```
    (
        command1 || exit 1
        command2 || exit 1
        exec command3
    ) > outputfile || exit 1
    echo "outputfile created successfully"
```

*Rationale*: Austin Group Defect Report(s) applied: 717. See http://austingroupbugs.net/view.php?id=717.

**Change Number: XCU/TC2/D4/0050** [960]

On Page: 2348 Line: 74258 Section: 2.14 exit
(2013 edition Page: 2371 Line: 75552)

In the RATIONALE section, add a new paragraph:

The behavior of *exit* when given an invalid argument or unknown option is unspecified, because of differing practices in the various historical implementations. A value larger than 255 might be truncated by the shell, and be unavailable even to a parent process that uses *waitid*() to get the full exit value. It is recommended that implementations that detect any usage error should cause a non-zero exit status (or, if the shell is interactive and the error does not cause the shell to abort, store a non-zero value in $?), but even this was not done historically in all shells.

*Rationale*: Austin Group Defect Report(s) applied: 960. See http://austingroupbugs.net/view.php?id=960.

**Change Number: XCU/TC2/D4/0051** [654]

On Page: 2349 Line: 74290 Section: 2.14 export
(2013 edition Page: 2372 Line: 75584)

No change required to POSIX.1-2008.

Revert Change Number XCU/TC1/D5/0043 from TC1, removing the text added:

If a variable assignment precedes the command name of *export* but that variable is not also listed as an operand of *export*, then that variable shall be set in the current shell execution environment after the completion of the *export* command, but it is unspecified whether that variable is marked for export.

*Rationale*: Austin Group Defect Report(s) applied: 654. See http://austingroupbugs.net/view.php?id=654. This change is layered on top of Change Number: XCU/TC1/D5/0043 from TC1. Earlier changes in section 2.9.1 now mean that the change in TC1 is not required.

**Change Number: XCU/TC2/D4/0052** [960]

On Page: 2350 Line: 74312 Section: 2.14 export
(2013 edition Page: 2373 Line: 75609)

In the EXIT STATUS section, change from:

Zero.

to:

0 All *name* operands were successfully exported.

>0 At least one *name* could not be exported, or the **-p** option was specified and an error occurred.

On Page: 2350 Line: 74316 Section: 2.14 export
(2013 edition Page: 2373 Line: 75613)

In the APPLICATION USAGE section, change from:

None.

to:

Note that, unless X was previously marked readonly, the value of $? after

```
export X=$(false)
```

will be 0 (because *export* successfully set X to the empty string) and that execution continues, even if *set* **-e** is in effect. In order to detect command substitution failures, a user must separate the assignment from the export, as in

```
X=$(false)
export X
```

On Page: 2353 Line: 74396 Section: 2.14 readonly
(2013 edition Page: 2376 Line: 75695)

In the EXIT STATUS section, change from:

Zero.

to:

0 All *name* operands were successfully marked readonly.

>0 At least one *name* could not be marked readonly, or the **-p** option was specified and an error occurred.

On Page: 2355 Line: 74460 Section: 2.14 return
(2013 edition Page: 2378 Line: 75759)

In the EXIT STATUS section, change from:

If *n* is greater than 255, the results are undefined.

to:

If *n* is not an unsigned decimal integer, or is greater than 255, the results are unspecified.

*Rationale*: Austin Group Defect Report(s) applied: 960. See http://austingroupbugs.net/view.php?id=960. The standard was unclear whether certain shell special builtins must have exit status of 0, or whether the overall rule for non-zero status after error detection applied. Existing shells have non-zero status after errors.

**Change Number: XCU/TC2/D4/0053** [584]

On Page: 2357 Line: 74508, 74510 Section: 2.14 set

In the DESCRIPTION section, change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0054** [717]

On Page: 2358 Line: 74545 Section: 2.14 set
(2013 edition Page: 2381 Line: 75845)

In the DESCRIPTION section, replace the description of **-e** with:

When this option is on, when any command fails (for any of the reasons listed in Section 2.8.1 (on page xxx) or by returning an exit status greater than zero) the shell immediately shall exit, as if by executing the *exit* special built-in utility with no arguments, with the following exceptions:

1. The failure of any individual command in a multi-command pipeline shall not cause the shell to exit. Only the failure of the pipeline itself shall be considered.

2. The **-e** setting shall be ignored when executing the compound list following the **while**, **until**, **if**, or **elif** reserved word, a pipeline beginning with the ! reserved word, or any command of an AND-OR list other than the last.

3. If the exit status of a compound command other than a subshell command was the result of a failure while **-e** was being ignored, then **-e** shall not apply to this command.

This requirement applies to the shell environment and each subshell environment separately. For example, in:

```
set -e; (false; echo one) | cat; echo two
```

the *false* command causes the subshell to exit without executing `echo one`; however, `echo two` is executed because the exit status of the pipeline `(false; echo one) | cat` is zero.

*Rationale*: Austin Group Defect Report(s) applied: 717. See http://austingroupbugs.net/view.php?id=717.

This is layered change on XCU/TC1/D5/0046 [52]. The change over the 2013 edition is to change from:
the shell immediately shall exit with the following exceptions
to:
the shell immediately shall exit, as if by executing the exit special built-in utility with no arguments, with the following exceptions

**Change Number: XCU/TC2/D4/0055** [717]

On Page: 2359 Line: 74590 Section: 2.14 set

In the DESCRIPTION section, change the description of **-u** from:

The shell shall write a message to standard error when it tries to expand a variable that is not set and immediately exit. An interactive shell shall not exit.

to:

When the shell tries to expand an unset parameter other than the '@' and '*' special parameters, it shall write a message to standard error and the expansion shall fail with the consequences specified in 2.8.1 Consequences of Shell Errors (on page xxx).

*Rationale*: Austin Group Defect Report(s) applied: 717. See http://austingroupbugs.net/view.php?id=717. This is a layered change on XCU/TC1/D5/0047 [155,280]. This change replaces the previous revised text for the -u option.

**Change Number: XCU/TC2/D4/0056** [960]

On Page: 2360 Line: 74627 Section: 2.14 set
(2013 edition Page: 2383 Line: 75944)

In the EXIT STATUS section, change from:

Zero.

to:

0 Successful completion.

>0 An invalid option was specified, or an error occurred.

On Page: 2366 Line: 74829 Section: 2.14 times
(2013 edition Page: 2389 Line: 76171)

In the EXIT STATUS section, change from:

Zero.

to:

0 Successful completion.

>0 An error occurred.

*Rationale*: Austin Group Defect Report(s) applied: 960. See http://austingroupbugs.net/view.php?id=960. The standard was unclear whether certain shell special builtins must have exit status of 0, or whether the overall rule for non-zero status after error detection applied. Existing shells have non-zero status after

errors.

**Change Number: XCU/TC2/D4/0057** [584]

On Page: 2415 Line: 76517, 76518 Section: ar

In the RATIONALE section, change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0058** [584]

On Page: 2434 Line: 77280 Section: awk

In the EXTENDED DESCRIPTION section, change from:

sequence begins with a minus-sign,

to:

sequence begins with a <hyphen-minus>,

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0059** [963]

On Page: 2437 Line: 77382 Section: awk
(2013 edition Page: 2459 Line: 78745)

In the EXTENDED DESCRIPTION section, change from:

operands shall be converted to strings as required and a string comparison shall be made using the locale-specific collation sequence. The value of the comparison expression shall be 1 if the relation is true, or 0 if the relation is false.

to:

operands shall be converted to strings as required and a string comparison shall be made as follows:

- For the "!=" and "==" operators, the strings should be compared to check if they are identical but may be compared using the locale-specific collation sequence to check if they collate equally.
- For the other operators, the strings shall be compared using the locale-specific collation sequence.

The value of the comparison expression shall be 1 if the relation is true, or 0 if the relation is false.

*Rationale*: Austin Group Defect Report(s) applied: 963. See http://austingroupbugs.net/view.php?id=963.

**Change Number: XCU/TC2/D4/0060** [226]

On Page: 2447 Line: 77855 Section: awk
(2013 edition Page: 2470 Line: 79219)

In the EXTENDED DESCRIPTION section, apply the following *diff* to the grammar:

```
 program          : item_list
-                 | actionless_item_list
+                 | item_list item
                  ;


-item_list        : newline_opt
-                 | actionless_item_list item terminator
-                 | item_list            item terminator
-                 | item_list          action terminator
+item_list        : /* empty */
+                 | item_list item terminator
                  ;

-actionless_item_list : item_list            pattern terminator
-                 | actionless_item_list pattern terminator
-                 ;
-
-
-item             : pattern action
+item             : action
+                 | pattern action
+                 | normal_pattern
                  | Function NAME      '(' param_list_opt ')'
                       newline_opt action
                  | Function FUNC_NAME '(' param_list_opt ')'
```

On Page: 2448 Line: 77878 Section: awk
(2013 edition Page: 2471 Line: 79242)

In the EXTENDED DESCRIPTION section, apply the following *diff* to the grammar:

```
-pattern          : Begin
-                 | End
-                 | expr
+pattern          : normal_pattern
+                 | special_pattern
+                 ;
+
+normal_pattern   : expr
                  | expr ',' newline_opt expr
                  ;


+special_pattern  : Begin
+                 | End
+                 ;
+
+
```

```
 action           : '{' newline_opt                                '}'
                  | '{' newline_opt terminated_statement_list   '}'
                  | '{' newline_opt unterminated_statement_list '}'
                  ;


-terminator       : terminator ';'
-                 | terminator NEWLINE
+terminator       : terminator NEWLINE
                  |            ';'
                  |            NEWLINE
                  ;
```

*Rationale*: Austin Group Defect Report(s) applied: 226. See http://austingroupbugs.net/view.php?id=226. The standard does not require support for awk '/foo/; {print}', but this was unintentional.

awk BEGIN (with no action) and awk END (with no action) are allowed by the grammar but forbidden by the EXTENDED DESCRIPTION section on Special Patterns (see Issue 7, P2440, L77545 and in Issue 7, 2013 Edition, P2463, L78909).

The standard requires support for awk '{print};;{print}', but this is not historic practice and is considered to be a poor programming practice.

**Change Number: XCU/TC2/D4/0061** [663]

On Page: 2455 Line: 78223 Section: awk
(2013 edition Page: 2478 Line: 79587)

In the APPLICATION USAGE section, add a new paragraph:

When using *awk* to process pathnames, it is recommended that LC_ALL, or at least LC_CTYPE and LC_COLLATE, are set to POSIX or C in the environment, since pathnames can contain byte sequences that do not form valid characters in some locales, in which case the utility's behavior would be undefined. In the POSIX locale each byte is a valid single-byte character, and therefore this problem is avoided.

*Rationale*: Austin Group Defect Report(s) applied: 663. See http://austingroupbugs.net/view.php?id=663.

**Change Number: XCU/TC2/D4/0062** [963]

On Page: 2455 Line: 78223 Section: awk
(2013 edition Page: 2478 Line: 79587)

In the APPLICATION USAGE section, add two new paragraphs:

On implementations where the "==" operator checks if strings collate equally, applications needing to check whether strings are identical can use:

```
length(a) == length(b) && index(a,b) == 1
```

On implementations where the "==" operator checks if strings are identical, applications needing to check whether strings collate equally can use:

```
a <= b && a >= b
```

*Rationale*: Austin Group Defect Report(s) applied: 963. See http://austingroupbugs.net/view.php?id=963.

**Change Number: XCU/TC2/D4/0063** [226]

On Page: 2458 Line: 78313 Section: awk
(2013 edition Page: 2481 Line: 79677)

In the RATIONALE section, add a new paragraph:

Earlier versions of this standard required implementations to support multiple adjacent <semicolon>s, lines with one or more <semicolon>s before a rule ("pattern {action}" pairs), and lines with only <semicolon>(s). These are not required by this standard and are considered poor programming practice, but can be accepted by an implementation of awk as an extension.

*Rationale*: Austin Group Defect Report(s) applied: 226. See http://austingroupbugs.net/view.php?id=226.

**Change Number: XCU/TC2/D4/0064** [963]

On Page: 2463 Line: 78550 Section: awk
(2013 edition Page: 2486 Line: 79914)

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of this standard may require the "!=" and "==" operators to perform string comparisons by checking if the strings are identical (and not by checking if they collate equally).

*Rationale*: Austin Group Defect Report(s) applied: 963. See http://austingroupbugs.net/view.php?id=963.

**Change Number: XCU/TC2/D4/0065** [612]

On Page: 2465 Line: 78660 Section: basename

Add a new paragraph at the end of the EXAMPLES section :

The EXAMPLES section of the *basename*() function (see XREF to XSH basename() EXAMPLES section) includes a table showing examples of the results of processing several sample pathnames by the *basename*() and *dirname*() functions and by the *basename* and *dirname* utilities.

On Page: 2466 Line: 78676 Section: basename

In the SEE ALSO section, add a new paragraph at the end:

XSH *basename*(),

*Rationale*: Austin Group Defect Report(s) applied: 612. See http://austingroupbugs.net/view.php?id=612.

**Change Number: XCU/TC2/D4/0066** [584]

On Page: 2476 Line: 79064 Section: bc

In the EXTENDED DESCRIPTION section, change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0067** [679]

On Page: 2480 Line: 79259 Section: bc
(2013 edition Page: 2503 Line: 80630)

In the EXTENDED DESCRIPTION section, change from:

**j**( *expression*, *expression* )
Bessel function of integer order.

to:

**j**( *expression1*, *expression2* )
Bessel function of *expression2* of the first kind of integer order *expression1*.

*Rationale*: Austin Group Defect Report(s) applied: 679. See http://austingroupbugs.net/view.php?id=679.

**Change Number: XCU/TC2/D4/0068** [650]

On Page: 2489 Line: 79568 Section: c99

In the OPTIONS section, in the -E description change from:

expanding all preprocessor directives

to:

executing all preprocessor directives

*Rationale*: Austin Group Defect Report(s) applied: 650. See http://austingroupbugs.net/view.php?id=650.

**Change Number: XCU/TC2/D4/0069** [670]

On Page: 2491 Line: 79683,79696 Section: c99

In the EXTENDED DESCRIPTION section, at line 79683 (**-l c**), change from:

beginning with the prefix clock_ or time_

to:

beginning with the prefix clock_ or timer_

At line 79696 (**-l rt**), change from:

interfaces beginning with the prefix clock_ and time_

to:

interfaces beginning with the prefix clock_ and timer_

*Rationale*: Austin Group Defect Report(s) applied: 670. See http://austingroupbugs.net/view.php?id=670.

**Change Number: XCU/TC2/D4/0070** [638]

On Page: 2494 Line: 79799 Section: c99

In the CONSEQUENCES OF ERRORS section, change from:

it shall not perform the link phase and return a non-zero exit status

to:

it shall not perform the link phase and it shall return a non-zero exit status

*Rationale*: Austin Group Defect Report(s) applied: 638. See http://austingroupbugs.net/view.php?id=638.

**Change Number: XCU/TC2/D4/0071** [650]

On Page: 2497 Line: 79928 Section: c99

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

When the **-E** option is used, execution of some **#pragma** preprocessor directives may simply result in a copy of the directive being included in the output as part of the allowed extra information used by subsequent compilation passes (see STDOUT).

*Rationale*: Austin Group Defect Report(s) applied: 650. See http://austingroupbugs.net/view.php?id=650.

**Change Number: XCU/TC2/D4/0072** [784]

On Page: 2497 Line: 79930 Section: c99
(2013 edition Page: 2520 Line: 81307)

In the FUTURE DIRECTIONS section, change from:

None.

to:

Unlike all of the other non-OB-shaded utilities in this standard, a utility by this name probably will not appear in the next revision of this standard. This utility's name is tied to the current revision of the C Standard at the time this standard is approved. Since the C Standard and this standard are maintained by different organizations on different schedules, we cannot predict what the compiler will be named in the next revision of the standard.

*Rationale*: Austin Group Defect Report(s) applied: 784. See http://austingroupbugs.net/view.php?id=784.

**Change Number: XCU/TC2/D4/0073** [876]

On Page: 2502 Line: 80083 Section: cat
(2013 edition Page: 2526 Line: 81474)

In the STDOUT section, add a new sentence:

If the standard output is a regular file, and is the same file as any of the input file operands, the implementation may treat this as an error.

On Page: 2502 Line: 80115-80118 Section: cat
(2013 edition Page: 2526 Line: 81506-81509)

In the EXAMPLES section, change from:

Because of the shell language mechanism used to perform output redirection, a command such as this:

```
cat doc doc.end > doc
```

causes the original data in doc to be lost.

to:

Because of the shell language mechanism used to perform output redirection, a command such as this:

```
cat doc doc.end > doc
```

causes the original data in **doc** to be lost before *cat* even begins execution. This is true whether the *cat* command fails with an error or silently succeeds (the specification allows both behaviors). In order to append the contents of **doc.end** without losing the original contents of **doc**, this command should be used instead:

```
cat doc.end >> doc
```

*Rationale*: Austin Group Defect Report(s) applied: 876. See http://austingroupbugs.net/view.php?id=876.
A cat command which redirects its standard output to a file that is also named as a file operand is likely to run until the output file reaches the maximum output file size allowed for that process or the underlying filesystem runs out of space. This is a common application error that accidentally consumes a lot of space needed by other users on the system. Therefore, many implementations of the cat utility check for this condition and, when it is found, print a diagnostic message and exit with a non-zero exit status. This behavior is not currently allowed by the standard, but should be.

**Change Number: XCU/TC2/D4/0074** [584]

On Page: 2507 Line: 80264 Section: cd

In the OPERANDS section, change from:

<hyphen>

to:

<hyphen-minus>

*Rationale*: Austin Group Defect Report(s) applied: 584. See http://austingroupbugs.net/view.php?id=584.

**Change Number: XCU/TC2/D4/0075** [478]

On Page: 2532 Line: 81246 Section: cmp

In the OPTIONS section, change the **-s** description from:

Write nothing for differing files; return exit status only.

to:

Write nothing to standard output or standard error when files differ; indicate differing files through exit status only. It is unspecified whether a diagnostic message is written to standard error when an error is encountered; if a message is not written, the error is indicated through exit status only.

On Page: 2533 Line: 81295 Section: cmp

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

If the **-s** option is used and an error occurs, it is unspecified whether a diagnostic message is written to standard error.

On Page: 2534 Line: 81314 Section: cmp

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

Since the behavior of **-s** differs between implementations as to whether error messages are written, the only way to ensure consistent behavior of cmp when **-s** is used is to redirect standard error to /dev/null.

If error messages are wanted, instead of using **-s** standard output should be redirected to /dev/null, and anything written to standard error should be discarded if the exit status is 1. For example:

```
silent_cmp() {
    # compare files with no output except error messages
    message=$(cmp "$@" 2>&1 >/dev/null)
    status=$?
    case $status in
    (0|1) ;;
    (*) printf '%s\n' "$message" ;;
```