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 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
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 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 Dempsy
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
Jarno Jaakkola
Felix Janda
Aurelio Jargas
Andrew Josey
John Kearney
Michael Kerrisk
Rob King
Andi Kleen
Bruce Korb
Kaz Klyheku
Antoine Leca
Wojtek Lerch
Sven Mascheck
Margot Hackett Miller
Joseph S. Myers
Szabolcs Nagy
Alexander Nasonov
Jonathan Nieder
Danny Niu
Gian Ntzik
Stepfen Nurmepso
Carlos O'Donell
Vladimir Támaro Pátiño
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
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

Copyright © 2016 IEEE and The Open Group. All rights reserved.
# Austin Group Working Group Members

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eitan Adler</td>
<td>Philip Guenther</td>
<td>Jim Pryor</td>
</tr>
<tr>
<td>Iwan Aucamp</td>
<td>Joseph M. Gwinn</td>
<td>James C. Pugsley</td>
</tr>
<tr>
<td>Bogdan Barbu</td>
<td>Bruno Haible</td>
<td>Yury Pukhalsky</td>
</tr>
<tr>
<td>David Bartley</td>
<td>Richard Hansen</td>
<td>Steve Rago</td>
</tr>
<tr>
<td>Steve Bartolomei</td>
<td>Mark Harris</td>
<td>Chet Ramey</td>
</tr>
<tr>
<td>Fabrice Bautzac</td>
<td>Barry E. Hedquist</td>
<td>Martin Rehák</td>
</tr>
<tr>
<td>Guido Berhoerster</td>
<td>David Holland</td>
<td>Tom Ridge</td>
</tr>
<tr>
<td>Eric Blake</td>
<td>Tom Honermann</td>
<td>Torvald Riegel</td>
</tr>
<tr>
<td>Harvey Block</td>
<td>Cyril Hrubis</td>
<td>Xavier Roche</td>
</tr>
<tr>
<td>Hans Boehm</td>
<td>Jarmo Jaakkola</td>
<td>Askar Safin</td>
</tr>
<tr>
<td>Bill Bogstad</td>
<td>Felix Janda</td>
<td>Anton Salikhmetov</td>
</tr>
<tr>
<td>Pádraig Brady</td>
<td>Aurelio Jargas</td>
<td>Jörg Schilling</td>
</tr>
<tr>
<td>Davide Brini</td>
<td>Ross Johnson</td>
<td>Ed Schouten</td>
</tr>
<tr>
<td>Andries E. Brouwer</td>
<td>Andrew Josey</td>
<td>Konrad Schwarz</td>
</tr>
<tr>
<td>Mark S. Brown</td>
<td>John Kearney</td>
<td>Ingo Schwarze</td>
</tr>
<tr>
<td>David Butenhof</td>
<td>Dan Kegei</td>
<td>Jens Schweikhardt</td>
</tr>
<tr>
<td>Milan Cermak</td>
<td>Michael Kerrisk</td>
<td>Mike Scudder</td>
</tr>
<tr>
<td>Stephane Chazelas</td>
<td>Rob King</td>
<td>Martin Sebor</td>
</tr>
<tr>
<td>Alexander Cherepanov</td>
<td>Tomas Klacko</td>
<td>Glen Seeds</td>
</tr>
<tr>
<td>Mark Clancy</td>
<td>Andi Kleen</td>
<td>Jeffrey Sheinberg</td>
</tr>
<tr>
<td>Geoff Clare</td>
<td>Bruce Korb</td>
<td>Thor Lancelot Simon</td>
</tr>
<tr>
<td>David Clissold</td>
<td>David Korn</td>
<td>Keld Simonsen</td>
</tr>
<tr>
<td>Alan Coopersmith</td>
<td>Kaz Kylheku</td>
<td>Ranjit Singh</td>
</tr>
<tr>
<td>Donald W. Cragun</td>
<td>Rob Landley</td>
<td>Paul Smith</td>
</tr>
<tr>
<td>Martijn Dekker</td>
<td>Antoine Leca</td>
<td>Steven Stewart-Gallus</td>
</tr>
<tr>
<td>Matthew Dempsky</td>
<td>Vincent Lefèvre</td>
<td>Nicholas M. Stoughton</td>
</tr>
<tr>
<td>Thomas E. Dickey</td>
<td>Sean Leonard</td>
<td>Alfred M. Szmidt</td>
</tr>
<tr>
<td>Casper Dik</td>
<td>Wojtek Lerch</td>
<td>Marcel Telka</td>
</tr>
<tr>
<td>Dan Douglas</td>
<td>Yonggang Luo</td>
<td>Donn Terry</td>
</tr>
<tr>
<td>Niall Douglas</td>
<td>Scott Lurndal</td>
<td>Keith Thompson</td>
</tr>
<tr>
<td>Ulrich Drepper</td>
<td>Roger Marquis</td>
<td>Jilles Tjøelker</td>
</tr>
<tr>
<td>Lawrence D.K.B. Dwyer</td>
<td>Sven Mascheck</td>
<td>William Toh</td>
</tr>
<tr>
<td>Paul Eggert</td>
<td>Per Mildner</td>
<td>Daniel Trebiën</td>
</tr>
<tr>
<td>Daniel Eischen</td>
<td>Margot Hackett Miller</td>
<td>Miloslav Trmac</td>
</tr>
<tr>
<td>Robert Elz</td>
<td>Joseph S. Myers</td>
<td>Fred J. Tydeman</td>
</tr>
<tr>
<td>Steve Emmerson</td>
<td>Szabolcs Nagy</td>
<td>Ted Unangst</td>
</tr>
<tr>
<td>Laszlo Ersek</td>
<td>Alexander Nasonov</td>
<td>Brian Utterback</td>
</tr>
<tr>
<td>Bruce Evans</td>
<td>Jonathan Nieder</td>
<td>Stijn van Drongelen</td>
</tr>
<tr>
<td>David Egan Evans</td>
<td>Danny Niu</td>
<td>Christopher Vance</td>
</tr>
<tr>
<td>Roger Faulkner</td>
<td>Gian Nizik</td>
<td>Jonathan Wakely</td>
</tr>
<tr>
<td>Richard Felker</td>
<td>Steffen Nurpmeso</td>
<td>Nathan Weeks</td>
</tr>
<tr>
<td>Jeffrey K. Fellin</td>
<td>Carlos O’Donnell</td>
<td>Florian Weimer</td>
</tr>
<tr>
<td>Hal Finkel</td>
<td>Isabella Parakiss</td>
<td>David A. Wheeler</td>
</tr>
<tr>
<td>Glenn Fowler</td>
<td>Vladimir Támara Pátiño</td>
<td>Mats D. Wichmann</td>
</tr>
<tr>
<td>Cathy Fox</td>
<td>Phil Pennock</td>
<td>Michael Wilson</td>
</tr>
<tr>
<td>Mike Frysinger</td>
<td>Andres Perera</td>
<td>Garrett Wollman</td>
</tr>
<tr>
<td>Andrea Giugliano</td>
<td>Peter Petrov</td>
<td>Jörg Wunsch</td>
</tr>
<tr>
<td>Thorsten Glaser</td>
<td>William Pitcock</td>
<td>James Youngman</td>
</tr>
<tr>
<td>Glenn D. Golden</td>
<td>Wayne Pollock</td>
<td>André Zepezauer</td>
</tr>
<tr>
<td>Jim Grisanzio</td>
<td></td>
<td>Mark Ziegast</td>
</tr>
</tbody>
</table>
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 | Donald W. Cragun | Andrew Josey |
| Geoff Clare | Roger Faulkner | Martin Rehák |
| David Clissold | Darrin Johnson | 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 | Randall Groves | Kenneth Lang |
| Juan Carreon | Joseph M. Gwinn | Vincent Lefèvre |
| Keith Chow | Barry E. Hedquist | Peter Petrov |
| Donald W. Cragun | Werner Hoeltz | Stephen Schwarm |
| Sourav Dutta | Noriyuki Ikeuchi | Walter Strupper |
| Andrew Fieldsend | Andrew Josey | Mark-Rene Uchida |
| David Fuschi | Piotr Karocki | Oren Yuen |
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
IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/PR/disclaimers.html.

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

In the line below the Source Documents header, change from:
base documents for POSIX.1-2008
to:
base documents for POSIX.1-2001


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

On Page: xxxviii Line: NA Section: Acknowledgements

Remove the period from the first bullet list item.

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:

```c
[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:

```c
[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.


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()].


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

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

Change from:

```c
_POSIX_TIMERS
```

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

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

On Page: 35 Line: 1195 Section: 3.17 Application

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.


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].


The restrictions on using longjmp() and siglongjmp() are more restrictive than they need to be on POSIX.

Copyright © 2016 IEEE and The Open Group. All rights reserved.
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.


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.


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.


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

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

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

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.

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 a non-standard utility with equivalent flags, and which makes use of interfaces that are only made available by c99 when the -lpthread 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 -lpthread 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](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](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>

On Page: 80 Line: 2244-2250 Section: 3.289 Process

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.


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>


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>

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


218 Change from:
219 ... ed, lex, ...
220
to:
221 ... ed, ex, lex, ...


224 Change from:
225 ... ed, lex, ...
226
to:
227 ... ed, ex, lex, ...


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


231 Add a sentence:
232 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.


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

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

237 Change from:
238 <hyphen>
239
to:
240 <hyphen-minus>

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


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.


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


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]


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


Change from:

<hyphen>

to:

<hyphen-minus>


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


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).


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

On Page: 110 Line: 2980 Section: 4.11 Memory Synchronization


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.

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, \texttt{errno} shall be set to \texttt{[ENAMETOOLONG]} and an error indication shall be returned.


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]

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

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


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.


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>),


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


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.


Change from:

Symbolic Name

to:
### Symbolic Name(s)

On Page: 125 Line: 3460-3466 Section: 6.1 Portable Character Set


Change from:

<table>
<thead>
<tr>
<th>Symbolic Name(s)</th>
<th>Codepoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle alert\rangle</td>
<td>\langle U0007\rangle</td>
<td>BELL (BEL)</td>
</tr>
<tr>
<td>\langle backspace\rangle</td>
<td>\langle U0008\rangle</td>
<td>BACKSPACE (BS)</td>
</tr>
<tr>
<td>\langle tab\rangle</td>
<td>\langle U0009\rangle</td>
<td>CHARACTER TABULATION (HT)</td>
</tr>
<tr>
<td>\langle carriage-return\rangle</td>
<td>\langle U000D\rangle</td>
<td>CARRIAGE RETURN (CR)</td>
</tr>
<tr>
<td>\langle newline\rangle</td>
<td>\langle U000A\rangle</td>
<td>LINE FEED (LF)</td>
</tr>
<tr>
<td>\langle vertical-tab\rangle</td>
<td>\langle U000B\rangle</td>
<td>LINE TABULATION (VT)</td>
</tr>
<tr>
<td>\langle form-feed\rangle</td>
<td>\langle U000C\rangle</td>
<td>FORM FEED (FF)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Symbolic Name(s)</th>
<th>Codepoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle alert\rangle, \langle BEL\rangle</td>
<td>\langle U0007\rangle</td>
<td>BELL</td>
</tr>
<tr>
<td>\langle backspace\rangle, \langle BS\rangle</td>
<td>\langle U0008\rangle</td>
<td>BACKSPACE</td>
</tr>
<tr>
<td>\langle tab\rangle, \langle HT\rangle</td>
<td>\langle U0009\rangle</td>
<td>CHARACTER TABULATION</td>
</tr>
<tr>
<td>\langle newline\rangle, \langle LF\rangle</td>
<td>\langle U000A\rangle</td>
<td>LINE FEED (LF)</td>
</tr>
<tr>
<td>\langle vertical-tab\rangle, \langle VT\rangle</td>
<td>\langle U000B\rangle</td>
<td>LINE TABULATION</td>
</tr>
<tr>
<td>\langle form-feed\rangle, \langle FF\rangle</td>
<td>\langle U000C\rangle</td>
<td>FORM FEED (FF)</td>
</tr>
<tr>
<td>\langle carriage-return\rangle, \langle CR\rangle</td>
<td>\langle U000D\rangle</td>
<td>CARRIAGE RETURN (CR)</td>
</tr>
</tbody>
</table>


Change from:
to (note that the glyph used for \texttt{<hyphen-minus>}, \texttt{<hyphen>} should be the same one that is used in the
construct "\texttt{(-\textunderscore )}" in the rest of the standard):

\begin{tabular}{|c|c|}
\hline
\texttt{<hyphen-minus>}, \texttt{<hyphen>} & \texttt{-} \texttt{<U002D>} HYPHEN-MINUS \\
\hline
\texttt{<full-stop>}, \texttt{<period>} & \texttt{.} \texttt{<U002E>} FULL STOP \\
\hline
\texttt{<slash>}, \texttt{<solidus>} & \texttt{/} \texttt{<U002F>} SOLIDUS \\
\hline
\end{tabular}


Change from:

\begin{tabular}{|c|c|}
\hline
\texttt{<backslash>} & \texttt{\textbackslash} \texttt{<U005C>} REVERSE SOLIDUS \\
\hline
\texttt{<reverse-solidus>} & \texttt{\textbackslash} \texttt{<U005C>} REVERSE SOLIDUS \\
\hline
\end{tabular}


Change from:

\begin{tabular}{|c|c|}
\hline
\texttt{<circumflex-accent>} & \texttt{\textasciicircum} \texttt{<U005E>} CIRCUMFLEX ACCENT \\
\hline
\end{tabular}
<circumflex> ^ <U005E> CIRCUMFLEX ACCENT
<low-line> _ <U005F> LOW LINE
<underscore> _ <U005F> LOW LINE


Change from:
<left-brace> [ <U007B> LEFT CURLY BRACKET
<left-curly-bracket> [ <U007B> LEFT CURLY BRACKET

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


Change from:
<right-brace> } <U007D> RIGHT CURLY BRACKET
<right-curly-bracket> } <U007D> RIGHT CURLY BRACKET

to:

Delete the sentence:

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


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


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.

This is a layered change on XBD/TC1/D5/0019 [291].

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


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.

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]

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.

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
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
Change from:

Table 6-2 Control Character Set

\begin{verbatim}
<ACK> <DC2> <ENQ> <FS> <IS4> <SOH> 
<BEL> <DC3> <EOT> <GS> <LF> <STX>
\end{verbatim}
to:

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

Table 6-2 Non-Portable Control Characters

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


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.


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

On Page: 136 Line: 3849 Section: 7.2 POSIX locale

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.
496 On Page: 139 Line: 3996 Section: 7.3.1 LC_CTYPE
498 Change from:
499 In the POSIX locale, the 26 uppercase letters shall be included:
500 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
501 to:
502 In the POSIX locale, only:
503 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
504 shall be included.
505 On Page: 139 Line: 4003 Section: 7.3.1 LC_CTYPE
507 Change from:
508 In the POSIX locale, the 26 lowercase letters shall be included:
509 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
510 to:
511 In the POSIX locale, only:
512 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
513 shall be included.
514 On Page: 139 Line: 4009 Section: 7.3.1 LC_CTYPE
516 Change from:
517 In the POSIX locale, all characters in the classes upper and lower shall be included.
518 to:
519 In the POSIX locale, only characters in the classes upper and lower shall be included.
520 On Page: 141 Line: 4091 Section: 7.3.1 LC_CTYPE
522 Change from:
523 In the POSIX locale, at a minimum, the 26 lowercase characters:
In the POSIX locale, the 26 lowercase characters:

On Page: 142 Line: 4105 Section: 7.3.1 LC_CTYPE

Change from:

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

to:

In the POSIX locale, the 26 uppercase characters:


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

On Page: 143 Line: 4142 Section: 7.3.1.1 LC_CTYPE Category in the POSIX Locale

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

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"

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>


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>


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

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).
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.

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.

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

UNDEFINED IGNORE;...
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.

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.


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

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.

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>


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

On Page: 163 Line: 5073 Section: 7.3.5.3 LC_TIME Category in the POSIX Locale

In the LC_TIME category definition, change from:
<p percent_sign>
to:
<p percent-sign>


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>


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


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.

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:

that is not preceded by a <backslash>

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)

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).

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

On Page: 184 Line: 5902 Section: 9.3.5 RE Bracket Expression

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.

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

In list item 1, change from:

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

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’.

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>

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

On Page: 184 Line: 5926 Section: 9.3.5 RE Bracket Expression

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>).


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

On Page: 184 Line: 5928 Section: 9.3.5 RE Bracket Expression

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

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


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

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

Change from:

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

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

Change from:

<hyphen>

<hyphen-minus>


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.

<hyphen>
<hyphen-minus>

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".

<hyphen>
<hyphen-minus>

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

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).

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".


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 'S' to the list of characters.

858 Change Number: XBD/TC2/D4/0053 [916]
859 On Page: 195 Line: 6387 Section: 9.5.3 ERE Grammar
860 (2013 edition Page: 195 Line: 6433)
861 Change from:
862 Conforming applications cannot use such constructs.
863 to:
864 Strictly conforming applications cannot use such constructs.
866 Change Number: XBD/TC2/D4/0054 [967]
867 On Page: 198 Line: 6438-6454 Section: 10.2 Output Devices and Terminal Types
869 Delete the two Symbolic Name columns from Table 10-1.
871 Change Number: XBD/TC2/D4/0055 [745]
872 On Page: 204 Line: 6682,6684 Section: 11.1.9 Special Characters
873 On line 6682, for CR change from:
874 carriage-return
875 to:
876 <carriage-return>
877 On line 6684, append a sentence to the description of CR:
878 It cannot be changed.
880 Change Number: XBD/TC2/D4/0056 [584]
882 Change from:
883 <hyphen>
to:

<hyphen-minus>


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


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.


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>`.


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.

On Page: 240 Line: 7878 Section: <fcntl.h>
In the DESCRIPTION section, change from:

O_DIRECTORY Fail if not a directory.

to:

O_DIRECTORY Fail if file is a non-directory file.


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 `{\langle m,n\rangle};` 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 `{\langle m,n\rangle};` 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).


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

On Page: 283 Line: 9438 Section: `<locale.h>`


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>`


Delete:

[CX]Implementations may add additional masks using the form `LC_*_MASK`.\[/CX]\n
On Page: 284 Line: 9464 Section: `<locale.h>`


In the RATIONALE section, change from:

None.

None.
It is suggested that each category macro name for use in setlocale() have a corresponding macro name ending in _MASK for use in newlocale().


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

On Page: 286 Line: 9512 Section: <math.h>

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


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

On Page: 292 Line: 9776 Section: <math.h>

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.


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

On Page: 303 Line: 10154 Section: <netinet/in.h>

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>`

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.


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

On Page: 306 Line: 10239 Section: `<netinet/in.h>`

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:

```c
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:

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

or

```c
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.


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

On Page: 306 Line: 10241 Section: `<netinet/in.h>`


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.


**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.

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


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].


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

On Page: 331 Line: 11073 Section: <signal.h>


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.)


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.


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>`

In the DESCRIPTION section, change from:

```c
int    sigqueue(pid_t, int, const union sigval);
```
to:

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


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

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

In the SEE ALSO section, add:

```
XCU kill
```


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

On Page: 355 Line: 11953 Section: `<stdlib.h>`

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]
```


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>

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_.


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

On Page: 385 Line: 12911 Section: <sys/socket.h>

In the DESCRIPTION section, add a new paragraph:

The value of AF_UNSPEC shall be 0.


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.


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

On Page: 399 Line: 13386 Section: <sys/types.h>

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

`suseconds_t` Used for time in microseconds.

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.

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

On Page: 403 Line: 13525 Section: <sys/un.h>


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.


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>`.


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>`.


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

On Page: 409 Line: 13695-13698 Section: `<tar.h>`


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

<table>
<thead>
<tr>
<th>TMAGIC</th>
<th>&quot;ustar&quot;</th>
<th>Used in the magic field in the ustar header block, including the trailing null byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMAGLEN</td>
<td>6</td>
<td>Length in octets of the magic field</td>
</tr>
<tr>
<td>TVERSION</td>
<td>&quot;00&quot;</td>
<td>Used in the version field in the ustar header block, excluding the trailing null byte</td>
</tr>
<tr>
<td>TVERSLEN</td>
<td>2</td>
<td>Length in octets of the version field</td>
</tr>
</tbody>
</table>


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

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

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


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

On Page: 437 Line: 14773 Section: `<unistd.h>`


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.

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

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

In the DESCRIPTION section, change from:

```c
intptr_t
type as described in `<inttypes.h>`.  
```

to:

```c
intptr_t
type as described in `<stdint.h>`.
```


In the DESCRIPTION section, add a new paragraph:

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


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

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

In the DESCRIPTION section, change from:

```c
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:

```c
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.
```


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

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


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.


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.


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.

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.


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

On Page: 472 Line: 16038 Section: 2.2.2 The Name Space


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
```


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>`.

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, and strtoumax.
- Delete wctof, wcstomax, 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
- sprintf
- snprintf
- va_copy
- vsprintf
- 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
- ceil
- lddiv
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>


**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.


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

On Page: 482 Line: 16442 Section: 2.3 Error Numbers


Change from:

EOPNOTSUP

to:

EOPNOTSUPP


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

On Page: 487 Line: 16657 Section: 2.4.3 Signal Actions


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.

Rationale: Austin Group Defect Report(s) applied: 690.

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


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-capturing functions:

to:

Therefore, applications can call them, without restriction, from signal-capturing 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-capturing function (see [xref to longjmp()]).

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


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

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


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(), stmpcpy(), streclt(), strchr(), strcmp(), strcspn(), strlen(), strncat(), strncmp(), strncpy(),
strlclen(), strpbrk(), strstr(), strspn(), strtok_r(), wmemcpy(), wmemset(), wcschr(), wcscat(), wcschr(),
wcsncmp(), wcsncpy(), wcscpy(), wcscspn(), wcslcat(), wcscmp(), wcscpy(), wcslen(), wcsncat(), wcsncmp(), wcsncpy(), wcsncsi(),
wcsspbrk(), wcsrchr(), wcsstr(), wcsstok(), wmemchr(), wmemcmp(), wmemset()\


Since these functions do not modify or depend on global state, there are no known implementations where
these functions cannot be safely used in async-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.


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().


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.

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

On Page: 490 Line: 16775 Section: 2.4.4 Signal Effects on Other Functions

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 \texttt{errno} is unspecified if the signal-catching function or any function it called assigned a value to \texttt{errno} and the signal-catching function did not save and restore the original value of \texttt{errno}.


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 \texttt{fgetc}(); all output takes place as if bytes were written by successive calls to \texttt{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 \texttt{fgetwc}() function. Each conversion shall occur as if by a call to the \texttt{mbrtowc}() function, with the conversion state described by the stream's own \texttt{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 \texttt{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 \texttt{fputwc}() function. Each conversion shall occur as if by a call to the \texttt{wctomb}() function, with the conversion state described by the stream's own \texttt{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 \texttt{fputc}() function.

The \texttt{perror}(), \texttt{psiginfo}() and \texttt{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 \texttt{perror}(), \texttt{psiginfo}() and \texttt{psignal}() that write to streams but are neither wide character output nor byte output functions (\texttt{getopt()} and \texttt{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.

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

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),

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.


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

On Page: 507 Line: 17498 Section: 2.9.1 Thread-Safety

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


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.

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.


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()
- getprotobynumber()
- 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()
- getervbyname()
- setpwent()
- dbm_store()
- getnetbyaddr()
- getervbyport()
- setutxent()
- endgrent()
- getnetbyname()
- getservent()
- strerror()
- endpwent()
- getnetent()
- getutxent()
- tbyname()

Remove pclose() from the list of optional 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.


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.


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.


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

On Page: 517 Line: 17899 Section: 2.9 Threads

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, 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 initialize 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.


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

On Page: 541 Line: 18857 Section: 2.12.1 Defined Types

In the description of sig_atomic_t, change from:

Integer type ...

to:

Possibly volatile-qualified integer type ...

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()`.
  - 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.


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.


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.


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

On Page: 545 Line: 18925 Section: _Exit()

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]

  [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(), waitid(), or
  waitpid(), 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
... 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].


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

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.


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

In the NAME section, delete:

relative to directory file descriptor


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

Before the faccessat SYNOPSIS line, insert a line with OH shading:

#include <fcntl.h>

On Page: 561 Line: 19449 Section: access()


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()


In the DESCRIPTION section, change from:

The fasccessat() function shall be equivalent to the access() function, except ...

to:

The fasccessat() function when called with a flag value of zero shall be equivalent to the access() function, except ...


Change Number: XSH/TC2/D4/0040 [817]

On Page: 561 Line: 19460 Section: access()


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.

In the DESCRIPTION section, change from:

If `faccessat()` is passed the special value AT_FDCWD in the \texttt{fd} parameter, the current working directory is used and the behavior shall be identical to a call to \texttt{access()}.


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


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

\texttt{fd} was not opened with O\_SEARCH and ...

Change Number: XSH/TC2/D4/0044 [838]

On Page: 562 Line: 19515 Section: access()

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()

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.

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.

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()


In the SYNOPSIS section, change from:

```c
long atol(const char *str);
```

to:

```c
long atol(const char *nptr);
```

On Page: 613 Line: 20937 Section: atol()


In the DESCRIPTION section, change from:

The call `atol(str)` shall be equivalent to:

```c
strtol(str, (char **)NULL, 10)
```
Except as noted below, the call `atol(nptr)` shall be equivalent to:

```
strtol(nptr, (char **)NULL, 10)
```

On Page: 613 Line: 20939 section atol()

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 atoll()

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 atoll()

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.


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 \texttt{basename()}. 

The \texttt{basename()} function may modify the string pointed to by \texttt{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 \texttt{basename()}. The returned pointer might also be invalidated if the calling thread is terminated.

\textbf{Rationale}: Austin Group Defect Report(s) applied: 656. See \url{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.

\textbf{Change Number: XSH/TC2/D4/0048} [928]

\textbf{Change Number: XSH/TC2/D4/0049} [612]

In the EXAMPLES section, change from:

\begin{verbatim}
char *name = "/usr/lib";
\end{verbatim}

to:

\begin{verbatim}
char name[] = "/usr/lib";
\end{verbatim}

\textbf{Rationale}: Austin Group Defect Report(s) applied: 928. See \url{http://austingroupbugs.net/view.php?id=928}.

In the following table, the input string is the value pointed to by \texttt{path}, and the output string is the return value of the \texttt{basename()} function.

<table>
<thead>
<tr>
<th>Input String</th>
<th>Output String</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;/usr/lib&quot;</td>
<td>&quot;lib&quot;</td>
</tr>
<tr>
<td>&quot;/usr/&quot;</td>
<td>&quot;usr&quot;</td>
</tr>
<tr>
<td>&quot;/&quot;</td>
<td>&quot;/&quot;</td>
</tr>
</tbody>
</table>

Copyright © 2016 IEEE and The Open Group. All rights reserved.
Sample Input and Output Strings for the `basename()` and `dirname()` functions and the `basename` and `dirname` utilities

<table>
<thead>
<tr>
<th>basename() and dirname() functions path argument</th>
<th>string returned by basename()</th>
<th>string returned by dirname()</th>
<th>basename and dirname utilities string operand</th>
<th>output written by basename utility</th>
<th>output written by dirname utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;usr&quot;</td>
<td>&quot;usr&quot;</td>
<td>&quot;.&quot;</td>
<td>usr</td>
<td>usr</td>
<td>.</td>
</tr>
<tr>
<td>&quot;/usr/&quot;</td>
<td>&quot;/usr&quot;</td>
<td>&quot;.&quot;</td>
<td>/usr/</td>
<td>/usr/</td>
<td>.</td>
</tr>
<tr>
<td>&quot;//&quot;</td>
<td>&quot;.&quot;</td>
<td>&quot;.&quot;</td>
<td>&quot;&quot;</td>
<td>. or empty string</td>
<td>.</td>
</tr>
<tr>
<td>&quot;///&quot;</td>
<td>&quot;///&quot;</td>
<td>&quot;///&quot;</td>
<td>&quot;/&quot;</td>
<td>/ or //</td>
<td>/ or //</td>
</tr>
<tr>
<td>&quot;/usr/&quot;</td>
<td>&quot;/usr/&quot;</td>
<td>&quot;///&quot;</td>
<td>&quot;/usr/&quot;</td>
<td>/usr/</td>
<td>/</td>
</tr>
<tr>
<td>&quot;/usr/lib&quot;</td>
<td>&quot;lib&quot;</td>
<td>&quot;/usr/&quot;</td>
<td>/usr/lib</td>
<td>lib</td>
<td>/usr</td>
</tr>
<tr>
<td>&quot;/home//dwc//test&quot;</td>
<td>&quot;test&quot;</td>
<td>&quot;/home//dwc//test&quot;</td>
<td>&quot;/home//dwc//test&quot;</td>
<td>test</td>
<td>/home//dwc</td>
</tr>
</tbody>
</table>


Change Number: XSH/TC2/D4/0050 [822]

On Page: 617 Line: 21086 Section: bind()

In the ERRORS section, change from:

[ENOENT]
A component of the pathname does not name an existing file or the pathname is an empty string.

[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.


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


Change Number: XSH/TC2/D4/0052 [663]

On Page: 622 Line: 21263 Section: btowc()

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]


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.
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.


Change Number: XSH/TC2/D4/0054 [645]

On Page: 639 Line: 21683 Section: catopen()}

In the DESCRIPTION section, change from:

complete name

to:

pathname


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 ...

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.


In the NAME section, delete:

relative to directory file descriptor


Before the `fchmodat` SYNOPSIS line, insert a line with OH shading:

```c
#include <fcntl.h>
```


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.

1998  Change Number: XSH/TC2/D4/0060 [817]

1999  On Page: 656 Line: 22207 Section: chmod()
2001  In the ERRORS section, for the [EACCES] error, change from:
2002  fd was not opened with O_SEARCH and ...
2003  to:
2004  The access mode of the open file description associated with fd is not O_SEARCH and ...

2006  Change Number: XSH/TC2/D4/0061 [893]

2009  In the EXAMPLES section, change from:
2010  chmod("home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
2011  status = stat("home/cnd/mod1", &buffer);
2012  to:
2013  chmod("/home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
2014  status = stat("/home/cnd/mod1", &buffer);

2016  Change Number: XSH/TC2/D4/0062 [873]

2019  In the NAME section, delete:
2020  relative to directory file descriptor

2022  Change Number: XSH/TC2/D4/0063 [591]

2023  On Page: 659 Line: 22312 Section: chown()
2024  Before the fchownat SYNOPSIS line, insert a line with OH shading:
#include <fcntl.h>


**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.


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()


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.
```


**Change Number:** XSH/TC2/D4/0066 [817]

On Page: 660 Line: 22376 Section: chown()


72
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 ...
```


**Change Number:** XSH/TC2/D4/0067 [686]


In the APPLICATION USAGE section, 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.


**Change Number:** XSH/TC2/D4/0068 [909]

On Page: 671 Line: 22759 Section: clock_nanosleep()


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.
```


**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.


**Change Number:** XSH/TC2/D4/0070 [810]

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]

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]

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 *double*, $710.5 < |x|$ implies that $\cosh(x)$ has overflowed.

Change Number: XSH/TC2/D4/0073 [899]

On Page: 705 Line: 23827 Section: crypt()

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.


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, ...


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.
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 \texttt{errno}

and shall set \texttt{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].

Change Number: XSH/TC2/D4/0077 [830]

On Page: 725 Line: 24372 Section: dirname()


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 \texttt{dirname()} function shall not perform pathname resolution; the result shall not be affected by whether or not \texttt{path} exists or by its file type.

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.
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()`.


**Change Number:** XSH/TC2/D4/0079 [830]

On Page: 725 Line: 24378 Section: `dirname()`


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.


**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.


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.


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.


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()


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()


In the SEE ALSO section, add random().

Change Number: XSH/TC2/D4/0084 [753]

On Page: 743 Line: 24972 Section: duplocale()

In the EXAMPLES section, change from:

```c
return retval; }
```

to:

```c
return retval;
```


Change Number: XSH/TC2/D4/0085 [899]

On Page: 745 Line: 25023 Section: encrypt()


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.


Change Number: XSH/TC2/D4/0086 [493]


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.


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.

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:

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.

This item is a layered change on XSH/TC1/D5/0081 [75,428].
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.


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 getprotobynumber(), getprotobynumber(), or getprotoent(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.


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]


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, \texttt{getpwent}() shall open it and return a pointer to a \texttt{passwd} structure containing the first entry in the database.

In the DESCRIPTION section, at line 25267 change from:

The \texttt{setpwent}() function effectively rewinds the user database to allow repeated searches.

to:

The \texttt{setpwent}() function shall rewind the user database so that the next \texttt{getpwent}() call returns the first entry, allowing repeated searches.

In the DESCRIPTION section, at line 25268 change from:

The \texttt{endpwent}() function may be called to close the user database when processing is complete.

to:

The \texttt{endpwent}() function shall close the user database.

The \texttt{setpwent}() and \texttt{endpwent}() functions shall not change the setting of \texttt{errno} if successful.

On error, the \texttt{setpwent}() and \texttt{endpwent}() functions shall set \texttt{errno} to indicate the error.

Since no value is returned by the \texttt{setpwent}() and \texttt{endpwent}() functions, an application wishing to check for error situations should set \texttt{errno} to 0, then call the function, then check \texttt{errno}.

In the RETURN VALUE section, at line 25271 change from:

The \texttt{getpwent}() function shall return a null pointer on end-of-file or error.

to:

On successful completion, \texttt{getpwent}() shall return a pointer to a \texttt{passwd} structure. On end-of-file, \texttt{getpwent}() shall return a null pointer and shall not change the setting of \texttt{errno}. On error, \texttt{getpwent}() shall return a null pointer and \texttt{errno} shall be set to indicate the error.


The changes are made for consistency.

\textbf{Change Number: XSH/TC2/D4/0093 [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]


In the ERRORS section, insert before the EIO error:

[EINTR] A signal was caught during the operation.

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.

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.

Change Number: XSH/TC2/D4/0097 [584]

On Page: 780 Line: 26023 Section: exec

In the RATIONALE section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XSH/TC2/D4/0098 [898]

On Page: 780 Line: 26024 Section: exec

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.


Change Number: XSH/TC2/D4/0099 [734]

On Page: 782 Line: 26142 Section: exec

In the RATIONALE section, change from:

Alternatively, an function like openat()

to:

Alternatively, a function like openat()


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, \textit{status} \& 0377) shall be available from \texttt{wait()} and \texttt{waitpid()}; the full value shall be available from \texttt{waitid()} and in the \texttt{siginfo_t} passed to a signal handler for SIGCHLD.

\textit{Rationale:} Austin Group Defect Report(s) applied: 594. See \url{http://austingroupbugs.net/view.php?id=594}.

The standard is unclear about when \texttt{si_status} contains an exit status and when it contains a signal.

\textbf{Change Number: XSH/TC2/D4/0101} [630]

On Page: 787 Line: 26297 Section: \texttt{exp()}

In the APPLICATION USAGE section, delete:

Note that for IEEE Std 754-1985 \texttt{double}, 709.8 < \textit{x} implies \texttt{exp(\textit{x})} has overflowed. The value \textit{x} < -708.4 implies \texttt{exp(\textit{x})} has underflowed.

\textit{Rationale:} Austin Group Defect Report(s) applied: 630. See \url{http://austingroupbugs.net/view.php?id=630}.

\textbf{Change Number: XSH/TC2/D4/0102} [630]

On Page: 789 Line: 26366 Section: \texttt{exp2()}

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 \texttt{double}, 1024 \leq \textit{x} implies \texttt{exp2(\textit{x})} has overflowed. The value \textit{x} < -1022 implies \texttt{exp(\textit{x})} has underflowed.

\textit{Rationale:} Austin Group Defect Report(s) applied: 630. See \url{http://austingroupbugs.net/view.php?id=630}.

\textbf{Change Number: XSH/TC2/D4/0103} [630]

On Page: 791 Line: 26426 Section: \texttt{expm1()}

In the APPLICATION USAGE section, delete:

For IEEE Std 754-1985 \texttt{double}, 709.8 < \textit{x} implies \texttt{expm1(\textit{x})} has overflowed.

\textit{Rationale:} Austin Group Defect Report(s) applied: 630. See \url{http://austingroupbugs.net/view.php?id=630}.

\textbf{Change Number: XSH/TC2/D4/0104} [555]

On Page: 806 Line: 26839 Section: \texttt{fclose()}

In the APPLICATION USAGE section, change from:

None.

\textit{to:}

Since after the call to \texttt{fclose()} any use of \texttt{stream} results in undefined behavior, \texttt{fclose()} should not be used on \texttt{stdin}, \texttt{stdout}, or \texttt{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.


Change Number: XSH/TC2/D4/0105 [835]

On Page: 807 Line: 26877 Section: fcntl()

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.


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 \texttt{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 \texttt{fildes} does not
refer to a socket, the results are unspecified.

\textit{Rationale}: Austin Group Defect Report(s) applied: 677. See \url{http://austingroupbugs.net/view.php?id=677}.

\textbf{Change Number: XSH/TC2/D4/0107} [484]

On Page: 808 Line: 26923 Section: \texttt{fcntl()}

In the DESCRIPTION section, for \texttt{F_GETLK} change from:

Get the first lock which blocks ...

to:

Get any lock which blocks ...

\textit{Rationale}: Austin Group Defect Report(s) applied: 484. See \url{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.

\textbf{Change Number: XSH/TC2/D4/0108} [675]

On Page: 810 Line: 27040 Section: \texttt{fcntl()}


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

\texttt{[ESRCH]}

The \texttt{cmd} argument is \texttt{F_SETOWN} and no process or process group can be found corresponding to
that specified by \texttt{arg}.

\textit{Rationale}: Austin Group Defect Report(s) applied: 675. See \url{http://austingroupbugs.net/view.php?id=675}.

\textbf{Change Number: XSH/TC2/D4/0109} [675,677]

On Page: 811 Line: 27044 Section: \texttt{fcntl()}

In the ERRORS section, add to the "may fail" errors:

\texttt{[EINVAL]}

The \texttt{cmd} argument is \texttt{F_SETOWN} and the value of the argument is not valid as a process or process
group identifier.

\texttt{[EPERM]}

The \texttt{cmd} argument is \texttt{F_SETOWN} and the calling process does not have permission to send a
SIGURG signal to any process specified by \texttt{arg}.

On Page: 812 Line: 27107 Section: \texttt{fcntl()}
In the APPLICATION USAGE section, add a new paragraph:

On systems which do not perform permission checks at the time of an \texttt{fcntl()} call with \texttt{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 \texttt{fcntl()} has no way of discovering that this has happened. A call to \texttt{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 \texttt{kill()}.  


\textbf{Change Number: XSH/TC2/D4/0110 [501]}

On Page: 815 Line: 27225 Section: fdatasync()

In the ERRORS section, change from:

The \texttt{fildes} argument is not a valid file descriptor open for writing.

to:

The \texttt{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 \texttt{fdatasync()}.

\textit{Rationale}: Austin Group Defect Report(s) applied: 501. See \url{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.

\textbf{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]


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,


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.

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.
- [EBADF] The file descriptor underlying stream is not a valid file descriptor.


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()`.


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
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: fnemopen()

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.


Change Number: XSH/TC2/D4/0119 [818]

On Page: 868 Line: 28284 Section: fnemopen()

In the FUTURE DIRECTIONS section, change from:

None.

to:

A future revision of this standard may require support of zero length buffer streams explicitly.


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.

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).

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.

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()


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.


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.

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}.


Change Number: XSH/TC2/D4/0126 [894]

On Page: 895 Line: 29808 Section: fprintf()


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).

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.


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()


In the SEE ALSO section, add <inttypes.h> to the list.


Change Number: XSH/TC2/D4/0129 [926]

On Page: 913 Line: 30537 Section: fread()


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.


95

Copyright © 2016 IEEE and The Open Group. All rights reserved.
Change Number: XSH/TC2/D4/0130 [939]

On Page: 916 Line: 30649 Section: freeaddrinfo()

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.


Change Number: XSH/TC2/D4/0131 [979]

On Page: 917 Line: 30683 Section: freeaddrinfo()

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, *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.


Change Number: XSH/TC2/D4/0132 [918]

On Page: 919 Line: 30762 Section: freeaddrinfo()

In the EXAMPLES section, change from:

```c
struct addrinfo hints = {};
```

```c
96
```
struct addrinfo hints = {0};


Change Number: XSH/TC2/D4/0133 [934]

On Page: 920 Line: 30796 Section: freeaddrinfo()

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.


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] or [ENOTDIR] 97
```

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.


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()

In the SEE ALSO section, add `<inttypes.h>` to the list.


**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:

```c
#include <fcntl.h>
```


**Change Number:** XSH/TC2/D4/0137 [817]

On Page: 945 Line: 31686 Section: fstatat()

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.


**Change Number:** XSH/TC2/D4/0138 [817]

On Page: 946 Line: 31717 Section: fstatat()

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

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.

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()


Change Number: XSH/TC2/D4/0144 [73]

On Page: 973 Line: 32586 Section: fwprintf()

A clarification has been made in the C11 standard.

Change Number: XSH/TC2/D4/0145 [894]

On Page: 974 Line: 32634 Section: fwprintf()

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).


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.


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()

In the SEE ALSO section, add <inttypes.h> to the list.

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().


A clarification has been made in the C11 standard.

Change Number: XSH/TC2/D4/0149 [823]

On Page: 988 Line: 33162 Section: fwscanf()


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.


Change Number: XSH/TC2/D4/0150 [936]

On Page: 989 Line: 33200 Section: fwscanf()


In the SEE ALSO section, add <inttypes.h> to the list.

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 ...

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()


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:


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

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.

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: getegid()

In the DESCRIPTION section, on line 33813 add a sentence:

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.


**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()` 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 the string content might be overwritten by a subsequent call to `getenv()`, `[CX]` `setenv()`, `unsetenv()`, `[CX]` `[XSI]or `putenv()` 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](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, \texttt{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 \texttt{(uid_t)-1} and set \texttt{errno} to indicate such an environment; doing so does not violate this standard, since such an environment is already an extension.

\textit{Rationale:} Austin Group Defect Report(s) applied: 511. See \url{http://austingroupbugs.net/view.php?id=511}.

\textbf{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

\texttt{The getgid()} function shall not modify \texttt{errno}.

In the RATIONALE section, on line 33977 change from:

None.

to:

In a conforming environment, \texttt{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 \texttt{(gid_t)-1} and set \texttt{errno} to indicate such an environment; doing so does not violate this standard, since such an environment is already an extension.

\textit{Rationale:} Austin Group Defect Report(s) applied: 511. See \url{http://austingroupbugs.net/view.php?id=511}.

\textbf{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 \texttt{errno} to 0 before calling \texttt{getgrgid()}(). If \texttt{getgrgid()} returns a null pointer and \texttt{errno} is set to non-zero, an error occurred.

\textit{Rationale:} Austin Group Defect Report(s) applied: 808. See \url{http://austingroupbugs.net/view.php?id=808}.

\textbf{Change Number: XSH/TC2/D4/0161} [808]

On Page: 1014 Line: 34019 Section: getgrgid()

In the RETURN VALUE section, change from:

On error, \texttt{errno} shall be set to indicate the error.

to:

If the requested entry was not found, \texttt{errno} shall not be changed. On error, \texttt{errno} shall be set to indicate the error.

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.


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.


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.

Change Number: XSH/TC2/D4/0165 [808]

On Page: 1018 Line: 34147 Section: getgrnam()

In the RETURN VALUE section, change from:

On error, \texttt{errno} shall be set to indicate the error.

to:

If the requested entry was not found, \texttt{errno} shall not be changed. On error, \texttt{errno} shall be set to indicate the error.

\textit{Rationale}: Austin Group Defect Report(s) applied: 808. See \url{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 \texttt{getgrent()}, \texttt{getgrgid()}, or \texttt{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 \texttt{getgrent()}, \texttt{getgrgid()}, or \texttt{getgrnam}(). The returned pointer, and pointers within the structure, might also be invalidated if the calling thread is terminated.

\textit{Rationale}: Austin Group Defect Report(s) applied: 656. See \url{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 \texttt{errno} to 0 before calling \texttt{getgrnam}(). If \texttt{errno} is set on return, an error occurred.

\textit{Rationale}: Austin Group Defect Report(s) applied: 808. See \url{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.


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>.


Change Number: XSH/TC2/D4/0170 [808]


In the RETURN VALUE section, change from:
On error, \texttt{errno} shall be set to indicate the error.

If the requested entry was not found, \texttt{errno} shall not be changed. On error, \texttt{errno} shall be set to indicate the error.

\textit{Rationale: Austin Group Defect Report(s) applied: 808. See \url{http://austingroupbugs.net/view.php?id=808}.}

\textbf{Change Number: XSH/TC2/D4/0171 [656]}

On Page: 1057 Line: 35319 Section: \texttt{getpwnam()}

In the \textit{RETURN VALUE} section, change from:

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

\textit{Rationale: Austin Group Defect Report(s) applied: 656. See \url{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.

\textbf{Change Number: XSH/TC2/D4/0172 [808]}

On Page: 1061 Line: 35451 Section: \texttt{getpwuid()}


In the \textit{RETURN VALUE} section, change from:

On error, \texttt{errno} shall be set to indicate the error.

\textit{Rationale: Austin Group Defect Report(s) applied: 808. See \url{http://austingroupbugs.net/view.php?id=808}.}
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.

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/D4/0173 [656]

On Page: 1078 Line: 35928 Section: getsubopt()

In the SYNOPSIS section, add a CX margin marker and shading to all the synopsis.

This function in <stdlib.h> is an extension over the C standard contents for the header.

Change Number: XSH/TC2/D4/0174 [791]

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.


Change Number: XSH/TC2/D4/0176 [897]

On Page: 1083 Line: 36124 Section: getuid()


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.

```c
#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.

```c
#include <unistd.h>
...
seteuid(getuid());
```


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]).


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.

```c
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.


Change Number: XSH/TC2/D4/0179 [743]

On Page: 1119 Line: 37298 Section: initstate()

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.


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

3294 to:

is a wide-character code corresponding to a valid character in the locale used by the function


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

3301 to:

is a wide-character code corresponding to a valid character in the locale used by the function


Change Number: XSH/TC2/D4/0182 [685]

On Page: 1173 Line: 38974 Section: iswblank()
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


Change Number: XSH/TC2/D4/0183 [685]

On Page: 1174 Line: 39015 Section: iswctrl()

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


Change Number: XSH/TC2/D4/0184 [799]

On Page: 1176 Line: 39070 Section: iswctype()


In the RETURN VALUE section, change from:

If `charclass` is 0

to:

If `charclass` is `(wctype_t)0`


Change Number: XSH/TC2/D4/0185 [799]

On Page: 1177 Line: 39130 Section: iswctype()


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.

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


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


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


Change Number: XSH/TC2/D4/0189 [685]

On Page: 1184 Line: 39293 Section: iswprint()
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


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


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


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

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


Change Number: XSH/TC2/D4/0194 [765]


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().


Change Number: XSH/TC2/D4/0195 [873]

On Page: 1216 Line: 40102 Section: link()

In the NAME section, delete:

relative to two directory file descriptors

3415  Change Number: XSH/TC2/D4/0196 [591]
3416  On Page: 1216 Line: 40106 Section: link()
3417  Before the linkat SYNOPSIS line, insert a line with OH shading:
3418  #include <fcntl.h>
3420  Change Number: XSH/TC2/D4/0197 [817]
3421  On Page: 1216 Line: 40126 Section: link()
3423  In the DESCRIPTION section, change from:
3424  If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches
3425  are permitted using the current permissions of the directory underlying the file descriptor. If the file
3426  descriptor was opened with O_SEARCH, the function shall not perform the check.
3427  to:
3428  If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the
3429  function shall check whether directory searches are permitted using the current permissions of the directory
3430  underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.
3432  Change Number: XSH/TC2/D4/0198 [822]
3433  On Page: 1217 Line: 40156 Section: link()
3435  In the RETURN VALUE section, add after the [ENOENT] error:
3436  [ENOENT] or [ENOTDIR]
3437  The path1 argument names an existing non-directory file, and the path2 argument contains at least
3438  one non-<slash> character and ends with one or more trailing <slash> characters. If path2 without
3439  the trailing <slash> characters would name an existing file, an [ENOENT] error shall not occur.
3441  Change Number: XSH/TC2/D4/0199 [817]
3442  On Page: 1217 Line: 40170 Section: link()
3444  In the ERRORS section, add:
3445  [EACCES]
The access mode of the open file description associated with \texttt{fd1} or \texttt{fd2} is not \texttt{O_SEARCH} and the permissions of the directory underlying \texttt{fd1} or \texttt{fd2} respectively do not permit directory searches.

\textit{Rationale:} Austin Group Defect Report(s) applied: 817. See \url{http://austingroupbugs.net/view.php?id=817}.

\textbf{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 \texttt{localeconv()}. In addition, calls to \texttt{setlocale()} with the categories \texttt{LC\_ALL}, \texttt{LC\_MONETARY}, or \texttt{LC\_NUMERIC} or calls to \texttt{uselocale()} which change the categories \texttt{LC\_MONETARY} or \texttt{LC\_NUMERIC} may overwrite the contents of the structure.

Change 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 might be overwritten by a subsequent call to \texttt{localeconv()}. In addition, the returned pointer, and pointers within the structure, might be invalidated or the storage areas might be overwritten by subsequent calls to \texttt{setlocale()} with the categories \texttt{LC\_ALL}, \texttt{LC\_MONETARY}, or \texttt{LC\_NUMERIC}, \texttt{LC\_MONETARY} or \texttt{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.

\textit{Rationale:} Austin Group Defect Report(s) applied: 656. See \url{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.

\textbf{Change Number: XSH/TC2/D4/0201 [664]}

On Page: 1238 Line: 40811 Section: localtime()

In the DESCRIPTION section, change from:

If \texttt{localtime()} does not set \texttt{tzname}, it shall not set \texttt{daylight} and shall not set \texttt{timezone}.

Change to:

If \texttt{localtime()} sets \texttt{tzname}, it shall also set \texttt{daylight} and \texttt{timezone}. If \texttt{localtime()} does not set \texttt{tzname}, it shall not set \texttt{daylight} and shall not set \texttt{timezone}.

\textit{Rationale:} Austin Group Defect Report(s) applied: 664. See \url{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().


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 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.


**Change Number:** XSH/TC2/D4/0204 [663,674]

On Page: 1270 Line: 41775 Section: mblen()


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()


In the ERRORS section, change from:

[EILSEQ]
An invalid character sequence is detected.

to:

[EILSEQ] 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: 1275 Line: 41890 Section: mbrtowc()


In the ERRORS section, change from:

[EILSEQ] An invalid character sequence is detected.
An invalid character sequence is detected. [CX]In the POSIX locale an EILSEQ error cannot occur since all byte values are valid characters.[/CX]


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.


Change Number: XSH/TC2/D4/0206 [663]

On Page: 1278 Line: 41998 Section: mbsrtowcs()

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]


Change Number: XSH/TC2/D4/0207 [601]
On Page: 1278 Line: 42008 Section: mbstowcs()

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.


Change Number: XSH/TC2/D4/0208 [663,674]

On Page: 1279 Line: 42051 Section: mbstowcs()


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]


Change Number: XSH/TC2/D4/0209 [663,674]

On Page: 1281 Line: 42104 Section: mbtowc()


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]


Change Number: XSH/TC2/D4/0210 [873]

On Page: 1289 Line: 42299 Section: mkdir()

In the NAME section, delete:

relative to directory file descriptor


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>
```


Change Number: XSH/TC2/D4/0212 [817]

On Page: 1289 Line: 42323 Section: mkdir()

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.


Change Number: XSH/TC2/D4/0213 [817]

On Page: 1290 Line: 42348 Section: mkdir()

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.


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.


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.

> 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.

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.


**Change Number:** XSH/TC2/D4/0216 [873]

On Page: 1295 Line: 42496 Section: mkfifo()


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>
```

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.


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.

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

fd was not opened with O_SEARCH and ...

Copyright © 2016 IEEE and The Open Group. All rights reserved.
The access mode of the open file description associated with \textit{fd} is not \texttt{O\_SEARCH} and ...

\textit{Rationale}: Austin Group Defect Report(s) applied: 817. See \url{http://austingroupbugs.net/view.php?id=817}.

\textbf{Change Number: XSH/TC2/D4/0221} [591]

On Page: 1297 Line: 42594 Section: mkfifo()

Add a reference to \texttt{<fcntl.h>} to the SEE ALSO list.

\textit{Rationale}: Austin Group Defect Report(s) applied: 591. See \url{http://austingroupbugs.net/view.php?id=591}.

\textbf{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:

\begin{verbatim}
#include <fcntl.h>
\end{verbatim}

\textit{Rationale}: Austin Group Defect Report(s) applied: 591. See \url{http://austingroupbugs.net/view.php?id=591}.

\textbf{Change Number: XSH/TC2/D4/0223} [817]

On Page: 1299 Line: 42669 Section: mknod()


In the DESCRIPTION section, change from:

If the file descriptor was opened without \texttt{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 \texttt{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 \texttt{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 \texttt{O\_SEARCH}, the function shall not perform the check.

\textit{Rationale}: Austin Group Defect Report(s) applied: 817. See \url{http://austingroupbugs.net/view.php?id=817}.

\textbf{Change Number: XSH/TC2/D4/0224} [822]

On Page: 1299 Line: 42689 Section: mknod()

In the ERRORS section, change from:

[\texttt{ENOENT}]

A component of the path prefix specified by \textit{path} does not name an existing directory or \textit{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.

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()

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 ...


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.


Change Number: XSH/TC2/D4/0227 [902]


In the NAME section, change from:

create a unique directory
to:

create a unique file

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


Change Number: XSH/TC2/D4/0229 [852]

On Page: 1311 Line: 43068 Section: mmap()


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()

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()

In the RATIONALE section, change from:

and the MEMLOCK_FUTURE argument

to:

and the MCL_FUTURE argument


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.

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.


---

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>`


---

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*


---

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.

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.


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()

In the NAME section, delete:

relative to directory file descriptor


Change Number: XSH/TC2/D4/0236 [835]

On Page: 1379 Line: 45169 Section: open()

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].

On Page: 1379 Line: 45193 Section: open()

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()

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.


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()

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.


Change Number: XSH/TC2/D4/0239 [835]

On Page: 1382 Line: 45300 Section: open()

In the RETURN VALUE section, change from:
representing the lowest numbered unused file descriptor

to:

representing the file descriptor


**Change Number:** XSH/TC2/D4/0240 [847]

On Page: 1382 Line: 45314 Section: open()


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.


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.
```


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()


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 ...


Change Number: XSH/TC2/D4/0243 [943]

On Page: 1383 Line: 45361 Section: open()


In the ERRORS section, before the [ETXTBSY] error, add:

[EOPNOTSUPP] The path argument names a socket.


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().


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.

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.


Change Number: XSH/TC2/D4/0247 [835]

On Page: 1400 Line: 45837 Section: pipe()


Change from:

To:

Their integer values shall be the two lowest available at the time of the pipe() call.

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


Change Number: XSH/TC2/D4/0248 [467,835]

On Page: 1400 Line: 45851 Section: pipe()


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.


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().

Change Number: XSH/TC2/D4/0250 [683]

On Page: 1404 Line: 45987 Section: poll()

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).


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.


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()`.
```

```c
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
...
```

Rationale:


Change Number: XSH/TC2/D4/0253 [835]

On Page: 1420 Line: 46464 Section: `posix_openpt()`

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:

Change Number: XSH/TC2/D4/0254 [835]

On Page: 1420 Line: 46474 Section: posix_openpt()

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.


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.


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()

Change from:
The file descriptor is used by other functions to refer to that typed memory object.

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.


**Change Number**: XSH/TC2/D4/0257 [835]

On Page: 1516 Line: 48924 Section: posix_typed_mem_open()

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.


**Change Number**: XSH/TC2/D4/0258 [835]

On Page: 1516 Line: 48939 Section: posix_typed_mem_open()

Change from:

... return a non-negative integer representing the lowest numbered unused file descriptor.

to:

... return a non-negative integer representing the file descriptor.


**Change Number**: XSH/TC2/D4/0259 [680]

On Page: 1526 Line: 49239 Section: pselect()

In the ERRORS section, change from:

[EINTR]

The function was interrupted before any of the selected events occurred and before the timeout interval expired.
4175  to:

4176  [EINTR]
4177  The function was interrupted while blocked waiting for any of the selected descriptors to
4178  become ready and before the timeout interval expired.


4180  **Change Number:** XSH/TC2/D4/0260 [629]

4181  On Page: 1528 Line: 49313,49319 Section: psiginfo()

4182  In the NAME section, change from:

4183  print signal information to standard error

4184  to:

4185  write signal information to standard error

4186  In the DESCRIPTION section, at line 49319 change from:

4187  The *psiginfo()* and *psignal()* functions shall print a message out on *stderr* associated with a signal number.

4188  If message is not null and is not the empty string, then the string pointed to by the *message* argument shall

4189  be printed first, followed by a <colon>, a <space>, and the signal description string indicated by *signum*, or

4190  by the signal associated with *pinfo*. If the *message* argument is null or points to an empty string, then only

4191  the signal description shall be printed. For *psiginfo()*, the argument *pinfo* references a valid siginfo_t

4192  structure. For *psignal()*, if *signum* is not a valid signal number, the behavior is implementation-defined.

4193  to:

4194  The *psiginfo()* and *psignal()* functions shall write a language-dependent message associated with a signal

4195  number to the standard error stream as follows:

4196  • First, if *message* is not a null pointer and is not the empty string, the string pointed to by the

4197  *message* argument shall be written, followed by a <colon> and a <space>.

4198  • Then the signal description string associated with *signum* or with the signal indicated by *pinfo*

4199  shall be written, followed by a <newline>.

4200  For *psiginfo()*, the application shall ensure that the argument *pinfo* references a valid siginfo_t structure.

4201  For *psignal()*, if *signum* is not a valid signal number, the behavior is implementation-defined.


4203  **Change Number:** XSH/TC2/D4/0261 [858]

4204  On Page: 1529 Line: 49363 Section: pthread_atfork()


4206  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()

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()

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.


Change Number: XSH/TC2/D4/0262 [757]

On Page: 1541 Line: 49831 Section: pthread_attr_getinheritsched()

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.

Change Number: XSH/TC2/D4/0263 [757]

On Page: 1545 Line: 49961 Section: pthread_attr_getschedpolicy()

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.


Change Number: XSH/TC2/D4/0264 [757]

On Page: 1547 Line: 50023 Section: pthread_attr_getscope()

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.


Change Number: XSH/TC2/D4/0265 [757]

On Page: 1552 Line: 50161 Section: pthread_attr_getstacksize()

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.

Change Number: XSH/TC2/D4/0266 [972]

On Page: 1562 Line: 50259-50261 Section: pthread_barrier_destroy()

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.

Change Number: XSH/TC2/D4/0267 [757]

On Page: 1568 Line: 50418-50421 Section: pthread_barrierattr_getpshared()

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.


Change Number: XSH/TC2/D4/0268 [624]

On Page: 1574 Line: 50552 Section: pthread_cleanup_pop()

In the DESCRIPTION section, 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.

These functions may be implemented as macros.

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.


**Change Number**: XSH/TC2/D4/0269 [972]

On Page: 1582 Line: 50872-50874 Section: pthread_cond_destroy()


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.


**Change Number**: XSH/TC2/D4/0270 [910]

On Page: 1583 Line: 50927 Section: pthread_cond_destroy()


In the EXAMPLES section, change from:

```c
rp->notbusy
```

to:

```c
ep->notbusy
```


**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],


Change Number: XSH/TC2/D4/0272 [972]

On Page: 1596 Line: 51368-51371 Section: pthread_condattr_getpshared()

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.


Change Number: XSH/TC2/D4/0273 [757]

On Page: 1597 Line: 51409 Section: pthread_condattr_getpshared()

In the CHANGE HISTORY section for Issue 7, change from:

The pthread_condattr_getpshared() and pthread_condattr_getpshared() functions are moved from the Threads option.

to:

The pthread_condattr_getpshared() and pthread_condattr_getpshared() functions are marked only as part of the Thread Process-Shared Synchronization option as the Threads option is now part of the Base.


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

Change Number: XSH/TC2/D4/0275 [757]

On Page: 1611 Line: 51797 Section: pthread_getcpuclockid()

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.


Change Number: XSH/TC2/D4/0276 [757]

On Page: 1614 Line: 51885 Section: pthread_getschedparam()

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.


Change Number: XSH/TC2/D4/0277 [765]

On Page: 1625 Line: 52287 Section: pthread_kill()

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()

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.


**Change Number:** XSH/TC2/D4/0278 [811]

On Page: 1628 Line: 52362 Section: pthread_mutex_destroy()


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.


**Change Number:** XSH/TC2/D4/0279 [972]

On Page: 1628 Line: 52369-52371 Section: pthread_mutex_destroy()


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.


**Change Number:** XSH/TC2/D4/0280 [811]

On Page: 1632 Line: 52537-52557 Section: pthread_mutex_destroy()


In the RATIONALE section, change from:
A mutex can be destroyed immediately after it is unlocked. For example, consider the following code:

```c
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);
        pthread_mutex_destroy(&op->om);
        free(op);
    } else
        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).


Change Number: XSH/TC2/D4/0281 [972]

On Page: 1657 Line: 53355-53358 Section: pthread_mutexattr_getpshared()

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.


Change Number: XSH/TC2/D4/0282 [757]

On Page: 1658 Line: 53396 Section: pthread_mutexattr_getpshared()

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.


Change Number: XSH/TC2/D4/0283 [748]


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()`.


Change Number: XSH/TC2/D4/0284 [863]


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.


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.


Change Number: XSH/TC2/D4/0285 [874]

On Page: 1669 Line: 53628 Section: pthread_once()

In the RATIONALE section, change from:
extern int initialize_random();
to:
extern void initialize_random(void);


Change Number: XSH/TC2/D4/0286 [874]

On Page: 1670 Line: 53640 Section: pthread_once()

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()

In the RATIONALE section, change from:
extern int initialize_random();
to:
extern void initialize_random(void);


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>.


Change Number: XSH/TC2/D4/0288 [972]
On Page: 1671 Line: 53698-53701 Section: pthread_rwlock_destroy()


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.

See [xref to section 2.9.9] for further requirements.


Change Number: XSH/TC2/D4/0289 [758]
On Page: 1672 Line: 53728 Section: pthread_rwlock_destroy()


In the RATIONALE section, change from:

pthread_rwlockr_init()

to:

pthread_rwlock_init()
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).

4554
to:

4555

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.


4559
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()

4560

In the DESCRIPTION section, remove the line:

Implementations may favor writers over readers to avoid writer starvation.


4564
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()

4568

In the DESCRIPTION section, change from:

4577
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.


**Change Number:** XSH/TC2/D4/0293 [757]

On Page: 1690 Line: 54240 Section: pthread_rwlockattr_getpshared()


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.


**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.


**Change Number:** XSH/TC2/D4/0295 [615]

On Page: 1695 Line: 54349 Section: pthread_setcancelstate()

In the FUTURE DIRECTIONS section, change from:

None.

Copyright © 2016 IEEE and The Open Group. All rights reserved.
The `pthread_setcancelstate()` function may be added to the table of async-signal-safe functions in section 2.4.3 on page 489.


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()

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.


Change Number: XSH/TC2/D4/0297 [972]

On Page: 1705 Line: 54589-54593 Section: pthread_spin_destroy()

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.


Change Number: XSH/TC2/D4/0298 [503]

On Page: 1712 Line: 54750 Section: pidname()

In the RETURN VALUE section, change from:
Upon failure `ptsname()` shall return a null pointer.

Upon failure `ptsname()` shall return a null pointer and may set `errno`.


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.


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.

On Page: 1734 Line: 55339 Section: rand()
In the APPLICATION USAGE section, change from:
The \texttt{drand48()} function provides a much more elaborate random number generator.
to:
The \texttt{drand48()} and \texttt{random()} functions provide much more elaborate pseudo-random number generators.

On Page: 1734 Line: 55342-55343 Section: rand()
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()
In the SEE ALSO section, add \texttt{initstate()}.

\textit{Rationale}: Austin Group Defect Report(s) applied: 743. See \url{http://austingroupbugs.net/view.php?id=743}.

On Page: 1739 Line: 55477 Section: read()
In the DESCRIPTION section, change from:
The \texttt{pread()} function shall be equivalent to \texttt{read()}, except that it shall read from a given position in the file without changing the file pointer.
to:
The \texttt{pread()} function shall be equivalent to \texttt{read()}, except that it shall read from a given position in the file without changing the file offset.

\textit{Rationale}: Austin Group Defect Report(s) applied: 710. See \url{http://austingroupbugs.net/view.php?id=710}. 

Copyright © 2016 IEEE and The Open Group. All rights reserved.
Change Number: XSH/TC2/D4/0303 [676,710]

On Page: 1739 Line: 55506 Section: read()

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).

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.

This item is a layered change on XSH/TC1/D5/0484 [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:

```c
#include <fcntl.h>
```


Change Number: XSH/TC2/D4/0306 [817]

On Page: 1749 Line: 55858 Section: readlink()

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.


Change Number: XSH/TC2/D4/0307 [817]

On Page: 1750 Line: 55885 Section: readlink()

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 ...


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.

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.

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.


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.

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]

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 \textit{ptr} shall not be changed.


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 \textit{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 \textit{errno} set to an implementation-defined value[/CX].

to:

A null pointer shall be returned [CX]and, if \textit{ptr} is not a null pointer, \textit{errno} shall set to an implementation-defined value[/CX].

\textbf{Change Number: XSH/TC2/D4/0311 [873]}

On Page: 1781 Line: 56918 Section: rename()


In the NAME section, delete:

relative to directory file descriptor


\textbf{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:

\texttt{#include <fcntl.h>}


\textbf{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 \textit{new} shall remain visible to other processes throughout the renaming operation

to:

In this case, a link named \textit{new} shall remain visible to other threads throughout the renaming operation

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.


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 ...


Add a reference to <fcntl.h> to the SEE ALSO list.


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.


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()


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()


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.


**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.


**Change Number:** XSH/TC2/D4/0320 [899]

On Page: 1867 Line: 59571 Section: setkey()


In the FUTURE DIRECTIONS section, change from:

None.

Copyright © 2016 IEEE and The Open Group. All rights reserved.
A future version of the standard may mark this interface as obsolete or remove it altogether.


**Change Number:** XSH/TC2/D4/0321 [826]

On Page: 1869 Line: 59629 Section: setlocale()

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

> [CX]The `setlocale` function need not be thread-safe.[/CX]


**Change Number:** XSH/TC2/D4/0322 [826]

On Page: 1869 Line: 59637 Section: setlocale()

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](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.


**Change Number:** XSH/TC2/D4/0324 [835]

On Page: 1898 Line: 60324 Section: shm_open()
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.


Change Number: XSH/TC2/D4/0325 [835]

On Page: 1898 Line: 60335 Section: shm_open()

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.


Change Number: XSH/TC2/D4/0326 [835]

On Page: 1899 Line: 60375 Section: shm_open()

Change from:

... return a non-negative integer representing the lowest numbered unused file descriptor.

to:

... return a non-negative integer representing the file descriptor.


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


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.


Change Number: XSH/TC2/D4/0329 [690]


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:

[XS] 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.[/XS]

(Note the addition of XSI shading.)


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 ...


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.


Change Number: XSH/TC2/D4/0332 [844]

On Page: 1944 Line: 61771 Section: sigqueue()
In the SYNOPSIS section, change from:
const union sigval value
to:
union sigval value


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:

Copyright © 2016 IEEE and The Open Group. All rights reserved.
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 ...


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.


Change Number: XSH/TC2/D4/0335 [835]

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.


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].

Change Number: XSH/TC2/D4/0336 [835]


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].


Change Number: XSH/TC2/D4/0337 [483,835]


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.


Change Number: XSH/TC2/D4/0338 [738]


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 `(strlen(s, size) + 1)` bytes. Applications should not assume that `strndup()` will allocate `size + 1` bytes when `strlen(s)` is smaller than `size`.


Change Number: XSH/TC2/D4/0339 [656]

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]

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 ('-')


Change Number: XSH/TC2/D4/0341 [796]

On Page: 2012 Line: 63745 Section: strftime()

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


Change Number: XSH/TC2/D4/0342 [584]

In the RATIONALE section, change from:

leading minus-sign for %F,

to:

leading <hyphen-minus> for %F,


Change Number: XSH/TC2/D4/0343 [584]


In the RATIONALE section, change from:

leading minus-sign ('-') when using %Y

to:

leading <hyphen-minus> ('-') when using %Y


Change Number: XSH/TC2/D4/0344 [560]


In the DESCRIPTION section, change from:

the terminating NUL character

to:

any terminating NUL character


In the RETURN VALUE section, change from:

The strlen() function shall return an integer containing the smaller of either the length of the string pointed to by s or maxlen.

to:

The strlen() 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.

Change Number: XSH/TC2/D4/0345 [920]


In the DESCRIPTION section, change from:

... conversion specifier other than C, F, or Y.

to:

... conversion specifier other than C or Y.

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]


In the EXAMPLES section, change from:

Data-Plus-Time

to:

Date-Plus-Time


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.

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>,


**Change Number: XSH/TC2/D4/0349 [796]**

On Page: 2036 Line: 64495 Section: strtod()


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


**Change Number: XSH/TC2/D4/0350 [878]**

On Page: 2040 Line: 64631-64633 Section: strtok()


In the SYNOPSIS section, change from:

```c
char *strtok(char *restrict s1, const char *restrict s2);
char *strtok_r(char *restrict s, const char *restrict sep,
   char **restrict lasts);
```

to:

```c
char *strtok(char *restrict s, const char *restrict sep);
char *strtok_r(char *restrict s, const char *restrict sep,
   char **restrict state);
```

---

174

Copyright © 2016 IEEE and The Open Group. All rights reserved.
In the second, third and fourth paragraphs of the DESCRIPTION section, replace all instances of \( s1 \) with \( s \) and \( s2 \) with \( sep \).

In the first call to \( 
\text{strtokg(}) \), \( s \) points to a null-terminated string, \( sep \) to a null-terminated string of separator characters, and the value pointed to by \( \text{lasts} \) is ignored. The \( \text{strtokg(}) \) 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 \( \text{lasts} \) points.

In subsequent calls, \( s \) is a null pointer and \( \text{lasts} \) 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.

The \( \text{strtokg(}) \) function shall be equivalent to \( \text{strtok} \rg( \), except that \( \text{strtokg(}) \) shall be thread-safe and the argument \( \text{state} \) points to a user-provided pointer that allows \( \text{strtokg(}) \) to maintain state between calls which scan the same string. The application shall ensure that the pointer pointed to by \( \text{state} \) is unique for each string \( s \) being processed concurrently by \( \text{strtokg(}) \) calls. The application need not initialize the pointer pointed to by \( \text{state} \) to any particular value. The implementation shall not update the pointer pointed to by \( \text{state} \) to point (directly or indirectly) to resources, other than within the string \( s \), that need to be freed or released by the caller.

Note that if \( sep \) is the empty string, \( \text{strtok} \rg( \) and \( \text{strtokg(}) \) return a pointer to the remainder of the string being tokenized.


The behavior of \( \text{strtokg(}) \) with an empty \( sep \) is unclear.

Change Number: XSH/TC2/D4/0351 [892]
In the SYNOPSIS section, change from:

5226  long strtol(const char *restrict str, char **restrict endptr, int base);
5227  long long strtoll(const char *restrict str, char **restrict endptr,
5228       int base)

5230  to:

5231  long strtol(const char *restrict nptr, char **restrict endptr, int base);
5232  long long strtoll(const char *restrict nptr, char **restrict endptr,
5233       int base)
On Page: 2043 Line: 64744 Section: strtol()

In the DESCRIPTION section, change from:

These functions shall convert the initial portion of the string pointed to by \textit{str} ...

to:

These functions shall convert the initial portion of the string pointed to by \textit{nptr} ...

\textbf{Rationale}: Austin Group Defect Report(s) applied: 892. See \url{http://austingroupbugs.net/view.php?id=892}.

\textbf{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 \texttt{<hyphen-minus>},

\textbf{Rationale}: Austin Group Defect Report(s) applied: 584. See \url{http://austingroupbugs.net/view.php?id=584}.

\textbf{Change Number}: XSH/TC2/D4/0353 [796]

On Page: 2043 Line: 64775 Section: strtol()

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

\textbf{Rationale}: Austin Group Defect Report(s) applied: 796. See \url{http://austingroupbugs.net/view.php?id=796}.

\textbf{Change Number}: XSH/TC2/D4/0354 [892]

On Page: 2044 Line: 64778 Section: strtol()

In the DESCRIPTION section, change from:
... the value of \textit{str} is stored ...

to:

... the value of \textit{nptr} shall be stored ...

\textit{Rationale}: Austin Group Defect Report(s) applied: 892. See \url{http://austingroupbugs.net/view.php?id=892}.
This item is a layered change on XSH/TC1/D5/0616 [453].

\textbf{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>,

\textit{Rationale}: Austin Group Defect Report(s) applied: 584. See \url{http://austingroupbugs.net/view.php?id=584}.

\textbf{Change Number: XSH/TC2/D4/0356} [796]

On Page: 2049 Line: 64879 Section: strtoul()

In the DESCRIPTION section, change from:

In other than the C [CX]or POSIX[/CX] locale[/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

\textit{Rationale}: Austin Group Defect Report(s) applied: 796. See \url{http://austingroupbugs.net/view.php?id=796}.

\textbf{Change Number: XSH/TC2/D4/0357} [873]

On Page: 2057 Line: 65044 Section: symlink()

In the NAME section, delete:

relative to directory file descriptor

\textit{Rationale}: Austin Group Defect Report(s) applied: 873. See \url{http://austingroupbugs.net/view.php?id=873}.
5290  Change Number: XSH/TC2/D4/0358 [591]
5291  On Page: 2057 Line: 65048 Section: symlink()
5292  Before the symlinkat SYNOPSIS line, insert a line with OH shading:
5293  #include <fcntl.h>
5295  Change Number: XSH/TC2/D4/0359 [641]
5296  On Page: 2057 Line: 65053 Section: symlink()
5297  In the DESCRIPTION section, change from:
5298  The string pointed to by path1 shall be treated only as a character string and shall not be validated as a
5299  pathname.
5300  to:
5301  The string pointed to by path1 shall be treated only as a string and shall not be validated as a pathname.
5303  Change Number: XSH/TC2/D4/0360 [817]
5304  On Page: 2057 Line: 65053 Section: symlink()
5306  In the DESCRIPTION section, change from:
5307  If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches
5308  are permitted using the current permissions of the directory underlying the file descriptor. If the file
5309  descriptor was opened with O_SEARCH, the function shall not perform the check.
5310  to:
5311  If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the
5312  function shall check whether directory searches are permitted using the current permissions of the directory
5313  underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.
5315  Change Number: XSH/TC2/D4/0361 [822]
5316  On Page: 2058 Line: 65096 Section: symlink()
5317  In the RETURN VALUE section, add after the [ENOENT] error:
5318  [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.


Change Number: XSH/TC2/D4/0362 [817]

On Page: 2058 Line: 65105 Section: symlink()


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 ...


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.


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).


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.

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:

```c
#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. */
    /* 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. */
```
node = tsearch((void *)elementptr, &root, node_compare);
if (node == NULL) {
    fprintf(stderr,
        "tsearch: Not enough space available\n\n");
    exit(EXIT_FAILURE);
}
else if (*((struct element **)node) != elementptr) { /* A node containing the element already exists */
    (*((struct element **)node)->count)++;
    free(elementptr);
}

if (node != NULL) {
    fprintf(stderr,
        "tsearch: Not enough space available\n\n");
    exit(EXIT_FAILURE);
}
else if (*((struct element **)node) != elementptr) { /* A node containing the element already exists */
    (*((struct element **)node)->count)++;
    free(elementptr);
}

for each node in the tree
    if (node != NULL) {
        elementptr = *((struct element **)root);
        printf("deleting node: string = %s, count = %d\n", elementptr->string, elementptr->count);
        tdelete((void *)elementptr, &root, delete_root);
        free(elementptr);
    }
}

return 0;

/* Delete all nodes in the tree */
while (root != NULL) {
    elementptr = *((struct element **)root);
    printf("deleting node: string = %s, count = %d\n", elementptr->string, elementptr->count);
    tdelete((void *)elementptr, &root, delete_root);
    free(elementptr);
}

return 0;

/* This routine compares two nodes, based on an alphabetical ordering of the string field. */
int node_compare(const void *node1, const void *node2) {
    return strcmp(((const struct element *) node1)->string, ((const struct element *) node2)->string);
}

/* This comparison routine can be used with tdelete() when explicitly deleting a root node, as no comparison is necessary. */
int delete_root(const void *node1, const void *node2) {
    return 0;
}

/* This routine prints out a node, the second time twalk encounters it or if it is a leaf. */
void print_node(const void *ptr, VISIT order, int level) {
    const struct element *p = *((const struct element **)ptr);
    if (order == postorder || order == leaf) {
        (void) printf("string = %s, count = %d\n", p->string, p->count);
    }
}
In the APPLICATION USAGE section, change from:

The `tsearch()` function uses

to:

The `twalk()` function uses


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.


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 \( 0 < x < 1/\text{DBL\_MAX} \), and 171.7 < \( x \).

Change Number: XSH/TC2/D4/0369 [659]

On Page: 2113 Line: 66876 Section: timer_delete()

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

The behavior is undefined if the value specified by the timerid argument does not correspond to a timer ID returned by timer_create() but not yet deleted by timer_delete().

On Page: 2113 Line: 66881-66882 Section: timer_delete()

In the ERRORS section, replace the entire section with:

No errors are defined.

On Page: 2113 Line: 66888 Section: timer_delete()

In the RATIONALE section, change from:

None.

to:

If an implementation detects that the value specified by the timerid argument to timer_delete() 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.


Change Number: XSH/TC2/D4/0370 [659]

On Page: 2115 Line: 66956 Section: timer_getoverrun()

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

The behavior is undefined if 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().

On Page: 2115 Line: 66968-66970 Section: timer_getoverrun()

In the ERRORS section, delete:

These functions may fail if:

[EINVAL] The timerid argument does not correspond to an ID returned by timer_create() but not yet deleted by timer_delete().

On Page: 2116 Line: 66997 Section: timer_getoverrun()

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.


**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`.


**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`.


**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

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

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.

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.

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

Copyright © 2016 IEEE and The Open Group. All rights reserved.
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()  

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()  

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.


Change Number: XSH/TC2/D4/0378 [873]

On Page: 2154 Line: 68001 Section: unlink()  

In the NAME section, delete:

relative to directory file descriptor


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>

5602 Change Number: XSH/TC2/D4/0380 [817]
5603 On Page: 2154 Line: 68023 Section: unlink()
5605 In the DESCRIPTION section, change from:
5606 If the file descriptor was opened without O_SEARCH, the function shall check whether directory searches
5607 are permitted using the current permissions of the directory underlying the file descriptor. If the file
5608 descriptor was opened with O_SEARCH, the function shall not perform the check.
5609 to:
5610 If the access mode of the open file description associated with the file descriptor is not O_SEARCH, the
5611 function shall check whether directory searches are permitted using the current permissions of the directory
5612 underlying the file descriptor. If the access mode is O_SEARCH, the function shall not perform the check.
5614 Change Number: XSH/TC2/D4/0381 [817]
5615 On Page: 2155 Line: 68063 Section: unlink()
5617 In the ERRORS section, for the [EACCES] error, change from:
5618   fd was not opened with O_SEARCH and ...
5619 to:
5620 The access mode of the open file description associated with fd is not O_SEARCH and ...
5622 Change Number: XSH/TC2/D4/0382 [582]
5623 On Page: 2162 Line: 68286 Section: uselocale()
5624 Replace the DESCRIPTION, RETURN VALUE, and ERRORS sections with:
5625 DESCRIPTION
5626 The uselocale() function shall set or query the current locale for the calling thread.
5627 The value for the newloc argument shall be one of the following:
5628 1. A value returned by the newlocale() or duplocale() functions
5629 2. The special locale object descriptor LC_GLOBAL_LOCALE
5630 3. (locale_t)0
5631 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:

```c
[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](http://austingroupbugs.net/view.php?id=582).*

**Change Number: XSH/TC2/D4/0383 [873]**

- In the NAME section, delete:

  relative to directory file descriptor


**Change Number: XSH/TC2/D4/0384 [690]**

- 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.

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()

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]


Change Number: XSH/TC2/D4/0385 [691]

On Page: 2183 Line: 68748-68753,68755-68758 Section: wait()

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.

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]


Change Number: XSH/TC2/D4/0387 [690]

On Page: 2190 Line: 69058 Section: waitid()

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]


Change Number: XSH/TC2/D4/0388 [73]

On Page: 2207 Line: 69521 Section: wcsftime()

In the DESCRIPTION section, change from:

x The argument format is a wide-character string and the conversion specifications are replaced by
corresponding sequences of wide characters.

to:

x 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.


A clarification has been made in the C11 standard.

Change Number: XSH/TC2/D4/0389 [740]

On Page: 2207 Line: 69521 Section: wcsftime()

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]

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.


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.

On Page: 2224 Line: 69991 Section: wcstod()  5808
Change from:

In other than the C [CX]or POSIX[/CX] locale[/CX]s[/CX], other implementation-defined subject 5811
sequences may be  5812
to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be


Change Number: XSH/TC2/D4/0392 [796]

On Page: 2230 Line: 70196 Section: wcstol()  5816
In the DESCRIPTION section, change from:

minus-sign, the sequence shall be interpreted as negated.  5819
to:

<hyphen-minus>, the sequence shall be interpreted as negated.


Change Number: XSH/TC2/D4/0393 [584]

On Page: 2231 Line: 70199 Section: wcstoul()  5816
In other than the C [CX]or POSIX[/CX] locale[/CX]s[/CX], other implementation-defined subject 5827
sequences may be  5828
to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be


Change Number: XSH/TC2/D4/0394 [796]

On Page: 2237 Line: 70357 Section: wcstoul()  5833
Change from:

In other than the C [CX]or POSIX[/CX] locale[/CX]s[/CX], other implementation-defined subject 5827
sequences may be  5828
to:

In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be

In the DESCRIPTION section, change from:

5835  minus-sign, the sequence shall be interpreted as negated.

5836 to:

5837  <hyphen-minus>, the sequence shall be interpreted as negated.

5838  


5839 Change Number: XSH/TC2/D4/0396 [796]

5840 On Page: 2238 Line: 70360 Section: wcstoul()


5842 Change from:

5843  In other than the C [CX]or POSIX[/CX] locale[/CX], other implementation-defined subject
5844  sequences may be

5845 to:

5846  In other than the C [CX]or POSIX[/CX] locale, additional locale-specific subject sequence forms may be


5848 Change Number: XSH/TC2/D4/0397 [608]

5849 On Page: 2258 Line: 71006 Section: wordexp()

5850 In the DESCRIPTION section, change from:

5851  wordexp() may write messages to stderr if syntax errors are detected while expanding words.

5852 to:

5853  wordexp() may write messages to stderr if syntax errors are detected while expanding words, unless the
5854  stderr stream has wide orientation in which case the behavior is undefined.


5856 Change Number: XSH/TC2/D4/0398 [704]

5857 On Page: 2258 Line: 71009 Section: wordexp()


5859 In the DESCRIPTION section, add a new paragraph:

5860 The results are unspecified if WRDE_APPEND and WRDE_REUSE are both specified.

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()


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.


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.


Change Number: XSH/TC2/D4/0401 [676,710]

On Page: 2266 Line: 71250 Section: write()


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).


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()

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]).

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}`

Change Number: XCU/TC2/D4/0002 [584]

On Page: 2289 Line: 71989 Section: 1.4 Utility Description Defaults

Change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0003 [913]

On Page: 2293 Line: 72201 Section: 1.4 Utility Description Defaults

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.


Change Number: XCU/TC2/D4/0004 [705]

On Page: 2296 Line: 72310 Section: 1.6 Built-In Utilities

In Table 1-5 Regular Built-In Utilities, add hash, type and ulimit to the table.


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.)


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.


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

Delete rule 7:

7. If the current character is an unquoted <newline>, the current token shall be delimited.


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.

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 ...


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.


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.

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.


Change Number: XCU/TC2/D4/0013 [888]

On Page: 2306 Line: 72709 Section: 2.6.2 Parameter Expansion

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.


Change Number: XCU/TC2/D4/0014 [867]

On Page: 2306 Line: 72716 Section: 2.6.2 Parameter Expansion

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

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}
```

If the shell variable \textit{x} contains a value that forms a valid integer constant, then ...

to:

If the shell variable \textit{x} contains a value that forms a valid integer constant, optionally including a leading <plus-sign> or <hyphen-minus>, then ...


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>

The shell shall treat each character of the \textit{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 \textit{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.

\textit{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.

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

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()]).


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 \( \text{word} \)

to:

If any part of \( \text{word} \)

On line 72985 change from:

If no characters in \( \text{word} \) are

to:

If no part of \( \text{word} \) is


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


Change Number: XCU/TC2/D4/0020 [882]

On Page: 2315 Line: 73032-73042 Section: 2.8.1 Consequences of Shell Errors

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:

<table>
<thead>
<tr>
<th>Error</th>
<th>Special Built-In</th>
<th>Other Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell language syntax error</td>
<td>Shall exit</td>
<td>Shall exit</td>
</tr>
<tr>
<td>Utility syntax error (option or operand error)</td>
<td>Shall exit</td>
<td>Shall not exit</td>
</tr>
<tr>
<td>Redirection error</td>
<td>Shall exit</td>
<td>Shall not exit</td>
</tr>
<tr>
<td>Variable assignment error</td>
<td>Shall exit</td>
<td>Shall not exit</td>
</tr>
<tr>
<td>Expansion error</td>
<td>Shall exit</td>
<td>Shall exit</td>
</tr>
<tr>
<td>Command not found</td>
<td>N/A</td>
<td>May exit</td>
</tr>
<tr>
<td>Dot script not found</td>
<td>Shall exit</td>
<td>N/A</td>
</tr>
</tbody>
</table>

to:
Certain errors shall cause the shell to write a diagnostic message to standard error and exit as shown in the following table:

<table>
<thead>
<tr>
<th>Error</th>
<th>Non-Interactive Shell</th>
<th>Interactive Shell</th>
<th>Shell Diagnostic Message Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell language syntax error</td>
<td>shall exit</td>
<td>shall not exit</td>
<td>yes</td>
</tr>
<tr>
<td>Special built-in utility error</td>
<td>shall exit</td>
<td>shall not exit</td>
<td>no(^1)</td>
</tr>
<tr>
<td>Other utility (not a special built-in) error</td>
<td>shall not exit</td>
<td>shall not exit</td>
<td>no(^2)</td>
</tr>
</tbody>
</table>
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.


Change Number: XCU/TC2/D4/0021 [717,882]

On Page: 2315 Line: 73047-73051 Section: 2.8.1 Consequences of Shell Errors

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.


**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.


**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).


**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:

If no command name results, variable assignments shall affect the current execution environment.
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.

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

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
- compdescribes
- compfiles
- compgen
- compgroups
- complete
- compquote
- comptags
- compytry
- 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.
Change Number: XCU/TC2/D4/0026 [705]

On Page: 2317 Line: 73130 Section: 2.9.1.1 Command Search and Execution

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.


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.


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.


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:  

to:

case word in

[[pattern1] compound-list；;
[If[patern[ pattern] ... ) compound-list；:] ... 
[If[patern[ pattern] ... ) compound-list]
esac

to:

case word in

[[ pattern1 ] compound-list ；;
[If[patern[ pattern] ... ) compound-list ；:] ... 
[If[patern[ pattern] ... ) compound-list]
esac

On Page: 2324 Line: 73379 Section: 2.9.5 Function Definition Command

Change from:

fname() compound-command[io-redirect ...]
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.


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 \lnewline\ shall be returned as the token identifier NEWLINE.


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


Change Number: XCU/TC2/D4/0033 [643,839]

On Page: 2326 Line: 73475-73481 Section: 2.10.2 Shell Grammar Rules

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.


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 ...

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 */


In the shell grammar, change from:

%start complete_command
%
complete_command : list separator
| list

; to:

%start program
%
program          : linebreak complete_commands linebreak
| linebreak
| ;
| linebreak
| ;
complete_commands: complete_commands newline_list complete_command
| complete_command
| ;
complete_command : list separator_op
| list
| ;


In the shell grammar, change from:

compound_list    :              term
| newline_list term
|              term separator

215
Copyright © 2016 IEEE and The Open Group. All rights reserved.
<table>
<thead>
<tr>
<th>newline_list term separator</th>
</tr>
</thead>
<tbody>
<tr>
<td>;</td>
</tr>
</tbody>
</table>

to:

compound_list : linebreak term|
linebreak term separator |
;                          


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
            ;
```


Change Number: XCU/TC2/D4/0039 [735]

On Page: 2328 Line: 73583 Section: 2.10.2 Shell Grammar Rules


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
                ;
```

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 ...


Change from:

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 ...


Change from:

The escaping <backslash> shall be discarded.


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

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.

&lt;small&gt;Note: a future version of this standard may require the byte-by-byte further comparison described above.&lt;/small&gt;


Change Number: XCU/TC2/D4/0044 [882]

On Page: 2334 Line: 73821 Section: 2.14 Special Built-In Utilities

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.


Change Number: XCU/TC2/D4/0045 [654]

On Page: 2334 Line: 73826 Section: 2.14 Special Built-In Utilities

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.
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.


Change Number: XCU/TC2/D4/0046 [842]


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.

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.)

```c
foo() {

}```
6605 for j in 1 2; do
6606     echo 'break 2' >/tmp/do_break
6607     echo " sourcing /tmp/do_break ($j)..."
6608     # the behavior of the break from running the following command
6609     # results in unspecified behavior:
6610     . /tmp/do_break
6611     do_continue() { continue 2; }
6612     echo " running do_continue ($j)..."
6613     # the behavior of the continue in the following function call
6614     # results in unspecified behavior if execution reaches this
6615     # point):
6616     do_continue
6617
6618     trap 'continue 2' USR1
6619     echo " sending SIGUSR1 to self ($$$)"
6620     # the behavior of the continue in the trap invoked from the
6621     # following signal results in unspecified behavior (if
6622     # execution reaches this point):
6623     kill -USR1 $$
6624     sleep 1
6625     done
6626 done
6627
6628 for i in 1 2; do
6629     echo "running foo ($i)..."
6630     foo
6631     done
6632

6633 Change the DESCRIPTION section from:

6634 The \texttt{continue} utility shall return to the top of the smallest enclosing \texttt{for}, \texttt{while}, or \texttt{until} loop, or to the top
6635 of the \texttt{n}th enclosing loop, if \texttt{n} is specified. This involves repeating the condition list of a \texttt{while} or \texttt{until} loop
6636 or performing the next assignment of a \texttt{for} loop, and re-executing the loop if appropriate.

6637 The value of \texttt{n} is a decimal integer greater than or equal to 1. The default shall be equivalent to \texttt{n=1}. If \texttt{n} is
6638 greater than the number of enclosing loops, the outermost enclosing loop shall be used.

6639
6640 to:

6641 If \texttt{n} is specified, the \texttt{continue} utility shall return to the top of the \texttt{n}th enclosing \texttt{for}, \texttt{while}, or \texttt{until} loop. If \texttt{n}
6642 is not specified, \texttt{continue} shall behave as if \texttt{n} was specified as 1. Returning to the top of the loop involves
6643 repeating the condition list of a \texttt{while} or \texttt{until} loop or performing the next assignment of a \texttt{for} loop, and re-
6644 executing the loop if appropriate. The value of \texttt{n} is a positive decimal integer. If \texttt{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 \texttt{break} utility.

6648 \textit{Rationale}: Austin Group Defect Report(s) applied: 842. See \url{http://austingroupbugs.net/view.php?id=842}.
6650  **Change Number: XCU/TC2/D4/0047 [717]**

6651  On Page: 2347 Line: 74214 Section: 2.14 exit
6653  In the DESCRIPTION section, change from:
6654  The `exit` utility shall cause the shell to exit with the exit status specified by the unsigned decimal integer \( n \).
6655  to:
6656  The `exit` utility shall cause the shell to exit from its current execution environment with the exit status
6657  specified by the unsigned decimal integer \( n \). If the current execution environment is a subshell
6658  environment, the shell shall exit from the subshell environment with the specified exit status and continue
6659  in the environment from which that subshell environment was invoked, otherwise the shell utility shall
6660  terminate with the specified exit status.
6662  The changes made here make the standard reflect existing practice.

6663  **Change Number: XCU/TC2/D4/0048 [960]**

6664  On Page: 2347 Line: 74240 Section: 2.14 exit
6666  In the EXIT STATUS section, change from:
6667  The exit status shall be \( n \), if specified.
6668  to:
6669  The exit status shall be \( n \), if specified, except that the behavior is unspecified if \( n \) is not an unsigned
6670  decimal integer or is greater than 255.

6672  **Change Number: XCU/TC2/D4/0049 [717]**

6673  On Page: 2348 Line: 74252 Section: 2.14 exit
6675  In the EXAMPLES section, add at the end of the section:
6676  Propagate error handling from within a subshell:
6677
6678  (  
6679    command1 || exit 1
6680    command2 || exit 1
6681    exec command3
6682  ) > outputfile || exit 1
6683  echo "outputfile created successfully"

Change Number: XCU/TC2/D4/0050 [960]

On Page: 2348 Line: 74258 Section: 2.14 exit


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.


Change Number: XCU/TC2/D4/0051 [654]

On Page: 2349 Line: 74290 Section: 2.14 export


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.


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


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.
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

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.
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>


Change Number: XCU/TC2/D4/0054 [717]

On Page: 2358 Line: 74545 Section: 2.14 set

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.


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
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).


This is a layered change on XCU/TC1/D5/0047 [155,280]. This change replaces the previous revised text for the -u option.

In the EXIT STATUS section, change from:

Zero.

to:

0 Successful completion.

>0 An invalid option was specified, or an error occurred.


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
6800 errors.

6801 Change Number: XCU/TC2/D4/0057 [584]

6802 On Page: 2415 Line: 76517, 76518 Section: ar

6803 In the RATIONALE section, change from:

6804 <hyphen>

6805 to:

6806 <hyphen-minus>


6808 Change Number: XCU/TC2/D4/0058 [584]

6809 On Page: 2434 Line: 77280 Section: awk

6810 In the EXTENDED DESCRIPTION section, change from:

6811 sequence begins with a minus-sign,

6812 to:

6813 sequence begins with a <hyphen-minus>,


6815 Change Number: XCU/TC2/D4/0059 [963]

6816 On Page: 2437 Line: 77382 Section: awk


6818 In the EXTENDED DESCRIPTION section, change from:

6819 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.

6820 to:

6821 operands shall be converted to strings as required and a string comparison shall be made as follows:

6822  

6823 • 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.

6824 • For the other operators, the strings shall be compared using the locale-specific collation sequence.

6825 The value of the comparison expression shall be 1 if the relation is true, or 0 if the relation is false.

Change Number: XCU/TC2/D4/0060 [226]

On Page: 2447 Line: 77855 Section: awk


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


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
  +   +   ;
  +   +   +
```

Copyright © 2016 IEEE and The Open Group. All rights reserved.
action : '{' newline_opt }'
| '{' newline_opt terminated_statement_list '}'
| '{' newline_opt unterminated_statement_list '}'
| ;

-terminator : terminator ';
- terminator : terminator NEWLINE
+terminator : terminator NEWLINE
| ';
| NEWLINE
| ;

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,

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]

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.


Change Number: XCU/TC2/D4/0062 [963]

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

Change Number: XCU/TC2/D4/0063 [226]

On Page: 2458 Line: 78313 Section: awk

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.


Change Number: XCU/TC2/D4/0064 [963]

On Page: 2463 Line: 78550 Section: awk

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).


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(),

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>


Change Number: XCU/TC2/D4/0067 [679]

On Page: 2480 Line: 79259 Section: bc

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.


Change Number: XCU/TC2/D4/0068 [650]


In the OPTIONS section, in the -E description change from:

expanding all preprocessor directives

to:

executing all preprocessor directives


Change Number: XCU/TC2/D4/0069 [670]


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 (-1 rt), change from:

interfaces beginning with the prefix clock_ and time_

to:

interfaces beginning with the prefix clock_ and timer_


Change Number: XCU/TC2/D4/0070 [638]


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


Change Number: XCU/TC2/D4/0071 [650]


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).


Change Number: XCU/TC2/D4/0072 [784]

On Page: 2497 Line: 79930 Section: c99


In the FUTURE DIRECTIONS section, change from:

None.
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.


Change Number: XCU/TC2/D4/0073 [876]

On Page: 2502 Line: 80083 Section: cat

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

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


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>


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:

```c
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" ;
```
esac
    return $status
}

On Page: 2534 Line: 81336 Section: cmp

In the FUTURE DIRECTIONS section, change from:

None.

to:

Future versions of this standard may require that diagnostic messages are written to standard error when the
-s option is specified.

Existing implementations have adopted both possible interpretations of the current wording.

Change Number: XCU/TC2/D4/0076 [963]

On Page: 2535 Line: 81354 Section: comm

In the DESCRIPTION section, add a new paragraph:

If the collating sequence of the current locale does not have a total ordering of all characters (see [xref to
XBD 7.3.2]) and any lines from the input files collate equally but are not identical, comm should treat them
different lines but may treat them as being the same. If it treats them as different, comm should expect
them to be ordered according to a further byte-by-byte comparison using the collating sequence for the
POSIX locale and if they are not ordered in this way, the output of comm can identify such lines as being
both unique to file1 and unique to file2 instead of being in both files.

On Page: 2536 Line: 81409 Section: comm
(2013 edition 2560 Line: 82810)

In the STDOUT section, change from:

If the input files were ordered according to the collating sequence of the current locale, the lines written
shall be in the collating sequence of the original lines.

to:

If the input files were ordered according to the collating sequence of the current locale, the lines written
shall be in the collating sequence of the current locale. If the input files contained any lines that collated
equally but were not identical and within each file those lines were ordered according to a further byte-by-
byte comparison using the collating sequence for the POSIX locale, and comm treated them as different
lines, then lines written that collate equally but are not identical should be ordered according to a further
byte-by-byte comparison using the collating sequence for the POSIX locale.

In the APPLICATION USAGE section, add a new paragraph:

When using `comm` 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.


In the APPLICATION USAGE section, add the following new paragraphs:

If the collating sequence of the current locale does not have a total ordering of all characters, this can affect the behaviour of `comm` in the following ways:

- If `comm` treats lines as being the same only if they are identical, some lines can be misleadingly identified as being both unique to `file1` and unique to `file2`.
- If `comm` treats lines as being the same if they collate equally and a line from `file1` collates equally with a line from `file2` but is not identical to it, one of the lines is misleadingly identified as being in both files and the other is not written to the output at all.

Such problems can be avoided by forcing the use of the POSIX locale, for example the following identifies lines in both `file1` and `file2`:

```sh
LC_ALL=POSIX sort file1 > file1.posix
LC_ALL=POSIX sort file2 > file2.posix
LC_ALL=POSIX comm -12 file1.posix file2.posix | sort
```

The final `sort` re-sorts the output of `comm` according to the collating sequence of the original locale. Doing this might be difficult if more than one column is output and leading blanks cannot be ignored.

In the FUTURE DIRECTIONS section, change from:

None.

A future version of this standard may require that if any lines from the input files collate equally but are not identical, then `comm` treats them as different lines and expects them to be ordered according to a further byte-by-byte comparison using the collating sequence for the POSIX locale.
A future version of this standard may require that if the input files contained any lines that collated equally but were not identical and within each file those lines were ordered according to a further byte-by-byte comparison using the collating sequence for the POSIX locale, then lines written that collate equally but are not identical are ordered according to a further byte-by-byte comparison using the collating sequence for the POSIX locale.


Change Number: XCU/TC2/D4/0079 [584]

On Page: 2556 Line: 82163 Section: crontab

In the INPUT FILES section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0080 [584]

On Page: 2568 Line: 82614, 82616, 82617, 82634 Section: cut

In the OPTIONS section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0081 [907]

On Page: 2583 Line: 83203 Section: dd


In the OPERANDS section, change from:

count=n  Copy only \textit{n} input blocks.

to:

\texttt{count=n}  Copy only \textit{n} input blocks. If \textit{n} is zero, it is unspecified whether no blocks or all blocks are copied.

There are two existing implementation behaviors and the standard only allows one of them.

**Change Number:** XCU/TC2/D4/0082 [584]

On Page: 2601 Line: 83825 Section: diff

In the STDOUT 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/D4/0083 [950]

On Page: 2603 Line: 83888,83889 Section: diff


In the STDOUT section, change from:

```
---VVVVVVLOH
...  
VVVVVVLOH
...  
```

to:

```
---V\WVV

", file1,
...
VVVVVVLOH
...
```

**Rationale:** Austin Group Defect Report(s) applied: 950. See http://austingroupbugs.net/view.php?id=950.

**Change Number:** XCU/TC2/D4/0084 [969]

On Page: 2604 Line: 83911 Section: diff


In the STDOUT section, change from:

```
Each <range> field shall be of the form:

"%1d", <beginning line number>
```

if the range contains exactly one line
to:

```
Each <range> field shall be of the form:

"%1d", <beginning line number>
```
or:

"%ld,1", <beginning line number>

if the range contains exactly one line

The standard only allows one form for specifying single line changes in unified diff output, but existing practice uses two forms.

Change Number: XCU/TC2/D4/0085 [929]

On Page: 2604 Line: 83929 Section: diff
In the STDOUT section, change from:
shall contain no more than \( n \) consecutive unaffected lines
to:
shall contain no more than \( 2n \) consecutive unaffected lines


Change Number: XCU/TC2/D4/0086 [612]

Change the table in the EXAMPLES section to:
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.


Change Number: XCU/TC2/D4/0087 [620]

On Page: 2609 Line: 84137-84139 Section: dirname
In the RATIONALE section, delete:
The dirname utility originated in System III. It has evolved through the System V releases to a version that matches the requirements specified in this description in System V Release 3. 4.3 BSD and earlier versions did not include dirname.

Change Number: XCU/TC2/D4/0088 [612]

On Page: 2610 Line: 84154 Section: dirname

In the SEE ALSO section, add a new paragraph:

XSH dirname()


Change Number: XCU/TC2/D4/0089 [527]

On Page: 2611 Line: 84170 Section: du

In the DESCRIPTION section, change from:

Files with multiple links shall be counted and written for only one entry.

to:

A file that occurs multiple times under one file operand and that has a link count greater than 1 shall be
counted and written for only one entry. It is implementation-defined whether a file that has a link count no
greater than 1 is counted and written just once, or is counted and written for each occurrence. It is
implementation-defined whether a file that occurs under one file operand is counted for other file operands.

On Page: 2611 Line: 84177 Section: du

In the OPTIONS section, for the -a option change from:

Regardless of the presence of the -a option, non-directories given as file operands shall always be listed.

to:

The -a option shall not affect whether non-directories given as file operands are listed.


In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of this standard may require that a file that occurs multiple times shall be counted and
written for only one entry, even if the occurrences are under different file operands.


Existing practice is varied in how du counts files.
Change Number: XCU/TC2/D4/0090 [584]

On Page: 2620 Line: 84532,84533 Section: ed

In the EXTENDED DESCRIPTION section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0091 [584]

On Page: 2621 Line: 84537,84539 Section: ed

In the EXTENDED DESCRIPTION section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0092 [584]

On Page: 2632 Line: 85026 Section: ed

In the RATIONALE section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0093 [584]

On Page: 2645 Line: 85512 Section: ex

In the EXTENDED DESCRIPTION section, change from:

A <plus-sign> ('+') or a minus-sign ('-')
A <plus-sign> (`+`) or a <hyphen-minus> (`-`)


**Change Number:** XCU/TC2/D4/0094 [942]

On Page: 2715 Line: 88231 Section: expr

In the EXTENDED DESCRIPTION section, change from:

The expressions are listed in order of increasing precedence

to:

The expressions are listed in order of decreasing precedence

On Page: 2715 Line: 88234-88264 Section: expr

In the EXTENDED DESCRIPTION section, rearrange the lines of the table into the following order (still with each group of lines forming a row of the table):

<table>
<thead>
<tr>
<th>Table Header</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer, string</td>
<td>88234 (2013 edition line 89640)</td>
</tr>
<tr>
<td>parentheses</td>
<td>88259-88261 (2013 edition line 89665-89667)</td>
</tr>
<tr>
<td>,</td>
<td>88258 (2013 edition line 89664)</td>
</tr>
<tr>
<td>*, /, %</td>
<td>88253-88257 (2013 edition line 98659-89663)</td>
</tr>
<tr>
<td>+, -</td>
<td>88251-88252 (2013 edition line 89657-89658)</td>
</tr>
<tr>
<td>=, &gt;, &gt;=, &lt;, &lt;=, !=</td>
<td>88240-88250 (2013 edition line 89646-89656)</td>
</tr>
<tr>
<td>&amp;</td>
<td>88238-88239 (2013 edition line 89644-89645)</td>
</tr>
<tr>
<td></td>
<td>88235-88237 (2013 edition line 89641-89643)</td>
</tr>
</tbody>
</table>

In the EXTENDED DESCRIPTION section, change from:

A string argument is an argument that cannot be identified as an integer argument or as one of the expression operator symbols shown in the OPERANDS section.

to:

Identification as Integer or String

An argument or the value of a subexpression that consists only of an optional unary minus followed by digits is a candidate for treatment as an integer if it is used as the left argument to the | operator or as either argument to any of the following operators: & > >= < <= != + - * / %. Otherwise, the argument or subexpression value shall be treated as a string.


In the APPLICATION USAGE section, change from:

expr $a =('='

to:

expr "$a" = '='

In the APPLICATION USAGE section, add a new paragraph:

For testing string equality the `test` utility is preferred over `expr`, as it is usually implemented as a shell built-in. However, the functionality is not quite the same because the `expr '='` and `!='` operators check whether strings collate equally, whereas `test` checks whether they are identical. Therefore, they can produce different results in locales where the collation sequence does not have a total ordering of all characters (see [xref to XBD 7.3.2]).


---

On Page: 2717 Line: 88304 Section: expr

Move lines 88304-88316 (2013 edition lines 89710-89722) from the EXAMPLES section to the start of the APPLICATION USAGE section before line 88289 (2013 edition line 89695).

Change Number: XCU/TC2/D4/0099 [584]

On Page: 2738 Line: 89085 Section: find

In the OPERANDS section, change from:

preceded by a plus ('+') or minus-sign ('-') sign

to:

preceded by a &lt;plus-sign&gt; ('+') or &lt;hyphen-minus&gt; ('-')


Change Number: XCU/TC2/D4/0100 [584]

On Page: 2738 Line: 89122, 89124, 89126 Section: find

In the OPERANDS section, change from:

&lt;hyphen&gt;

to:

&lt;hyphen-minus&gt;


Change Number: XCU/TC2/D4/0101 [584]

On Page: 2739 Line: 89129, 89131 Section: find

In the OPERANDS section, change from:

&lt;hyphen&gt;

to:

&lt;hyphen-minus&gt;


Change Number: XCU/TC2/D4/0102 [546]

On Page: 2751 Line: 89574 Section: fort77

In the SYNOPSIS section, change the shading from FD to OB FD.

In the FUTURE DIRECTIONS section, replace the existing text with:

Future versions of this standard may withdraw this utility. There are implementations of compilers that conform to much more recent versions of the Fortran programming language. Since there is no active Fortran binding to POSIX-2008, this standard does not need to specify any compiler.


In the OUTPUT FILES section, change from:

<hyphen>

to:

<hyphen-minus>


In the APPLICATION USAGE section, add a new paragraph:

When using grep 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.
In the APPLICATION USAGE section, change from:

None.

to:

When using *head* 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.


In the STDOUT section, change from:

a minus-sign (`-`).

to:

a <hyphen-minus> (`-`).


In the DESCRIPTION section, change from:

that have identical join fields

to:

that have join fields that collate equally

In the APPLICATION USAGE section, add a new paragraph:
If the collating sequence of the current locale does not have a total ordering of all characters (see [xref to XBD 7.3.2]), `join` treats fields that collate equally but are not identical as being the same. If this behavior is not desired, it can be avoided by forcing the use of the POSIX locale (although this means re-sorting the input files into the POSIX locale collating sequence.)


**Change Number:** XCU/TC2/D4/0110 [663]

On Page: 2817 Line: 91965 Section: join

In the APPLICATION USAGE section, add a new paragraph:

When using `join` 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.


**Change Number:** XCU/TC2/D4/0111 [971]

On Page: 2817 Line: 91980 Section: join

In the EXAMPLES section, change from:

would produce

to:

(where `<tab>` is a literal `<tab>` character) would produce


**Change Number:** XCU/TC2/D4/0112 [885]

On Page: 2818 Line: 92018 Section: join

In the RATIONALE section, change from:

objections that the `join` in the base documents does not support

to:

objections that the `join` in the base documents for POSIX.2-1992 did not support
In the RATIONALE section, change from:

base documents

to:

base documents for POSIX.2-1992


Change Number: XCU/TC2/D4/0113 [930]

In the DESCRIPTION section, for list items 1.b and 1.c change from:

destination

to:

the destination path


Change Number: XCU/TC2/D4/0114 [941]

In the EXAMPLES section, change from:

if printf "\n "$response" | grep -Eq "$(locale yesexpr)"

to:

printf 'Prompt for response: '
read response
if printf "\n "$response" | grep -- "$$(locale yesexpr)"


Change Number: XCU/TC2/D4/0115 [963]

In the DESCRIPTION section, add a new paragraph:
Whenever `ls` sorts filenames or pathnames according to the collating sequence in the current locale, if this collating sequence does not have a total ordering of all characters (see [xref to XBD 7.3.2]), then any filenames or pathnames that collate equally should be further compared byte-by-byte using the collating sequence for the POSIX locale.


Change Number: XCU/TC2/D4/0116 [963]

On Page: 2872 Line: 94057 Section: ls

In the FUTURE DIRECTIONS section, change from:

None.

to:

Allowing `-f` to ignore the `-A`, `-g`, `-l`, `-n`, `-o`, and `-s` options may be removed in a future version.

A future version of this standard may require that if the collating sequence for the current locale does not have a total ordering of all characters, any filenames or pathnames that collate equally are further compared byte-by-byte using the collating sequence for the POSIX locale.


This is a layered change on XCU/TC1/D5/0116 [424]. The change over the 2013 edition is the addition of the second paragraph.

Change Number: XCU/TC2/D4/0117 [964]

On Page: 2877 Line: 94286 Section: m4

In the EXTENDED DESCRIPTION section, for m4exit change from:

Exit from the `m4` utility. If the first argument is specified, it is the exit code. The default is zero. It shall be an error to specify an argument containing any non-numeric characters.

to:

Exit from the `m4` utility. If the first argument is specified, it shall be the exit code. If no argument is specified, the exit code shall be zero. It shall be an error to specify an argument containing any non-numeric characters. If the first argument is zero or no argument is specified, and an error has previously occurred (for example, a file operand that could not be opened), it is unspecified whether the exit status is zero or non-zero.

In the EXTENDED DESCRIPTION section, for mkstemp change from:

The first argument shall be taken as a template for creating an empty file, with trailing ‘X’ characters replaced with characters from the portable filename character set. The behavior is unspecified if the first argument does not end in at least six ‘X’ characters. If a temporary file is successfully created, then the defining text of the macro shall be the name of the new file. The user ID of the file shall be set to the effective user ID of the process. The group ID of the file shall be set to the group ID of the file's parent directory or to the effective group ID of the process. The file access permission bits are set such that only the owner can both read and write the file, regardless of the current umask of the process.

to:

The defining text shall be as if it were the resulting pathname after a successful call to the mkstemp() function defined in the System Interfaces volume of POSIX.1-2008 called with the first argument to the macro invocation. If a file is created, that file shall be closed.

The current standard does not match any existing practice. The m4 utility uses the mkstemp() function and does not modify the umask nor call chmod() after creating the file.

Change Number: XCU/TC2/D4/0119 [964]

In the EXTENDED DESCRIPTION section, for mkstemp change from:

If a file could not be created, the defining text of the macro shall be the empty string.

to:

If a file could not be created, the m4 utility shall write a diagnostic message to standard error and shall continue processing input but its final exit status shall be non-zero; the defining text of the macro shall be the empty string.


Change Number: XCU/TC2/D4/0120 [855]
Determine the pathname of the user start-up file.

On Page: 2889 Line: 94801 Section: mailx

In the EXTENDED DESCRIPTION section, change from:

5. Process the start-up file of *mailx* commands named in the user *MAILRC* variable.

to:

5. Process the user start-up file of *mailx* commands named in the *MAILRC* variable.

On Page: 2889 Line: 94803 Section: mailx

In the EXTENDED DESCRIPTION section, change from:

The following commands shall be invalid in the start-up file: *!*, *edit*, *hold*, *mail*, *preserve*, *reply*, *Reply*, *shell*, *visual*, *Copy*, *followup*, and *Followup*. Any errors in the start-up file shall either cause *mailx* to terminate with a diagnostic message and a non-zero status or to continue after writing a diagnostic message, ignoring the remainder of the lines in the start-up file.

to:

The following commands shall be invalid in a start-up file: *!*, *edit*, *hold*, *mail*, *preserve*, *reply*, *Reply*, *shell*, *visual*, *Copy*, *followup*, and *Followup*. Any errors in a start-up file shall either cause *mailx* to terminate with a diagnostic message and a non-zero status or to continue after writing a diagnostic message, ignoring the remainder of the lines in the file.


**Change Number**: XCU/TC2/D4/0121 [619]

On Page: 2904 Line: 95360-95362 Section: mailx

In the RATIONALE section, change from:

Implementations are encouraged to conform to the various delivery mechanisms described in the CCITT X.400 standards or to the equivalent Internet standards, described in Internet Request for Comment (RFC) documents RFC 819, RFC 822, RFC 920, RFC 921, and RFC 1123.

to:

Implementations are encouraged to conform to the various delivery mechanisms described in the CCITT X.400 standards or to the equivalent Internet standards, described in Internet Request for Comment (RFC) documents RFC 819, RFC 920, RFC 921, RFC 1123, and RFC 5322 (which succeeded RFC 822).

Change Number: XCU/TC2/D4/0122 [509]

On Page: 2909 Line: 95561-95562 Section: make

In the INPUT FILES section, change from:

and comments

to:

include lines, and comments


Change Number: XCU/TC2/D4/0123 [584]

On Page: 2910 Line: 95580, 95583 Section: make

In the ENVIRONMENT VARIABLES section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0124 [857]

On Page: 2911 Line: 95629 Section: make


In the EXTENDED DESCRIPTION section, change from:

A target is considered out-of-date if it is older than any of its prerequisites or if it does not exist.

to:

A target shall be considered up-to-date if it exists and is newer than all of its dependencies, or if it has
already been made up-to-date by the current invocation of make (regardless of the target's existence or age).
A target may also be considered up-to-date if it exists, is the same age as one or more of its prerequisites,
and is newer than the remaining prerequisites (if any).

On Page: 2911 Line: 95634 Section: make


In the EXTENDED DESCRIPTION section, change from:

After make has ensured that all of the prerequisites of a target are up-to-date and if the target is out-of-date,
the commands associated with the target entry shall be executed. If there are no commands listed for the
target, the target shall be treated as up-to-date.

to:

To ensure that a target is up-to-date, make shall ensure that all of the prerequisites of a target are up-to-date, then check to see if the target itself is up-to-date. If the target is not up-to-date, the target shall be made up-to-date by executing the rule's commands (if any). If the target does not exist after the target has been successfully made up-to-date, the target shall be treated as being newer than any target for which it is a prerequisite.

If a target exists and there is neither a target rule nor an inference rule for the target, the target shall be considered up-to-date. It shall be an error if make attempts to ensure that a target is up-to-date but the target does not exist and there is neither a target rule nor an inference rule for the target.


Change Number: XCU/TC2/D4/0125 [505]

On Page: 2911 Line: 95644,95656 Section: make

In the EXTENDED DESCRIPTION section, change from:

Comments start with a <number-sign> ('#') and continue until an unescaped <newline> is reached.

to:

There are three kinds of comments: blank lines, empty lines, and a <number-sign> ('#') and all following characters up to the first unescaped <newline> character. Blank lines, empty lines, and lines with <number-sign> ('#') as the first character on the line are also known as comment lines.

On line 95656 change from:

empty lines, and comments

to:

and comments

On Page: 2912 Line: 95671 Section: make

In the EXTENDED DESCRIPTION section, change from:

The trailing <newline> and any comment shall be discarded.

to:

The trailing <newline>, any <blank> characters immediately preceeding a comment, and any comment shall be discarded.


The standard as currently written does not allow blank lines and empty lines in places where all known
implementations have allowed them to appear. These changes more accurately reflect existing practice.

Change Number: XCU/TC2/D4/0126 [584]

On Page: 2912 Line: 95691, 95694 Section: make

In the EXTENDED DESCRIPTION section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0127 [505]

On Page: 2913 Line: 95722 Section: make

In the EXTENDED DESCRIPTION section, change from:

An empty or blank line, or a line beginning with ‘#’, may begin a new entry.

to:

Any comment line may begin a new entry.


The standard as currently written does not allow blank lines and empty lines in places where all known implementations have allowed them to appear. These changes more accurately reflect existing practice.

Change Number: XCU/TC2/D4/0128 [865]

On Page: 2915 Line: 95807 Section: make

In the EXTENDED DESCRIPTION section, change from:

If string1 in a macro expansion contains a macro expansion, the results are unspecified.

to:

If string1 in a macro expansion contains a macro expansion, the results are unspecified. If a <percent-sign> character appears as part of subst1 or subst2 after any macros have been recursively expanded, the results are unspecified.


Long standing existing practice is to treat a % in macro substitution as special (to perform pattern expansions).
Change Number: XCU/TC2/D4/0129 [693]

On Page: 2916 Line: 95840 Section: make

In the EXTENDED DESCRIPTION section, after the existing paragraph:

Before the makefile(s) are read, all of the `make` utility command line macro definitions (except the `MAKEFLAGS` macro or the `SHELL` macro) shall be added to the environment of `make`. Other implementation-defined variables may also be added to the environment of `make`.

append:

Macros defined by the `MAKEFLAGS` environment variable and macros defined in the makefile(s) shall not be added to the environment of `make` if they are not already in its environment. With the exception of `SHELL` (see below), it is unspecified whether macros defined in these ways update the value of an environment variable that already exists in the environment of `make`.


Change Number: XCU/TC2/D4/0130 [602]

On Page: 2917 Line: 95916 Section: make

In the EXTENDED DESCRIPTION section, change from:

Its prerequisites in turn shall be processed recursively until a target is found that has no prerequisites, at which point the recursion stops.

to:

Its prerequisites in turn shall be processed recursively until a target is found that has no prerequisites, or further recursion would require applying two inference rules one immediately after the other, at which point the recursion shall stop. As an extension, implementations may continue recursion when two or more successive inference rules need to be applied; however, if there are multiple different chains of such rules that could be used to create the target, it is unspecified which chain is used.


Change Number: XCU/TC2/D4/0131 [848]

On Page: 2919 Line: 95986 Section: make

In the EXTENDED DESCRIPTION section, change from:

CFLAGS=-O

to:

CFLAGS=-O 1

The specification for the c99 utility does not support -O without an option-argument. In addition, if a subsequent argument is provided, the argument will be (erroneously) interpreted as an optimization level.

Change Number: XCU/TC2/D4/0132 [763]

On Page: 2922 Line: 96124 Section: make

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

As a consequence of the general rules for target updating, a useful special case is that if a target has no prerequisites and no commands, and the target of the rule is a nonexistent file, then make acts as if this target has been updated whenever its rule is run. (Note: This implies that all targets depending on this one will always have their commands run.)


Change Number: XCU/TC2/D4/0133 [857]

On Page: 2922 Line: 96124 Section: make

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

Shell command sequences like make; cp original copy; make may have problems on filesystems where the timestamp resolution is the minimum (1 second) required by the standard and where make considers identical timestamps to be up to date. Conversely, rules like copy: original; cp -p original copy will result in redundant work on make implementations that consider identical timestamps to be out of date.


Change Number: XCU/TC2/D4/0134 [866]

On Page: 2922 Line: 96124 Section: make

In the APPLICATION USAGE section, add the following to the end of the section:

This standard does not specify precedence between macro definition and include directives. Thus the behavior of

```
include -foo.mk
```

is unspecified. To define a variable named include, either the whitespace before the <equal-sign> should be removed, or another macro should be used, as in

```
INCLUDE_NAME = include
$(INCLUDE_NAME) =-foo.mk
```

On the other hand, if the intent is to include a file which starts with an <equal-sign>, either the filename
should be changed to `.=/foo.mk`, or the makefile should be written as

```
INCLUDE_FILE = -foo.mk
include $(INCLUDE_FILE)
```


**Change Number:** XCU/TC2/D4/0135 [525]

**On Page:** 2923 Line: 96134 Section: make

**In the EXAMPLES section, change from:**

```
pgm: a.o b.o
to:
```

```
.POSIX:
pgm: a.o b.o
```


**The standard requires that a portable makefile start with the special target .POSIX, then proceeds to give an example makefile that does not use that target. The remaining examples only claim to be a portion of a makefile, rather than an entire makefile.**

**Change Number:** XCU/TC2/D4/0136 [848]


**In the EXAMPLES section, change from:**

```
.c.o:
c99 -c -O $*.c
```

or:

```
.c.o:
c99 -c -O <$
```

**to:**

```
.c.o:
c99 -c -O 1 $*.c
```

or:

```
.c.o:
c99 -c -O 1 $<
```


**The specification for the c99 utility does not support -O without an option-argument. In addition, if a**
subsequent argument is provided, the argument will be (erroneously) interpreted as an optimization level.

**Change Number:** XCU/TC2/D4/0137 [769]

On Page: 2924 Line: 96189 Section: make

In the RATIONALE section, insert a new bullet item:

- Specifying an alternate shell to use to process commands.


**Change Number:** XCU/TC2/D4/0138 [525]

On Page: 2926 Line: 96267-96271 Section: make

In the RATIONALE section, delete:

The best way to provide portable makefiles is to include all of the rules needed in the makefile itself. The rules provided use only features provided by other parts of this volume of POSIX.1-2008. The default rules include rules for optional commands in this volume of POSIX.1-2008. Only rules pertaining to commands that are provided in an implementation's default set.


This text is duplicated in the APPLICATION USAGE section (lines 96098-96102).

**Change Number:** XCU/TC2/D4/0139 [769]

On Page: 2926 Line: 96276-96279 Section: make

In the RATIONALE section, change from:

The historical *MAKESHELL* feature was omitted. In some implementations it is used to let a user override the shell to be used to run *make* commands. This was confusing; for a portable *make*, the shell should be chosen by the makefile writer or specified on the *make* command line and not by a user running *make*.

Further, a makefile writer cannot require an alternate shell to be used and still consider the makefile portable. While it would be possible to standardize a mechanism for specifying an alternate shell, existing implementations do not agree on such a mechanism, and makefile writers can already invoke an alternate shell by specifying the shell name in the rule for a target (e.g., `python -c "foo"`).

7810 Change Number: XCU/TC2/D4/0140 [505]
7811 On Page: 2926 Line: 96289-96293 Section: make
7812 In the RATIONALE section, change from:
7813 The algorithm for determining a new entry for target rules is partially unspecified. Some historical makes allow blank, empty, or comment lines within the collection of commands marked by leading <tab> characters. A conforming makefile must ensure that each command starts with a <tab>, but implementations are free to ignore blank, empty, and comment lines without triggering the start of a new entry.
7814
7815 to:
7816 The algorithm for determining a new entry for target rules is partially unspecified. Some historical makes allow comment lines (including blank and empty lines) within the collection of commands marked by leading <tab> characters. A conforming makefile must ensure that each command starts with a <tab>, but implementations are free to ignore comments without triggering the start of a new entry.
7817
7819 The standard as currently written does not allow blank lines and empty lines in places where all known implementations have allowed them to appear. These changes more accurately reflect existing practice.

7826 Change Number: XCU/TC2/D4/0141 [693]
7827 On Page: 2927 Line: 96309 Section: make
7829 In the RATIONALE section, change from:
7830 The environment is the same as the environment to make except that MAKEFLAGS and macros defined on the make command line are added.
7831 to:
7832 The environment is the same as the environment to make except that MAKEFLAGS and macros defined on the make command line are added, and except that macros defined by the MAKEFLAGS environment variable and macros defined in the makefile(s) may update the value of an existing environment variable (other than SHELL).

7837 Change Number: XCU/TC2/D4/0142 [505]
7838 On Page: 2928 Line: 96368 Section: make
7839 In the RATIONALE section, add a new paragraph to the end of the section:
7840 Earlier versions of this standard defined comment lines only as lines with '#' as the first character. Many places then talked about comments, blank lines, and empty lines; but some places inadvertently only mentioned comments when blank lines and empty lines had also been accepted in all known implementations. The standard now defines comment lines to be blank lines, empty lines, and lines starting
with a '#' character and explicitly lists cases where blank lines and empty lines are not acceptable.


The standard as currently written does not allow blank lines and empty lines in places where all known implementations have allowed them to appear. These changes more accurately reflect existing practice.

Change Number: XCU/TC2/D4/0143 [857]

On Page: 2928 Line: 96368 Section: make

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

On most historic systems, the make utility considered a target with a prerequisite that had an identical timestamp as up-to-date. The HP-UX implementation of make treated it as out-of-date. The standard now allows either behavior, but implementations are encouraged to follow the example set by HP-UX. This is especially important on filesystems where the timestamp resolution is the minimum (1 second) required by the standard. All implementations of make should make full use of the finest timestamp resolution available on the filesystems holding targets and prerequisites to ensure that targets are up-to-date even for prerequisite files with timestamps that were updated within the same second. However, if the timestamp resolutions of the filesystems containing a target and a prerequisite are different, the timestamp with the more precise resolution should be rounded down to the resolution of the less precise timestamp for the comparison.


Change Number: XCU/TC2/D4/0144 [693,865]

On Page: 2928 Line: 96370 Section: make

In the FUTURE DIRECTIONS section, change from:

None.

to:

Some implementations of make include an "export" directive to add specified make variables to the environment. This may be considered for standardization in a future revision.

A future version of this standard may require that macro expansions using the forms $(string1:[op][os]=[np][%][ns]) or ${string1:[op][os]=[np][%][ns]} are treated as pattern macro expansions.

In the EXIT STATUS section, change from:

All the specified directories were created successfully or the -p option was specified and all the specified directories now exist.

to:

All the specified directories were created successfully, or the -p option was specified and all the specified directories either already existed or were created successfully.


Change Number: XCU/TC2/D4/0146 [584]

On Page: 2945 Line: 96927 Section: more

In the ENVIRONMENT VARIABLES section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0147 [534]

On Page: 2955 Line: 97307 Section: mv

In the DESCRIPTION section, change from:

name the same existing file

to:

resolve to either the same existing directory entry or different directory entries for the same existing file


Change Number: XCU/TC2/D4/0148 [744]

On Page: 2973 Line: 97928 Section: nm

In the DESCRIPTION section, change from:
On XSI-conformant systems, it shall be decimal.

to:

On XSI-conformant systems, it shall be decimal if the -P option is not specified.

In the OPTIONS section, change from:

do The offset is written in decimal [XSI](default)/[XSI].

d o The offset is written in octal.

dx The offset is written in hexadecimal.

to:

do decimal [XSI](default if -P is not specified)/[XSI].

do octal.

dx hexadecimal (default if -P is specified).

The changes remove the ambiguity in the specification.

Change Number: XCU/TC2/D4/0149 [973]

On Page: 2990 Line: 98598 Section: paste

In the OPTIONS section, for the -s option change from:

Concatenate all of the lines of each separate input file in command line order. The <newline> of every line except the last line in each input file shall be replaced with the <tab>, unless otherwise specified by the -d option.

to:

Concatenate all of the lines from each input file into one line of output per file, in command line order. The <newline> of every line except the last line in each input file shall be replaced with a <tab>, unless otherwise specified by the -d option. If an input file is empty, the output line corresponding to that file shall consist of only a <newline> character.

The description of paste -s does not say what happens if an input file is empty. Existing practice is to write an empty line for that file.

Change Number: XCU/TC2/D4/0150 [584]
On Page: 3001 Line: 99000 Section: pathchk
In the OPTIONS section, change from:

<pre>&lt;hyphen&gt;</pre>
to:

<pre>&lt;hyphen-minus&gt;</pre>


Change Number: XCU/TC2/D4/0151 [584]

On Page: 3004 Line: 99135 Section: pathchk
In the RATIONALE section, change from:

<pre>&lt;hyphen&gt;</pre>
to:

<pre>&lt;hyphen-minus&gt;</pre>


Change Number: XCU/TC2/D4/0152 [886]

On Page: 3006 Line: 99191 Section: pax
In the DESCRIPTION section, change from:

... except that there may be hard links between the original and the copied files.
to:

... except that copying of sockets may be supported even if archiving them in write mode is not supported, and that there may be hard links between the original and the copied files.


Change Number: XCU/TC2/D4/0153 [814]

On Page: 3012 Line: 99444 Section: pax
In the OPTIONS section, add the following text:

If the <code>-o delete=pattern</code>, <code>-o keyword=value</code>, or <code>-o keyword:=value</code> options are used to override or remove any extended header data needed to find files in an archive (e.g., <code>-o delete=size</code> for a file whose size...
cannot be represented in a *ustar* header or \( -o \text{ size}=100 \) for a file whose size is not 100 bytes), the behavior is undefined.


If a user overrides data needed to determine where a file is in an archive, the results are undefined.

**Change Number:** XCU/TC2/D4/0154 [886]

On Page: 3026 Line: 100038 Section: pax

In the EXTENDED DESCRIPTION section, change from:

Attempts to archive a socket using *ustar* interchange format shall produce a diagnostic message.

to:

Attempts to archive a socket shall produce a diagnostic message when *ustar* interchange format is used, but may be allowed when *pax* interchange format is used.


**Change Number:** XCU/TC2/D4/0155 [707]

On Page: 3041 Line: 100701 Section: pax

Add `<tar.h>` to the XBD part of the SEE ALSO list.


**Change Number:** XCU/TC2/D4/0156 [727]

On Page: 3050 Line: 101034 Section: printf

In the EXTENDED DESCRIPTION section, in item 7 change from:

a string that may contain

to:

a string that can contain

7990 Change Number: XCU/TC2/D4/0157 [727,932]
7991 On Page: 3050 Line: 101054 Section: printf
7993 In the EXTENDED DESCRIPTION section, in item 9 change from:
7994 Any extra c or s conversion
7995 to:
7996 Any extra b, c, or s conversion
7997 Rationale: Austin Group Defect Report(s) applied: 727,932. See
8000 This appears to be an oversight in the standard. All known implementations are believed to already behave
8001 as per this change.
8002 Change Number: XCU/TC2/D4/0158 [584]
8003 On Page: 3051 Line: 101068 Section: printf
8004 In the EXTENDED DESCRIPTION section, change from:
8005 A leading <plus-sign> or minus-sign shall be allowed.
8006 to:
8007 A leading <plus-sign> or <hyphen-minus> shall be allowed.
8009 Change Number: XCU/TC2/D4/0159 [727]
8010 On Page: 3051 Line: 101078 Section: printf
8012 In the EXTENDED DESCRIPTION section, change from:
8013 completely used for a c or s conversion
8014 to:
8015 completely used for a b, c, or s conversion
Change Number: XCU/TC2/D4/0160 [584]

On Page: 3063 Line: 101564 Section: ps

In the STDOUT section, change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XCU/TC2/D4/0161 [471]

On Page: 3067 Line: 101688 Section: pwd

In the OPTIONS section, change from:

If the PWD environment variable contains an absolute pathname of the current directory that does not contain the filenames dot or dot-dot, pwd shall write this pathname to standard output. Otherwise, if the PWD environment variable contains a pathname of the current directory that is longer than {PATH_MAX} bytes including the terminating null, and the pathname does not contain any components that are dot or dot-dot, it is unspecified ...

to:

If the PWD environment variable contains an absolute pathname of the current directory and the pathname does not contain any components that are dot or dot-dot, pwd shall write this pathname to standard output, except that if the PWD environment variable is longer than {PATH_MAX} bytes including the terminating null, it is unspecified ...


Change Number: XCU/TC2/D4/0162 [958]

On Page: 3128 Line: 103902 Section: read


In the NAME section, change from:

read a line from standard input

to:

read from standard input into shell variables
On Page: 3128 Line: 103906 Section: read


In the DESCRIPTION section, change from:

The read utility shall read a single line from standard input.

to:

The read utility shall read a single logical line from standard input into one or more shell variables.


Change Number: XCU/TC2/D4/0163 [542]

On Page: 3135 Line: 104163 Section: rm

In the SYNOPSIS section, change from:

rm [-f1rR] file...

to:

rm [-iRr] file...


Change Number: XCU/TC2/D4/0164 [819]

On Page: 3135 Line: 104178 Section: rm

In the DESCRIPTION section, insert the following text at the start of step 2b:

If file is an empty directory, rm may skip to step 2d.

The standard does not allow an optimization that is currently in use.

Change Number: XCU/TC2/D4/0165 [542]

On Page: 3136 Line: 104209 Section: rm

In the OPTIONS section, change from:

Do not write diagnostic messages or modify the exit status in the case of nonexistent operands.

to:

Do not write diagnostic messages or modify the exit status in the case of no file operands, or in the case of
operands that do not exist.


8076  **Change Number:** XCU/TC2/D4/0166 [945]

8077  On Page: 3154 Line: 104823 Section: sed
8079  In the STDERR section, change from:
8080  The standard error shall be used only for diagnostic messages.
8081  to:
8082  The standard error shall be used only for diagnostic and warning messages.


8084  **Change Number:** XCU/TC2/D4/0167 [944]

8085  On Page: 3156 Line: 104896 Section: sed
8087  In the EXTENDED DESCRIPTION section, change from:
8088  Command verbs other than {, a, b, c, i, r, t, w, }, and # can be followed by a <semicolon>, optional
8089  <blank> characters, and another command verb. However, when the s command verb is used with the w
8090  flag, following it with another command in this manner produces undefined results.
8091  to:
8092  Editing commands other than {...}, a, b, c, i, r, t, w, and # can be followed by a <semicolon>, optional
8093  <blank> characters, and another editing command. However, when an s editing command is used with the
8094  w flag, following it with another command in this manner produces undefined results.
8095  On Page: 3156 Line: 104900 Section: sed
8097  In the EXTENDED DESCRIPTION section, change from:
8098  A function can be preceded by one or more '! ' characters,
8099  to:
8100  A function can be preceded by a '!' character,

8102  Some implementations do not support multiple '!' characters before a function. Since it is not portable in
8103  practice, it seems unlikely that any applications would be affected by removing the requirement from the
8104  standard.
In the EXTENDED DESCRIPTION section, add a new paragraph:

If a `label` argument (to a `b`, `t`, or `:` command) contains characters outside of the portable filename character set, or if a `label` is longer than 8 bytes, the behavior is unspecified. The implementation shall support `label` arguments recognized as unique up to at least 8 bytes; the actual length (greater than or equal to 8) supported by the implementation is unspecified. It is unspecified whether exceeding the maximum supported label length causes an error or a silent truncation.


Historical sed implementations did not consistently handle certain characters in label arguments; some interpreted backslashes as escape characters while others did not, some did not treat characters like `'`, `}`, and whitespace like other label characters, and some limited label length by bytes rather than characters. Limiting conforming scripts to using labels with names created from characters in the portable filename character set allows those scripts to run on all implementations.

In the EXTENDED DESCRIPTION section, change from:

Execute a list of *sed* functions only when the pattern space is selected. The list of *sed* functions shall be surrounded by braces and separated by `<newline>` characters, and conform to the following rules. The braces can be preceded or followed by `<blank>` characters. The functions can be preceded by `<blank>` characters, but shall not be followed by `<blank>` characters. The `<right-brace>` shall be preceded by a `<newline>` or `<semicolon>` (before any optional `<blank>` characters preceding the `<right-brace>`).

Each command in the list of commands shall be terminated by a `<newline>` character, or by a `<semicolon>` character if permitted when the command is used outside the braces. The editing commands can be preceded by `<blank>` characters, but shall not be followed by `<blank>` characters.


This is a layered change on XCU/TC1/D5/0135 [269].

The use of `<semicolon>` within `{...}` is common practice and should be specified in the standard. The standard developers are not aware of any implementation that does not already support it.
8142  Change Number: XCU/TC2/D4/0170 [945]
8143  On Page: 3156 Line: 104917 Section: sed
8145  In the EXTENDED DESCRIPTION section, change the entire description of the b command to:
8146  Branch to the : command verb bearing the label argument. If label is not specified, branch to the end of the script.
8149  Change Number: XCU/TC2/D4/0171 [533]
8150  On Page: 3158 Line: 104978,104982 Section: sed
8151  In the EXTENDED DESCRIPTION section, at line 104978 delete:
8152  The meaning of a <backslash> immediately followed by any character other than '&', <backslash>, a digit, or the delimiter character used for this command, is unspecified.
8153  At line 104982 change from:
8154  The application shall escape the <newline> in the replacement by preceding it by a <backslash>. A substitution ...
8155  to:
8156  The application shall escape the <newline> in the replacement by preceding it by a <backslash>.
8157  The meaning of an unescaped <backslash> immediately followed by any character other than '&', <backslash>, a digit, <newline>, or the delimiter character used for this command, is unspecified.
8158  A substitution ...
8163  Change Number: XCU/TC2/D4/0172 [663]
8164  On Page: 3159 Line: 105039 Section: sed
8166  In the APPLICATION USAGE section, add a new paragraph:
8167  When using sed 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.
In the RATIONALE section, change from:

Implementors are encouraged to provide warning messages about labels that are never used or jumps to labels that do not exist.

to:

Implementors are encouraged to provide warning messages about labels that are never referenced by a `b` or `t` command, jumps to labels that do not exist, and label arguments that are subject to truncation.


In the RATIONALE section, delete:

Historically, the `sed` `!` and `J` editing commands did not permit multiple commands on a single line using a `;` as a command delimiter. Implementations are permitted, but not required, to support this extension.


In the OPTIONS section, change from:

`<hyphen>`

to:

`<hyphen-minus>`


In the OPERANDS section, change from:

`<hyphen>`

In the INPUT FILES section, change from:

If the input file is empty or consists solely of blank lines or comments, or both, \texttt{sh} shall exit with a zero exit status.

to:

If the input file consists solely of zero or more blank lines and comments, \texttt{sh} shall exit with a zero exit status.

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

Change Number: XCU/TC2/D4/0177 [718]

On Page: 3165 Line: 105253 Section: \texttt{sh}

In the ENVIRONMENT VARIABLES section, delete the entry for \texttt{IFS}.

Rationale: Austin Group Defect Report(s) applied: 884. See \url{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/D4/0178 [884]

On Page: 3165-3166 Line: 105293-105300 Section: \texttt{sh}


In the EXIT STATUS section, change from:

1-125 A non-interactive shell detected a syntax, redirection, or variable assignment error.

to:

1-125 A non-interactive shell detected an error other than \texttt{command_file} not found or executable, including but not limited to syntax, redirection, or variable assignment errors.

126 A specified \texttt{command_file} could not be executed due to an ENOEXEC error (See Section...
2.9.1.1, item 2).


This change is layered on Change Number: XCU/TC1/D5/0141 [299]

Change Number: XCU/TC2/D4/0180 [884]

On Page: 3177 Line: 105801 Section: sh

In the RATIONALE section, change from:

The KornShell ignores the contents of IFS upon entry to the script. A conforming application cannot rely on importing IFS. One justification for this, beyond security considerations, is to assist possible future shell compilers.


One justification for ignoring the contents of IFS upon entry to the script, beyond security considerations, is to assist possible future shell compilers.


Change Number: XCU/TC2/D4/0181 [584]

On Page: 3177 Line: 105820 Section: sh

In the DESCRIPTION section, change from:

... shall be performed using the collating sequence of the current locale.


Change Number: XCU/TC2/D4/0182 [963]

On Page: 3183 Line: 106005 Section: sort

In the DESCRIPTION section, change from:

... shall be performed using the collating sequence of the current locale.

273
Copyright © 2016 IEEE and The Open Group. All rights reserved.

Change Number: XCU/TC2/D4/0183 [584]

On Page: 3184 Line: 106038 Section: sort
In the OPTIONS section, change from:
optional minus-sign,

to:
optional <hyphen-minus> character,


Change Number: XCU/TC2/D4/0184 [510]

On Page: 3184 Line: 106060 Section: sort
In the OPTIONS section, for the -k option, change from:

`type` is a modifier from the list

to:

`type` is one or more modifiers from the list


Change Number: XCU/TC2/D4/0185 [962]

On Page: 3184 Line: 106079 Section: sort
In the OPERANDS section, append the following to the description of the file operand:

If `sort` encounters an error when opening or reading a `file` operand, it may exit without writing any output to standard output or processing later operands.

On Page: 3186 Line: 106148 Section: sort

In the CONSEQUENCES OF ERRORS section, change from:

Default.

to:

The default requirements shall apply, except that if `sort` encounters an error when opening or reading a `file`
operand, it may exit without writing any output to standard output or processing later operands.


Existing implementations of sort stop processing immediately after encountering an error when reading input files.

Change Number: XCU/TC2/D4/0186 [663]

On Page: 3187 Line: 106180 Section: sort

In the APPLICATION USAGE section, add a new paragraph:

When using sort 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.


Change Number: XCU/TC2/D4/0187 [963]

On Page: 3187 Line: 106180 Section: sort

In the APPLICATION USAGE section, add a new paragraph:

If the collating sequence of the current locale does not have a total ordering of all characters, this can affect the behavior of sort in the following ways:

- As sort -u suppresses lines with duplicate keys, it suppresses lines that collate equally but are not identical.
- The output of sort (without -u) can contain identical lines that are not adjacent, if it does not implement the recommended further byte-by-byte comparison of lines that collate equally. This affects the use of sort with comm and uniq; see the APPLICATION USAGE for those utilities.

On Page: 3188 Line: 106244 Section: sort

In the RATIONALE section, add a new paragraph:

Implementations are encouraged to perform the recommended further byte-by-byte comparison of lines that collate equally, even though this may affect efficiency. The impact on efficiency can be mitigated by only performing the additional comparison if the current locale's collating sequence does not have a total ordering of all characters (if the implementation provides a way to query this) or by only performing the additional comparison if the locale name associated with the LC_COLLATE category has an '@' modifier in the name (since locales without an '@' modifier should have a total ordering of all characters - see [xref to XBD 7.3.2]). Note that if the implementation provides a stable sort option as an extension (usually -s), the additional comparison should not be performed when this option has been specified.

On Page: 3188 Line: 106246 Section: sort
In the FUTURE DIRECTIONS section, change from:

None.

to:

A future version of this standard may require that if the collating sequence of the current locale does not have a total ordering of all characters, any lines of input that collate equally when comparing them as whole lines are further compared byte-by-byte using the collating sequence for the POSIX locale.


Change Number: XCU/TC2/D4/0188 [731]

On Page: 3190 Line: 106264-106282 Section: split

In the NAME section, on line 106264 change from:

split files into pieces

to:

split a file into pieces

In the SYNOPSIS section, on lines 106266 and 106267 change from:

```
[file[name]]
```

to:

```
[file [name]]
```

In the DESCRIPTION section, on line 106269 change from:

one or more

to:

zero or more

On line 106282 add to the end of the final paragraph of the DESCRIPTION section:

If the input is an empty file, no output file shall be created and this shall not be considered to be an error.


When the input is an empty file, historical practice has been to produce no output files. We see no reason why conforming implementations should be required to created an empty output file in this case.
In the OPERANDS section, under Output Modes add a new item with XSI shading:

```
onlcr
(-onlcr)
```

Map (do not map) NL to CR-NL on output. This shall have the effect of setting (not setting) ONLCR in the `termios` c_oflag field, as defined in [xref to XBD Chapter 11].


This was present in XPG3, but appears to have been accidentally dropped in XPG4.

**Change Number:** XCU/TC2/D4/0189 [908]

---

**Change Number:** XCU/TC2/D4/0190 [663]

In the APPLICATION USAGE section, add a new paragraph:

When using `tail` to process pathnames, and the `-c` option is not specified, 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.


---

**Change Number:** XCU/TC2/D4/0191 [898]

In the DESCRIPTION section, change from:

```
In the second form of the utility, which uses "[ ]" rather than `test`, the application shall ensure that the square brackets are separate arguments.
```

to:

```
In the second form of the utility, where the utility name used is `/` rather than `test`, the application shall ensure that the closing square bracket is a separate argument. The `test` and `/` utilities may be implemented as a single linked utility which examines the basename of the zeroth command line argument to determine whether to behave as the `test` or `/` variant. Applications using the `exec()` family of functions to execute these utilities shall ensure that the argument passed in `argv[0]` or `argv[0]` is `"["` when executing the `/` utility and has a basename of "test" when executing the `test` utility.
```

On Page: 3223 Line: 107445 Section: test

In the DESCRIPTION section, change from:

True if pathname resolves to an existing directory entry for a block special file.

to:

True if pathname resolves to an existing directory entry for a block special file.


Change Number: XCU/TC2/D4/0193 [898]

On Page: 3228 Line: 107661 Section: test


In the APPLICATION USAGE section, add a new paragraph:

Note that none of the following examples are permitted by the syntax described:

[-f file]
[-f file]
[ -f file]
[ -f file

test -f file]

In the first two cases, if a utility named /f exists, that utility would be invoked, and not test. In the remaining cases, the brackets are mismatched, and the behavior is unspecified. However,

test !]

does have a defined meaning, and must exit with status 1. Similarly,

test ]

must exit with status 0.


Change Number: XCU/TC2/D4/0194 [723]

On Page: 3233 Line: 107828 Section: time

In the STDERR section, change from:

The standard error shall be used to write the timing statistics. If -p is specified, the following format shall be used in the POSIX locale:

to:

Copyright © 2016 IEEE and The Open Group. All rights reserved.
If the utility utility is invoked, the standard error shall be used to write the timing statistics and may be used to write a diagnostic message if the utility terminates abnormally; otherwise the standard error shall be used to write diagnostic messages and may also be used to write the timing statistics.

If -p is specified, the following format shall be used for the timing statistics in the POSIX locale:

```
```


Existing practice includes the possibility of diagnostics, and also the possibility of no timing when only diagnostics appear.

**Change Number:** XCU/TC2/D4/0195 [474]

On Page: 3239 Line: 108047-108053 Section: touch

In the APPLICATION USAGE, delete:

One ambiguous situation occurs if `-t time` is not specified, `-r ref_file` is not specified, and the first operand is an eight or ten-digit decimal number. A portable script can avoid this problem by using:

```
touch -- file
```

or:

```
touch ./file
```

in this case.


**Change Number:** XCU/TC2/D4/0196 [663]

On Page: 3250 Line: 108473 Section: tr

(2013 edition Page: 3277 Line: 110019) In the RATIONALE section, delete:

This meant that historical practice of being able to specify `tr -cd\000-\177` (which would delete all bytes with the top bit set) would have no effect because, in the C locale, bytes with the values octal 200 to octal 377 are not characters.


**Change Number:** XCU/TC2/D4/0197 [584]

On Page: 3265 Line: 108979 Section: umask

In the EXAMPLES section, change from:

```
<hyphen>
```

to:

```
```
8440 <hyphen-minus>
8442 Change Number: XCU/TC2/D4/0198 [885]
8444 In the RATIONALE section, change from:
8445 base documents
8446 to:
8447 base documents for POSIX.2-1992
8449 Change Number: XCU/TC2/D4/0199 [963]
8451 In the ENVIRONMENT VARIABLES section, delete:
8452 LC_COLLATE
8453 Determine the locale for ordering rules.
8455 In the APPLICATION USAGE section, change from:
8456 The sort utility can be used to cause repeated lines to be adjacent in the input file.
8457 to:
8458 If the collating sequence of the current locale has a total ordering of all characters, the sort utility can be used to cause repeated lines to be adjacent in the input file. If the collating sequence does not have a total ordering of all characters, the sort utility should still do this but it might not. To ensure that all duplicate lines are eliminated, and have the output sorted according the collating sequence of the current locale, applications should use:
8459 LC_ALL=C sort -u | sort
8460 instead of:
8461 sort | uniq
To remove duplicate lines based on whether they collate equally instead of whether they are identical, applications should use:

```
sort -u
```

instead of:

```
sort | uniq
```


**Change Number:** XCU/TC2/D4/0200 [663]

On Page: 3283 Line: 109551 Section: uniq

In the APPLICATION USAGE section, add a new paragraph:

When using `uniq` 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.


**Change Number:** XCU/TC2/D4/0201 [635]

On Page: 3291 Line: 109831 Section: uudecode

In the DESCRIPTION section, change from:

```
If the pathname of the file to be produced exists, and the user does not have write permission on that file, `uudecode` shall terminate with an error. If the pathname of the file to be produced exists, and the user has write permission on that file, the existing file shall be overwritten.
```

to:

```
If the pathname of the file to be produced exists, and the user does not have write permission on that file, `uudecode` shall terminate with an error. If the pathname of the file to be produced exists, and the user has write permission on that file, the existing file shall be overwritten and, if possible, the mode bits of the file (other than standard output) shall be set as described above; if the mode bits cannot be set, `uudecode` shall not treat this as an error.
```

**Rationale:** Austin Group Defect Report(s) applied: 635. See [http://austingroupbugs.net/view.php?id=635](http://austingroupbugs.net/view.php?id=635). The current standard does not match existing (and desired) behavior.

**Change Number:** XCU/TC2/D4/0202 [812]

On Page: 3314 Line: 110686 Section: vi

In the EXTENDED DESCRIPTION section, change from:
... specified by the map command (see ed).

to:

... specified by the map command (see ex).


Change Number: XCU/TC2/D4/0203 [499]

On Page: 3382 Line: 113189 Section: xargs

In the OPTIONS section, for the -L option change from:

... is a <blank>; a trailing <blank> signals continuation to the next non-empty line, inclusive.

to:

... is an unescaped <blank>; a trailing unescaped <blank> signals continuation to the next non-empty line, inclusive.


Change Number: XCU/TC2/D4/0204 [977]

On Page: 3388 Line: 113441-113443 Section: yacc


In the OPTIONS section, for the -d option change from:

Write the header file; by default only the code file is written. The #define statements associate the token codes assigned by yacc with the user-declared token names. This allows source files other than y.tab.c to access the token codes.

to:

Write the header file; by default only the code file is written. See the OUTPUT FILES section.

5. Changes to Rationale

This section contains the set of changes to the text of the Rationale. [Note to reviewers: References to defect reports are provided to aid reviewers.]

Change Number: XRAT/TC2/D4/0001 [937]

On Page: 3416 Line: 114407 Section: A.1.5 Terminology

Delete the paragraph:

The term "application" is not defined in POSIX.1-2008; it is assumed to be a part of general computer science terminology.


Change Number: XRAT/TC2/D4/0002 [934]


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

A.4.2 Default Initialization

There is no additional rationale provided for this section.


Change Number: XRAT/TC2/D4/0003 [584]

On Page: 3446 Line: 115631 Section: A.4.7 Filename Portability

Change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XRAT/TC2/D4/0004 [543]


Change from:
There is no additional rationale provided for this section.

It is intended that undeserved underflow and inexact floating-point exceptions are raised only if avoiding them would be too costly.


**Change Number:** XRAT/TC2/D4/0005 [663]

On Page: 3454 Line: 115933 Section: A.6.2 Character Encoding

Add a new paragraph to the end of the section:

Earlier versions of this standard did not state the requirement that the POSIX locale contains 256 single-byte characters. This was an oversight; the intention was always that the POSIX locale should have an 8-bit-clean single-byte encoding.


**Change Number:** XRAT/TC2/D4/0006 [745]

On Page: 3454 Line: 115933 Section: A.6.2 Character Encoding

Add a new paragraph to the end of the section:

The encodings for `<newline>` and `<carriage-return>` are required to be the same across all locales since they are special to the general terminal interface and cannot be changed (see [xref to XBD 11.1.9]).


**Change Number:** XRAT/TC2/D4/0007 [796]

On Page: 3457 Line: 116059 Section: A.7.2 POSIX Locale

Change from:

The POSIX locale is equal to the C locale.

to:

On POSIX.1 implementations the POSIX locale is equal to the C locale, even though the requirements for the POSIX locale are more extensive than the C standard's requirements for the C locale.

Change Number: XRAT/TC2/D4/0008 [584]

On Page: 3461 Line: 116234, 116235, 116236 Section: A.7.3.2 LC_COLLATE

Change from:

<hyphen>

to:

<hyphen-minus>


Change Number: XRAT/TC2/D4/0009 [938]

On Page: 3461 Line: 116237 Section: A.7.3.2 LC_COLLATE


Add a new paragraph:

This standard recommends (by the use of “should” in the normative text) that all implementation-provided locales 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. Defining locales in this way eliminates unexpected behavior when non-identical strings can collate equally (for example, `sort -u` and `sort | uniq` are not equivalent). The exception for locales with a suitable '@' modifier in the name allows implementations to supply locales which do not have a total ordering of all characters provided that they draw attention to it in the modifier name. For example, `@icase` could indicate that each upper- and lower-case character pair collates equally. Even with an '@' modifier, total ordering is preferred when possible; for example, characters that are "ignored" in dictionary order need not be completely ignored (by using `IGNORE` for all collation weights), but can instead be given a unique weight after one or more `IGNORE` weights.


Change Number: XRAT/TC2/D4/0010 [796]

On Page: 3468 Line: 116551 Section: A.8.2 Internationalization Variables


Change from:

If this were omitted, the ISO C standard specifies that the C locale would be used.

to:

If this were omitted, the ISO C standard specifies that the C (or POSIX) locale would be used.

Change Number: XRAT/TC2/D4/0011 [584]

On Page: 3469 Line: 116625 Section A.8.3 Other Environment Variables

In the description of TZ change from:

the character plus ('+'), the character minus ('-'), or digits),

to:

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


Change Number: XRAT/TC2/D4/0012 [496]

On Page: 3505 Line: 117934 Section: B.2.3 Error Numbers


Change from:

The return of [ENOSYS] is to be taken to indicate that the function of the interface is not supported at all;
the function will always fail with this error code.

to:

In some earlier versions of this standard, the difference between [ENOTSUP] and [ENOSYS] was that
[ENOSYS] indicated that the function was not supported at all. This is no longer the case as [ENOSYS]
can also be used to indicate non-support of optional functionality for a function that has some required
functionality. (See XSH encrypt().)

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

Change Number: XRAT/TC2/D4/0013 [681]

On Page: 3507 Line: 118020 Section: B.2.3 Error Numbers


Change from:

EOPNOTSUP

to:

EOPNOTSUPP

Change Number: XRAT/TC2/D4/0014 [516]

On Page: 3514 Line: 118375 Section: B.2.4.3 Signal Actions

Change from:

The behavior of unsafe functions, as defined by this section, is undefined when they are invoked from signal-catching functions in certain circumstances.

to:

The behavior of unsafe functions, as defined by this section, is undefined when they are called from (or after a \texttt{longjmp}() or \texttt{siglongjmp}() out of) signal-catching functions in certain circumstances.

On Page: 3515 Line: 118392,118400 Section: B.2.4.3 Signal Actions

At line 118392 change from:

Note that \texttt{longjmp}() and \texttt{siglongjmp}() are not in the list of async-signal-safe functions.

to:

Note that although \texttt{longjmp}() and \texttt{siglongjmp}() are in the list of async-signal-safe functions, there are restrictions on subsequent behavior after the function is called from a signal-catching function.

At line 118400 change from:

POSIX.1 does not define the behavior when any unsafe function is called in a signal handler that interrupts any unsafe function.

to:

POSIX.1 does not define the behavior when any unsafe function is called in (or after a \texttt{longjmp}() or \texttt{siglongjmp}() out of) a signal handler that interrupts any unsafe function or the non-async-signal-safe processing equivalent to \texttt{exit}() that is performed after return from the initial call to \texttt{main}().

\textit{Rationale}: Austin Group Defect Report(s) applied: 516. See \url{http://austingroupbugs.net/view.php?id=516}.

The restrictions on using \texttt{longjmp}() and \texttt{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: XRAT/TC2/D4/0015 [904]

On Page: 3578 Line: 121252 Section: B.2.9.1 Thread-Safety


Change from:

While a read from a pipe of [\texttt{PIPE_MAX}]*2 bytes may not generate a single atomic ...

to:
While a read from a pipe of $\text{PIPE\_BUF} \times 2$ bytes may not generate a single atomic ...


**Change Number:** XRAT/TC2/D4/0016 [829]

On Page: 3622 Line: 123021 Section: B.2.12.3 Pointer Types

Delete section B.2.12.3.


Section 2.12.3 in XSH has previously been deleted.

**Change Number:** XRAT/TC2/D4/0017 [690]

On Page: 3622 Line: 123023 Section: B.2 General Information


Add a new section:

**B.2.13 Status Information**

POSIX.1-2008 does not require all matching WNOWAIT threads (threads in a matching call to `waitid()` with the WNOWAIT flag set) to obtain a child's status information because the status information might be discarded (consumed or replaced) before one of the matching WNOWAIT threads is scheduled. If the status information is not discarded, it will remain available, so all of the matching WNOWAIT threads will (eventually) obtain the status information.


**Change Number:** XRAT/TC2/D4/0018 [835]

On Page: 3622 Line: 123023 Section: B.2 General Information

Add a new section:

**B.2.14 File Descriptor Allocation**

Functions such as `pipe()` and `socketpair()` which allocate two file descriptors are permitted to perform the two allocations independently. This means that other threads or signal handlers may perform operations on file descriptors in between the two allocations and this can result in the two file descriptors not having adjacent values or in the second allocation producing a lower value than the first.


**Change Number:** XRAT/TC2/D4/0019 [584]

On Page: 3642 Line: 123784 Section: C.1.5 Utility Description Defaults

Change from:
Some examples of the '*' and '@' properties, including the concatenation aspects:

```bash
set "abc" "def ghi" "jkl"

echo $* => "abc" "def" "ghi" "jkl"
```

but:

```bash
echo "$@" => "abc" "def" "ghi" "jkl"
```

In the preceding examples, the double-quote characters that appear after the "=>" do not appear in the output and are used only to illustrate word boundaries.

The following example illustrates the effect of setting `IFS` to a null string:

```bash
IFS=''
set foo bar bam
printf '%s
' $
```

The following examples illustrate some of the ways in which '*' and '@' can be expanded:

```bash
set "abc" "def ghi" "jkl"
unset var
IFS=' ' # a space
printf '%s
' $*
```
printf '%s
' "$*"
abc def ghi jkl
printf '%s
' xx$*yy
xxabc
def
gi
jklyy
printf '%s
' "xx$*yy"
xxabc def ghi jklyy
printf '%s
' "$*"
abc
def
gi
jklyy
printf '%s
' "$*"
abc
def ghi
jk
printf '%s
' "$*" ${1+"$@"}
abc
def ghi
jk
printf '%s
' xx$@yy
xxabc
def
gi
jklyy
printf '%s
' "xx$@yy"
xxabc
def ghi
jklyy
printf '%s
' "$@"
abc
def ghi
jk
printf '%s
' "$@"
abc
def ghi
jk
printf '%s
' "$@"
abc
def ghi
jk
printf '%s
' "$@"
abc
def ghi
jk
IFS=':'
printf '%s
' "$*"
abc:def ghi:jkl
var="$*": printf '%s
' "$var"
abc:def ghi:jkl
var="$*": printf '%s
' "$var"
abc:def ghi:jkl
unset var
printf '%s
' "$var="$*
abc
def ghi
jk
printf '%s
' "$(var="$*)"
printf '%s
' "\$*"
8803      abc:def ghi:jkl
8804      printf '%s
' ${var="\$*"}
8805      abc:def ghi:jkl
8806      printf '%s
' ${var="\$*"}
8807      abc
def ghi
8808      jkl
8809      printf 'var=%s\n' "\$var"
8810      var=abc:def ghi:jkl
8811      unset var
8812      printf '%s\n' "\${var="\$*"}
8813      abc:def ghi:jkl
8814      printf 'var=%s\n' "\$var"
8815      var=abc:def ghi:jkl
8816      unset var
8817      IFS=' ' # null
8818      printf '%s\n' "\$*"
8819      abc
def ghi:jkl
8820      var="\$*"; printf '%s\n' "\$var"
8821      abc
def ghi:jkl
8822      var"="\$*"; printf '%s\n' "\$var"
8823      abc
def ghi:jkl
8824      unset var
8825      printf '%s\n' "\${var="\$*"}
8826      abc
def ghi:jkl
8827      printf '%s\n' "\${var="\$*"}
8828      abc
def ghi:jkl
8829      printf '%s\n' "\${var="\$*"}
8830      abc
def ghi:jkl
8831      printf '%s\n' "\${var="\$*"}
8832      abc
def ghi:jkl
8833      printf '%s\n' "\${var="\$*"}
8834      var=abc
def ghi:jkl
8835      unset var
8836      printf '%s\n' "\${var="\$*"}
8837      abc
def ghi:jkl
8838      printf 'var=%s\n' "$var"
8839      var=abc
def ghi:jkl
8840      printf '%s
' "$@"
8841      abc
def ghi
8842      jkl
8843
8844      unset IFS
8845      printf '%s\n' "\$*"
8846      abc
def ghi:jkl
8847      var="\$*"; printf '%s\n' "\$var"
8848      abc
def ghi:jkl
8849      var=""""; printf '%s\n' "\$var"
8850      abc
def ghi:jkl
8851      unset var
8852      printf '%s\n' "\${var="\$*"}
8853      abc
def ghi:jkl
8854      printf '%s\n' "\${var="\$*"}
8855      abc
def ghi:jkl
8856
8857      printf '%s\n' "\${var="\$*"}
8858      abc
def ghi:jkl
8859      printf '%s\n' "\${var="\$*"}
8860      abc
def ghi:jkl
8861      printf '%s\n' "\${var="\$*"}
8862      abc
def ghi:jkl
8863      printf '%s\n' "\${var="\$*"}
8864      abc
def
In all of the following commands the results of the expansion of `@` (if performed) are unspecified:
printf '%s
' "$@foo"
8924    printf '%s
' $@@foo
8925    printf '%s
' "$foo"
8926    printf '%s
' $var%@
8927    printf '%s
' $var%@
8928    printf '%s
' $var"@
8929    printf '%s
' $var"@
8930    printf '%s
' $var"@
8931    printf '%s
' $var"@
8932    printf '%s
' $var"@
8933    printf '%s
' $var"@
8934    printf '%s
' $var"@
8935    printf '%s
' $var"@
8936    printf '%s
' $var"@
8937    printf '%s
' $var"@


Change Number: XRAT/TC2/D4/0021 [935]

On Page: 3664 Line: 124689 Section: C.2.9.1 Simple Commands

Insert a new paragraph:

Various historical implementations have used the names in item 1.b. as built-ins or reserved words. This standard does not specify their behavior, but their existence means that it is important for portable applications to avoid giving functions (or utilities in PATH) those names because the function (or utility in PATH) might not be executed as expected.


Change Number: XRAT/TC2/D4/0022 [521]

On Page: 3665 Line: 124711 Section: C.2.9.2 Pipelines

Change from:

The reserved word ! allows more flexible testing using AND and OR lists.

to:

The reserved word ! allows more flexible testing using AND and OR lists. The behavior of !( is unspecified because in the Korn Shell this introduces a negated pathname expansion. Portable applications need to separate the ! and ( to ensure the command is treated as a negated subshell.


Requiring a negated subshell conflicts with shells that provide an extension of negated pathname expansion.

Change Number: XRAT/TC2/D4/0023 [736]

On Page: 3670 Line: 124914 Section: C.2.10 Shell Grammar

Delete the final paragraph of the section:

The start symbol of the grammar (complete_command) represents either input from the command line or a shell script. It is repeatedly applied by the interpreter to its input and represents a single "chunk" of that input as seen by the interpreter.


Change Number: XRAT/TC2/D4/0024 [751]

On Page: 3671 Line: 124921 Section: C.2.11 Signals and Error Handling

Add a new first paragraph to the section:

Historically, some shell implementations silently ignored attempts to use trap to set SIGINT or SIGQUIT to the default action or to set a trap for them after they have been set to be ignored by the shell when it executes an asynchronous subshell (and job control is disabled). This behavior is not conforming. For example, if a shell script containing the following line is run in the foreground at a terminal:

```bash
(trap - INT; exec sleep 10) & wait
```

and is then terminated by typing the interrupt character, this standard requires that the sleep command is terminated by the SIGINT signal.


This is what the specification requires.

Change Number: XRAT/TC2/D4/0025 [885]

On Page: 3679 Line: 125270,125274 Section: C.4.3 Exclusion of Utilities

In these two lines, change from:

base documents

to:

base documents for POSIX.2-1992

On Page: 3679 Line: 125281 Section: C.4.3 Exclusion of Utilities

Change from:

evaluated by the standard developers of the base documents
evaluated by the POSIX.2-1992 standard developers

On Page: 3680 Line: 125313 Section: C.4.3 Exclusion of Utilities

Change from:
not be included in the base documents because

to:
not be included in POSIX.2-1992 because

On Page: 3681 Line: 125351 Section: C.4.3 Exclusion of Utilities

Change from:
in a historical draft of the base documents

to:
in an early draft of the POSIX.2-1992 UPE


Change Number: XRAT/TC2/D4/0026 [885]

On Page: 3685 Line: 125420 Section: D.1 User Requirements

Change from:

as typified by the base documents listed in [xref to section A.1.1].

to:

as typified by the base documents for POSIX.1-1996.
