IEEE Std 1003.1™-2001/Cor 2-2004

The Open Group Technical Standard
Base Specifications, Issue 6

1003.1™-2001/Cor 2-2004

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

Technical Corrigendum 2

Sponsor

Portable Applications Standards Committee
of the
IEEE Computer Society

and

The Open Group

[This page intentionally left blank]

Abstract

Technical Corrigendum 2 addresses problems discovered since the approval of the 2003 edition of The Open Group Base Specifications, Issue 6, IEEE Std 1003.1, and ISO/IEC 9945. These are mainly due to resolving integration issues raised by the merger of the original Base documents.

Keywords

application program interface (API), argument, asynchronous, basic regular expression (BRE), batch job, batch system, built-in utility, byte, child, command language interpreter, CPU, extended regular expression (ERE), FIFO, file access control mechanism, input/output (I/O), job control, network, portable operating system interface (POSIX®), parent, shell, stream, string, synchronous, system, thread, X/Open System Interface (XSI)

IEEE Std 1003.1-2001 / Cor 2-2004

Published 2 April 2004 by the Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, U.S.A.
ISBN: 0-7381-4017-1 / SH95216 PDF 0-7381-3987-4 / SS95216
Printed in the United States of America by the IEEE.

Published 2 April 2004 by The Open Group
Apex Plaza, Forbury Road, Reading, Berkshire RG1 1AX, U.K.
Document Number: U059
Printed in the U.K. by The Open Group.

Feedback

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

IEEE

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.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property, or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "AS IS".

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of the IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration. 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, explanation, or interpretation of the IEEE. Current interpretations can be accessed at *http://standards.ieee.org/reading/ieee/interp/index.html*.

Errata, if any, for this and all other standards can be accessed at *http://standards.ieee.org/reading/ieee/updates/errata/index.html*. Users are encouraged to check this URL for errata periodically.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with the IEEE.[1] Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, U.S.A.

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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE Standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and non-discriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

Authorization to photocopy portions of any individual standard for internal or personal use is granted in the U.S. by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to the Copyright Clearance Center.[2] Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center. To arrange for payment of the licensing fee, please contact:

Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923, U.S.A., Tel.: +1 978 750 8400

Amendments, corrigenda, and interpretations for this standard, or information about the IEEE standards development process, may be found at *http://standards.ieee.org*.

The IEEE publications catalog and ordering information are available at *http://shop.ieee.org/store*.

1. For this standard, please send comments via the Austin Group as requested on page ii.
2. Please refer to the special provisions for this standard on page ii concerning permissions from both copyright holders and arrangements to cover photocopying and reproduction across the world, as well as by commercial organizations wishing to license the material for use in product documentation.

The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX certification.

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

The Open Group has over 15 years' experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification. The Open Group portfolio of test suites includes the *Westwood* family of tests for this standard and the associated certification program for Version 3 of the Single UNIX Specification, as well tests for CDE, CORBA, Motif, Linux, LDAP, POSIX.1, POSIX.2, POSIX Realtime, Sockets, UNIX, XPG4, XNFS, XTI, and X11. The Open Group test tools are essential for proper development and maintenance of standards-based products, ensuring conformance of products to industry-standard APIs, applications portability, and interoperability. In-depth testing identifies defects at the earliest possible point in the development cycle, saving costs in development and quality assurance.

More information is available at *www.opengroup.org/testing*.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards and Guides, but which also includes white papers, technical studies, branding and testing documentation, and business titles. Full details and a catalog are available at *www.opengroup.org/pubs*.

As with all *live* documents, Technical Standards and Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards compatible and those which are not:

- A new *Version* indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it *replaces* the previous publication.
- A new *Issue* indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Readers should note that Corrigenda may apply to any publication. Corrigenda information is published at *www.opengroup.org/corrigenda*.

The Open Group publications catalog and ordering information are available at *www.opengroup.org/pubs*.

Participants

IEEE Std 1003.1-2001/Cor 2-2004 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/WG15.

The Austin Group

At the time of approval, the membership of the Austin Group was as follows:

Andrew Josey, Chair
Donald W. Cragun, Organizational Representative, IEEE PASC
Nicholas Stoughton, Organizational Representative, ISO/IEC JTC 1/SC22/WG15
Mark Brown, Organizational Representative, The Open Group

Cathy Fox, Technical Editor

Austin Group Technical Reviewers

Jay Ashford
Julian Blake
Mitchell Bonnett
Andries Brouwer
Mark Brown
Paul Buerger
Dave Butenhof
Keith Chow
Geoff Clare
Donald W. Cragun
Lee Damico
Maulik Dave
Juan A. de la Puente
Guru Dutt Dhingra
Loic Domaigne
Ulrich Drepper
Sourav Dutta
Larry Dwyer
Paul Eggert
Joanna Farley
Clive D.W. Feather
Yaacov Fenster
Mark Funkenhauser
Ernesto Garcia
Andrew Gollan
Michael Gonzalez
Jean-Denis Gorin
Matthew Gream
Scott Gudgel
Bruno Haible
Charles Hammons
Barry Hedquist
Jon Hitchcock
Lowell G. Johnson
Andrew Josey
Piotr Karocki
David Leciston
Ryan Madron
Roger J. Martin
George Miao
Rajesh Moorkath
Peter Petrov
Franklin Prindle
Vikram Punj
Eusebio Rufian-Zilbermann
Joerg Schilling
Stephen Schwarm
Gil Shultz
Keld Simonsen
Nicholas Stoughton
Alexander Terekhov
Donn Terry
Mark-Rene Uchida
Thomas Unsicker
Scott Valcourt
Mats Wichmann
Garrett A. Wollman
Oren Yuen
Mark Ziegast

Austin Group Working Group Members

Harold C. Adams
Jay Ashford
Theodore P. Baker
David J. Blackwood
Julian Blake
Mitchell Bonnett
Andries Brouwer
Mark Brown
Paul Buerger
Alan Burns
Dave Butenhof
Keith Chow
Geoff Clare
Donald W. Cragun
Dragan Cvetkovic
Lee Damico
Maulik Dave
Juan A. de la Puente
Loic Domaigne
Steven J. Dovich
Ulrich Drepper
Guru Dutt Dhingra
Sourav Dutta
Larry Dwyer
Paul Eggert
Joanna Farley
Clive D.W. Feather
Yaacov Fenster
Mark Funkenhauser
Ernesto Garcia
Lois Goldthwaite
Andrew Gollan
Michael Gonzalez
Karen D. Gordon
Jean-Denis Gorin
Matthew Gream
Scott Gudgel
Joseph M. Gwinn
Steven A. Haaser
Charles Hammons
Ben Harris
Barry Hedquist
Karl Heubaum
Jon Hitchcock
Andreas Jaeger
Lowell G. Johnson
Andrew Josey
Piotr Karocki
Kenneth Lang
Pi-Cheng Law
David Leciston
Wojtek Lerch
Jonathan Lennox
Nick Maclaren
Ryan Madron
Roger J. Martin
George Miao
Rajesh Moorkath
Vilhelm Mueller
Joseph S. Myers
Peter Petrov
Franklin Prindle
Vikram Punj
Kenneth Raeburn
Tim Robbins
Curtis Royster Jr.
Eusebio Rufian-Zilbermann
Joerg Schilling
Stephen Schwarm
Glen Seeds
Gil Shultz
Keld Simonsen
Nicholas Stoughton
Alexander Terekhov
Donn Terry
Mark-Rene Uchida
Thomas Unsicker
Scott Valcourt
Mats Wichmann
Mike Wilson
Garrett A. Wollman
Oren Yuen
Mark Ziegast
Jason Zions

The Open Group

When The Open Group approved the Base Specifications, Issue 6, Technical Corrigendum 2 on 18 December 2003, the membership of The Open Group Base Working Group was as follows:

Andrew Josey, Chair
Mark Brown, Austin Group Liaison

Cathy Fox, Technical Editor

Base Working Group Members

Mark Brown	IBM Corporation
Dave Butenhof	Hewlett-Packard Company
Donald W. Cragun	Sun Microsystems, Inc.
Larry Dwyer	Hewlett-Packard Company
Ulrich Drepper	Red Hat, Inc.
Joanna Farley	Sun Microsystems, Inc.
Andrew Gollan	Sun Microsystems, Inc.
Andrew K. Roach	Sun Microsystems, Inc.
Curtis Royster Jr.	US DoD DISA
Nicholas Stoughton	USENIX Association
Kenjiro Tsuji	Sun Microsystems, Inc.

IEEE

When the IEEE Standards Board approved IEEE Std 1003.1-2001/Cor 2-2004 on 9 February 2004, the membership of the committees was as follows:

Portable Applications Standards Committee (PASC)

Lowell G. Johnson, Chair
Joseph M. Gwinn, Vice-Chair
Jay Ashford, Functional Chair
Andrew Josey, Functional Chair
Curtis Royster Jr., Functional Chair
Nicholas Stoughton, Secretary

Balloting Committee

The following members of the balloting committee voted on IEEE Std 1003.1-2001/Cor 2-2004. Balloters may have voted for approval, disapproval, or abstention:

Jay Ashford
Julian Blake
Mark Brown
Keith Chow
Donald W. Cragun
Juan A. de la Puente
Guru Dutt Dhingra
Ernesto Garcia
Michael Gonzalez
Jean-Denis Gorin
Matthew Gream
Scott Gudgel
Charles Hammons
Barry Hedquist
Andrew Josey
Piotr Karocki
David Leciston
Ryan Madron
Roger J. Martin
George Miao
Rajesh Moorkath
Peter Petrov
Vikram Punj
Eusebio Rufian-Zilbermann
Stephen Schwarm
Gil Shultz
Mark-Rene Uchida
Thomas Unsicker
Scott Valcourt
Oren Yuen

IEEE-SA Standards Board

When the IEEE-SA Standards Board approved IEEE Std 1003.1-2001/Cor 2-2004 on 9 February 2004, the membership was as follows:

Don Wright, Chair
Judith Gorman, Secretary

Chuck Adams	Raymond Hapeman	Paul Nikolich
H. Stephen Berger	Richard J. Holleman	T. W. Olsen
Mark D. Bowman	Richard H. Hulett	Ronald C. Petersen
Joseph A. Bruder	Lowell G. Johnson	Gary S. Robinson
Bob Davis	Hermann Koch	Frank Stone
Roberto de Boisson	Joseph L. Koepfinger*	Malcolm V. Thaden
Julian Forster*	Thomas J. McGean	Doug Topping
Arnold M. Greenspan	Steve M. Mills	Joe D. Watson
Mark S. Halpin	Daleep C. Mohla	

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

Satish K. Aggarwal, NRC Representative
Richard DeBlasio, DOE Representative
Alan Cookson, NIST Representative

Savoula Amanatidis, IEEE Standards Managing Editor

* Member Emeritus

[This page intentionally left blank]

1. Introduction

1.1 Scope

This technical corrigendum addresses issues raised in defect reports and interpretation requests submitted up to 14th August 2003, and 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.

2. Changes to Base Definitions

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

0001 Change Number: XBD/TC2/D6/1 [XBD ERN 20]

0002 On Page: xli   Line: "ISO/IEC 8859" Section: Referenced Documents (2003 Ed.)
0003 On Page: xxxix Line: "ISO/IEC 8859" Section: Referenced Documents (2001 Ed.)

0004 Add after line starting "Part 10":

0005 "Part 11: Latin/Thai Alphabet"

0006 Add after line starting "Part 15":

0007 "Part 16: Latin Alphabet No. 10"

0008 **Change Number: XBD/TC2/D6/2** [XSH ERN 4,XBD ERN 3]

0009 On Page: 25 Line: 932-935 Section: Conformance (2003 Ed.)
0010 On Page: 23 Line: 921-924 Section: Conformance (2001 Ed.)

0011 Change from:
0012 "If _POSIX_PRIORITIZED_IO is supported, then asynchronous I/O operations
0013 performed by aio_read(), aio_write(), and lio_listio() shall be submitted
0014 at a priority equal to the scheduling priority of the process minus
0015 aiocbp->aio_reqprio."

0016 To:
0017 "If _POSIX_PRIORITIZED_IO is supported, then asynchronous I/O operations
0018 performed by aio_read(), aio_write(), and lio_listio() shall be submitted
0019 at a priority equal to the scheduling priority equal to a base scheduling
0020 priority minus aiocbp->aio_reqprio. If Thread Priority Scheduling is
0021 not supported then the base scheduling priority is that of the calling
0022 process, otherwise the base scheduling priority is that of the calling
0023 thread."

0024 Rationale:
0025 The previous wording did not take threads into account.

Change Number: XBD/TC2/D6/3 [XBD ERN 4]

On Page: 48 Line: 1631-1635 Section: 3.93 Child Process (2003 Ed.)
On Page: 46 Line: 1619-1623 Section: 3.93 Child Process (2001 Ed.)

Change From:
"3.93 Child Process

A new process created (by fork(), posix_spawn(), or posix_spawnp())
by a given process. A child process remains the child of the creating
process as long as both processes continue to exist.

Note: The fork(), posix_spawn(), and posix_spawnp() functions
are defined in detail in the System Interfaces volume of IEEE Std
1003.1-2001."

To:
"3.93 Child Process

A new process created (by fork(), posix_spawn(), posix_spawnp(), or
vfork()) by a given process. A child process remains the child of the
creating process as long as both processes continue to exist.

Note: The fork(), posix_spawn(), posix_spawnp() and vfork() functions
are defined in detail in the System Interfaces volume of IEEE Std
1003.1-2001."

Change Number: XBD/TC2/D6/4 [XBD ERN 30]

On Page: 61 Line: 2005-2006 Section: 3.187 Group Database (2003 Ed.)
On Page: 59 Line: 1993-1994 Section: 3.187 Group Database (2001 Ed.)

Change From:
"A system database of implementation-defined format that contains at
least the following information for each group ID:"

To:
"A system database that contains at least the following information for
each group ID:"

Change Number: XBD/TC2/D6/5 [XBD ERN 7]

On Page: 76 Line: 2433,2440 Section: 3.295 Process Lifetime (2003 Ed.)
On Page: 74 Line: 2421,2428 Section: 3.295 Process Lifetime(2001 Ed.)

In the Section 3.295 Process Lifetime
Change From:
"After a process is created with a fork() function, it is considered
active."

To:
"After a process is created by fork(), posix_spawn(), posix_spawnp()
or vfork(), it is considered active."

Change From:
"Note: The fork(), wait(), waitid(), and waitpid() functions
are defined in detail in the System Interfaces volume of IEEE Std
1003.1-2001."

To:
"Note: The fork(), posix_spawn(), posix_spawnp(), vfork(), wait(),
waitid(), and waitpid () functions are defined in detail in the System
Interfaces volume of IEEE Std 1003.1-2001."

Change Number: XBD/TC2/D6/6 [XBD ERN 5]

On Page: 76 Line: 2448-2454 Section: 3.297 Process Termination (2003 Ed.)
On Page: 74 Line: 2436-2442 Section: 3.297 Process Termination (2001 Ed.)

Change From:
"There are two kinds of process termination:

1. Normal termination occurs by a return from main() or when requested with the exit() or _exit()functions.

2. Abnormal termination occurs when requested by the abort() function or when some signals are received.

Note: The _exit(), abort(), and exit() functions are defined in detail in the System Interfaces volume of IEEE Std 1003.1-2001."

To:
"There are two kinds of process termination:

1. Normal termination occurs by a return from main(), when requested with the exit(), _exit(), or _Exit() functions; or when the last thread in the process terminates by returning from its start function, by calling the pthread_exit() function, or through cancellation.

2. Abnormal termination occurs when requested by the abort() function or when some signals are received.

Note: The _exit(), _Exit(), abort(), and exit() functions are defined in detail in the System Interfaces volume of IEEE Std 1003.1-2001."

Rationale:
The definition does not mention the ''passive exit'' on termination of the last thread or the _Exit() function.

Change Number: XBD/TC2/D6/7 [XBD ERN 31,32]

On Page: 88 Line: 2778-2779 Section: 3.382 System Console (2003 Ed.)
On Page: 86 Line: 2766-2767 Section: 3.382 System Console (2001 Ed.)

Change From:
"An implementation-defined device that receives messages sent by the syslog() function, and the fmtmsg() function when the MM_CONSOLE flat is set."

To:
"A device that receives messages sent by the syslog() function, and the fmtmsg() function when the MM_CONSOLE flag is set."

Change Number: XBD/TC2/D6/8 [XBD ERN 33]

On Page: 88 Line: 2802-2803 Section: 3.385 System Process (2003 Ed.)
On Page: 86 Line: 2790-2791 Section: 3.385 System Process (2001 Ed.)

Change From:
"An implementation-defined object, other than a process executing an application, that has a process ID."

To:
"An object other than a process executing an application, that is provided by the system and has a process ID."

Change Number: XBD/TC2/D6/9 [TC2d5 ERN 4]

On Page: 88 Line: 2783-2796 Section: 3.383 System Databases (2003 Ed.)
On Page: 86 Line: 2771-2784 Section: 3.383 System Databases (2001 Ed.)

Change From:
"An implementation provides two system databases.

The "group database" contains the following information for each group:

1. Group name
2. Numerical group ID
3. List of all users allowed in the group

The "user database" contains the following information for each user:

1. User name
2. Numerical user ID
3. Numerical group ID
4. Initial working directory
5. Initial user program

If the initial user program field is null, the system default is used. If the initial working directory field is null, the interpretation of that field is implementation-defined. These databases may contain other fields that are unspecified by IEEE Std 1003.1-2001."

To:
"An implementation provides two system databases, the "group database" (see also Section 3.187) and the "user database" (see also Section 3.424)."

Change Number: XBD/TC2/D6/10 [TC2d5 ERN 5]

On Page: 89 Line: 2805-2806 Section: 3.386 System Reboot (2003 Ed.)
On Page: 87 Line: 2793-2794 Section: 3.386 System Reboot (2001 Ed.)

Change From:
"An implementation-defined sequence of events that may result in the loss of transitory data; that is, data that is not saved in permanent storage."

To:
"An unspecified sequence of events that may result in the loss of transitory data; that is, data that is not saved in permanent storage."

Change Number: XBD/TC2/D6/11 [XBD ERN 36]

On Page: 94 Line: 2939-2940 Section: 3.424 User Database (2003 Ed.)
On Page: 92 Line: 2927-2928 Section: 3.424 User Database (2001 Ed.)

Change From:
"A system database of implementation-defined format that contains at least the following information for each user ID:"

To:
"A system database that contains at least the following information for each user ID:"

Change Number: XBD/TC2/D6/12 [XBD ERN 28]

On Page: 104 Line: 3222-3223 Section: 4.14 (2003 Ed.)
On Page: 102 Line: 3207-3208 Section: 4.14 (2001 Ed.)

Change From:
"How any changes to the value of seconds since the Epoch
are made to align to a desired relationship with the current
actual time are made is implementation-defined."

To:
"How any changes to the value of seconds since the Epoch
are made to align to a desired relationship with the current
actual time is implementation-defined."

Rationale:
This is an editorial change.

Change Number: XBD/TC2/D6/13 [XBD ERN 17]

On Page: 119 Line: 3758 Section: 6.3 (2003 Ed.)
On Page: 117 Line: 3742 Section: 6.3 (2001 Ed.)

Add to the end of the paragraph:
"This standard provides no means of defining a wide character codeset."

Change Number: XBD/TC2/D6/14 [XBD ERN 17]

On Page: 121 Line: 3860-3862,3867-3868,3869 Section: 6.4 (2003 Ed.)
On Page: 119 Line: 3844-3846,3851-3852,3853 Section: 6.4 (2001 Ed.)

Change From:
"Bytes shall be treated as unsigned octets, and carry shall be propagated
between the bytes as necessary to represent the range. For example,
the line:"

To:
"Bytes shall be treated as unsigned octets, and carry shall be propagated
between the bytes as necessary to represent the range. However, because
this causes a null byte in the second or subsequent bytes of a character,
such a declaration should not be specified.For example, the line:"

Change From:
<j0103> \d130\d0
<j0104> \d130\d1

To:
<j0103> \d130\d00
<j0104> \d130\d01

Change From:
"The comment is optional."

To:
"The expanded declaration of the symbol <j0103> in the above
example is an invalid specification, because it contains a null
byte in the second byte of a character.

The comment is optional."

Add to the end of the paragraph:
"This standard provides no means of defining a wide character codeset."

Change Number: XBD/TC2/D6/15 [XBD ERN 24]

On Page: 121-122 Line: 3873-3885 Section: 6.4 (2003 Ed.)
On Page: 119-120 Line: 3857-3859 Section: 6.4 (2001 Ed.)

Change From:
"WIDTH An unsigned positive integer value defining the column width (see Section 3.103 (on page 49)) for the printable characters in the coded character set specified in Table 6-1 (on page 115) and Table 6-2 (on page 120)."

To:
"WIDTH A non-negative integer value defining the column width (see Section 3.103 (on page 49)) for the printable characters in the coded character set specified in Table 6-1 (on page 115) and Table 6-2 (on page 120)."

Change From:
"WIDTH_DEFAULT An unsigned positive integer value defining the default column width for any printable character not listed by one of the WIDTH keywords."

To:
"WIDTH_DEFAULT A non-negative integer value defining the default column width for any printable character not listed by one of the WIDTH keywords."

Rationale: This change allows the value zero for the width value of WIDTH and WIDTH_DEFAULT. This is required to cover some existing locales

Change Number: XBD/TC2/D6/16 [XBD ERN 25]

On Page: 143 Line: 4836-4839 Section: 7.3.3 (2003 Ed.)
On Page: 141 Line: 4820-4823 Section: 7.3.3 (2001 Ed.)

On page 143 lines 4836-4839 (2003 Ed.), page 141 lines 4820-4823 (2001 Ed.)
Change From:
"p_sep_by_space An integer set to 0 if no space separates the currency_symbol from the value for a monetary quantity with a non-negative value, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent."

To:
"p_sep_by_space Set to a value indicating the separation of the currency_symbol, the sign string, and the |
value for a non-negative formatted monetary quantity.

The values of p_sep_by_space, n_sep_by_space, int_p_sep_by_space, and int_n_sep_by_space are interpreted according to the following:

0 No space separates the currency symbol and value.

1 If the currency symbol and sign string are adjacent, a space separates them from the value; otherwise, a space separates the currency symbol from the value.

2 If the currency symbol and sign string are adjacent, a space separates them; otherwise, a space separates the sign string from the value."

On page 143 lines 4843-4846 (2003 Ed.), page 141 lines 4827-2830 (2001 Ed.)
Change From:
"n_sep_by_space An integer set to 0 if no space separates the currency_symbol from the value for a monetary quantity with a negative value, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent."

To:
"n_sep_by_space Set to a value indicating the separation of the currency_symbol, the sign string, and the value for a negative

formatted monetary quantity."

On page 144 lines 4864-4871 (2003 Ed.), page 142 lines 4848-4855 (2001 Ed.)
Change From:
"int_p_sep_by_space An integer set to 0 if no space separates the
int_curr_symbol from the value for a monetary quantity with a non-negative
value, set to 1 if a space separates the symbol from the value, and set
to 2 if a space separates the symbol and the sign string, if adjacent.

int_n_sep_by_space An integer set to 0 if no space separates the
int_curr_symbol from the value for a monetary quantity with a negative
value, set to 1 if a space separates the symbol from the value, and set
to 2 if a space separates the symbol and the sign string, if adjacent."

To:
"int_p_sep_by_space Set to a value indicating the separation
of the int_curr_symbol, the sign string, and the value for a
non-negative internationally formatted monetary quantity."

int_n_sep_by_space Set to a value indicating the separation of
the int_curr_symbol, the sign string, and the value for a negative
internationally formatted monetary quantity."

Rationale:
The descriptions of p_sep_by_space, n_sep_by_space, int_p_sep_by_space,
and int_n_sep_by_space need to be updated to match the description of
these keywords in C99 and System Interfaces Volume, localeconv().

Change Number: XBD/TC2/D6/17 [XBD ERN 37]

On Page: 185 Line: 6540-6541 Section: 10.2 (2003 Ed.)
On Page: 183 Line: 6508-6509 Section: 10.2 (2001 Ed.)

Change From:
"The implementation shall document which terminal types it supports and
which of these features and utilities are not supported by each terminal."

To:
"The implementation shall document in the system documentation which
terminal types it supports and which of these features and utilities
are not supported by each terminal."

Change Number: XBD/TC2/D6/18 [XSH ERN 146]

On Page: 224 Line: 7897-7899 Section: fcntl.h (2003 Ed.)
On Page: 222 Line: 7864-7865 Section: fcntl.h (2001 Ed.)

In the DESCRIPTION section
Change From:
"[ADV] int posix_fadvise(int, off_t, size_t, int);
       int posix_fallocate(int, off_t, size_t);"

To:
"[ADV] int posix_fadvise(int, off_t, off_t, int);
       int posix_fallocate(int, off_t, off_t);"

Rationale: The previous prototype was not large-file aware, and the standard
developers felt it acceptable to make this change before
implementations of the functions become widespread."

Change Number: XBD/TC2/D6/19 [XBD ERN 12]

On Page: 259-260 Line: 9132-9169 Section: limits.h (2003 Ed.)
On Page: 257-258 Line: 9093-9131 Section: limits.h (2001 Ed.)

On line 9132-9134 (2003 Ed.); line 9093-9095 (2001 Ed.)
Change From:
"{INT_MAX}
    Maximum value of an int.
    Minimum Acceptable Value:-2 147 483 647"

To:
"{INT_MAX}
    Maximum value of an int.
[CX]Minimum Acceptable Value:-2 147 483 647[/CX]"

On line 9156-9158 (2003 Ed.); line 9117-9119 (2001 Ed.)
Change From:
"{UINT_MAX}
    Maximum value of type unsigned.
    Minimum Acceptable Value: 4 294 967 295"

To:
"{UINT_MAX}
    Maximum value of type unsigned.
[CX]Minimum Acceptable Value: 4 294 967 295[/CX]"

On line 9165-9167 (2003 Ed.); line 9126-9128 (2001 Ed.)
Change From:

"[XSI] {WORD_BIT}
    Number of bits in a word or type int.
    Minimum Acceptable Value: 16[/XSI]"

To:
"[XSI] {WORD_BIT}
    Number of bits in a type int.
    Minimum Acceptable Value: 32[/XSI]"

On Line 9168-9170 (2003 Ed.); line 9129-9131 (2001 Ed.)
Change From:
"{INT_MIN}
    Minimum value of type int.
    Maximum Acceptable Value: 2 147 483 647"

To:
"{INT_MIN}
    Minimum value of type int.
[CX]Maximum Acceptable Value: 2 147 483 647[/CX]"

Rationale: There was conflicting information about the size
of an integer.

Change Number: XBD/TC2/D6/20 [XBD ERN 26]

On Page: 260 Line: 9191-9193 Section: limits.h (2003 Ed.)
On Page: 258 Line: 9152-9154 Section: limits.h (2001 Ed.)

In the DESCRIPTION section
Under "Other Invariant Values"

Remove the lines:
"XSI {CHARCLASS_NAME_MAX}
     Maximum number of bytes in a character class name.
     Minimum Acceptable Value: 14"

Rationale:
CHARCLASS_NAME_MAX was defined under "Runtime Increasable Values"
and also "Other Invarient Values". This change corrects
an integration issue with the base specifications.

Change Number: XBD/TC2/D6/21 [XBD ERN 21]

On Page: 266 Line: 9440-9447 Section: math.h (2003 Ed.)
On Page: 264 Line: 9399-9406 Section: math.h (2001 Ed.)

Change From:
"The following optional macros indicate whether the fma() family of functions are fast compared with direct code:

FP_FAST_FMA
FP_FAST_FMAF
FP_FAST_FMAL

The FP_FAST_FMA macro shall be defined to indicate that the fma() function generally executes about as fast as, or faster than, a multiply and an add of double operands. The other macros have the equivalent meaning for the float and long double versions."

To:
"The following optional macros indicate whether the fma() family of functions are fast compared with direct code:

FP_FAST_FMA
FP_FAST_FMAF
FP_FAST_FMAL

If defined, the FP_FAST_FMA macro shall indicate that the fma() function generally executes about as fast as, or faster than, a multiply and an add of double operands. If undefined it is unspecified what the speed of execution is. The other macros have the equivalent meaning for the float and long double versions."

Change Number: XBD/TC2/D6/22 [XBD ERN 15]

On Page: 281 Line: 10000 Section: netinet/in.h (2003 Ed.)
On Page: 279 Line: 9953 Section: netinet/in.h (2001 Ed.)

Change From:
"The <netinet/in.h> header shall define the sockaddr_in structure that includes at least the following members (all in network byte order):"

To:
"The <netinet/in.h> header shall define the sockaddr_in structure that includes at least the following members listed below:"

Add after line 10004 (2003 Ed.); line 9957 (2001 Ed.):
"The sin_port and sin_addr members shall be in network byte order."

On line 10011 (2003 Ed.); line 9964 (2001 Ed.).
Change From:
"The <netinet/in.h> header shall define the sockaddr_in6 structure that includes at least the following members (all in network byte order):"

To:
"The <netinet/in.h> header shall define the sockaddr_in6 structure that includes at least the following members:"

Add after line 10017 (2003 Ed.); line 9970 (2001 Ed.):
"The sin6_port and sin6_addr members shall be in network byte order."

Change Number: XBD/TC2/D6/23 [XBD ERN 8]

On Page: 297 Line: 10588,10600 Section: <sched.h> (2003 Ed.)
On Page: 295 Line: 10537,10549 Section: <sched.h> (2001 Ed.)

In the DESCRIPTION section
Change From:
"int sched_priority Process execution scheduling priority."

To:
"int sched_priority Process [THR]or thread[/THR] execution scheduling priority."

Change From:
"Each process is controlled by an associated scheduling policy and priority."

To:
"Each process [THR]or thread[/THR] is controlled by an associated scheduling policy and priority."

Change Number: XBD/TC2/D6/24 [XBD ERN 18]

On Page: 306 Line: 10907-10918 Section: signal.h (2003 Ed.)
On Page: 304 Line: 10852-10863 Section: signal.h (2001 Ed.)

Change From:
"CX The <signal.h> header shall define the siginfo_t type as a structure that includes at least the following members:

```
[CX]  int si_signo           Signal number.
[XSI] int si_errno           If non-zero, an errno value associated with
                             this signal, as defined in <errno.h>.
[CX]  int si_code            Signal code.
[XSI] pid_t si_pid           Sending process ID.
      uid_t si_uid           Real user ID of sending process.
      void *si_addr          Address of faulting instruction.
      int si_status          Exit value or signal.
      long si_band           Band event for SIGPOLL.
[RTS] union sigval si_value  Signal value. "
```

To:
"[CX] The <signal.h> header shall define the siginfo_t type as a structure that includes at least the following members:

```
[CX]  int si_signo           Signal number.
      int si_code            Signal code.
[XSI] int si_errno           If non-zero, an errno value associated with
                             this signal, as defined in <errno.h>.
      pid_t si_pid           Sending process ID.
      uid_t si_uid           Real user ID of sending process.
      void *si_addr          Address of faulting instruction.
      int si_status          Exit value or signal.
      long si_band           Band event for SIGPOLL.
[RTS] union sigval si_value  Signal value. "
```

Rationale: This is an editorial change to the shading and no normative change is intended.

Change Number: XBD/TC2/D6/25 [XBD ERN 29]

On Page: 358 Line: 12708 Section: <sys/stat.h> (2003 Ed.)
On Page: 356 Line: 12643 Section: <sys/stat.h> (2001 Ed.)

Change From:
"Times shall be given in seconds since the Epoch."

To:
"The timespec structure may be defined as described in <time.h>.
Times shall be given in seconds since the Epoch."

Change Number: XBD/TC2/D6/26 [XBD ERN 9]

On Page: 369 Line: 13064 Section: sys/types.h (2003 Ed.)
On Page: 367 Line: 12988 Section: sys/types.h (2001 Ed.)

Add "pthread_t" to the list of types in lines 13052-13068 (2003 Ed.)
and lines 12976-12992 (2001 Ed.) with it margin marked as THR and shaded.

Rationale:The intent is to allow pthread_t to be a structure.

Change Number: XBD/TC2/D6/27 [XBD ERN 38]

On Page: 373 Line: 13189-13190 Section: sys/utsname.h (2003 Ed.)
On Page: 371 Line: 13113-13114 Section: sys/utsname.h (2001 Ed.)

Change From:
"char nodename[] Name of this node within an implementation-defined
communications network. "

To:
"char nodename[] Name of this node within the communications
network to which this node is attached,if any."

Change Number: XBD/TC2/D6/28 [XSH ERN 53]

On Page: 398 Line: 14041-14044 Section: ucontext.h (2003 Ed.)
On Page: 396 Line: 13960-13963 Section: ucontext.h (2001 Ed.)

Margin mark OB (obsolescent) and shade the getcontext(), makecontext(),
setcontext(), swapcontext() functions.

Change Number: XBD/TC2/D6/29 [XBD ERN 16]

On Page: 401-404 Line: 14130-14133 Section: unistd.h (2003 Ed.)
On Page: 399-403 Line: 14049-14052 Section: unistd.h (2001 Ed.)

Change From:
"_POSIX_CHOWN_RESTRICTED
The use of chown() and fchown() is restricted to a process with
appropriate privileges, and to changing the group ID of a file only to
the effective group ID of the process or to one of its supplementary
group IDs. This symbol shall always be set to a value other than -1."

To:
"_POSIX_CHOWN_RESTRICTED
The use of chown() and fchown() is restricted to a process with
appropriate privileges, and to changing the group ID of a file only to
the effective group ID of the process or to one of its supplementary
group IDs. This symbol shall either be undefined or defined with a value
other than -1."

On lines 14167-14169 (2003 Ed.); lines 14083-14085 (2001 Ed.)
Change from:
"_POSIX_NO_TRUNC
Pathname components longer than {NAME_MAX} generate an
error. This symbol shall always be set to a value other than -1."

To:
"_POSIX_NO_TRUNC
Pathname components longer than {NAME_MAX} generate an error. This symbol
shall either be undefined or defined with a value other than -1."

For the Constants for Options and Option Groups on pages 401-406 lines
14121-14350 (2003 Ed.); pages 399-404 lines 14040-14266 (2001 Ed.):

Change occurrences of the text From:
"If this symbol has a value other than -1 or 0, it shall have the
value 200112L."

To:
"If this symbol is defined in <unistd.h>, it shall be defined to be -1,
0, or 200112L. The value of this symbol reported by sysconf() shall
either be -1 or 200112L."

Rationale:
This further clarifies the requirements on when constants in unistd.h
can be defined or undefined.

Change Number: XBD/TC2/D6/30 [XSH ERN 106]

On Page: 405 Line: 14309-14320 Section: unistd.h (2003 Ed.)
On Page: 403 Line: 14225-14236 Section: unistd.h (2003 Ed.)

In the DESCRIPTION section, line 14309 (2003 Ed.); line 14225 (2001 Ed.)
Change From:
"_V6_ILP32_OFF32"

To:
"_POSIX_V6_ILP32_OFF32"

In the DESCRIPTION section, line 14312 (2003 Ed.); line 14228 (2001 Ed.)
Change From:
"_V6_ILP32_OFFBIG"

To:
"_POSIX_V6_ILP32_OFFBIG"

In the DESCRIPTION section, line 14315 (2003 Ed.); line 14231 (2001 Ed.)
Change From:
"_V6_LP64_OFF64"

To:
"_POSIX_V6_LP64_OFF64"

In the DESCRIPTION section, line 14318 (2003 Ed.); line 14234 (2001 Ed.)
Change From:
"_V6_LPBIG_OFFBIG"

To:
"_POSIX_V6_LPBIG_OFFBIG"

Rationale: Consistency with the sysconf() and c99 man pages.

Change Number: XBD/TC2/D6/31 [XCU ERN 19]

On Page: 408 Line: 14437-14440 Section: unistd.h (2003 Ed.)
On Page: 406 Line: 14350-14353 Section: unistd.h (2001 Ed.)

Change From:
"This value is a <newline>-separated list of names of programming environments supported by the implementation in which the widths of the blksize_t, cc_t, mode_t, nfds_t, pid_t, ptrdiff_t, size_t, speed_t, ssize_t, suseconds_t, tcflag_t, useconds_t, wchar_t, and wint_t types are no greater than the width of type long."

To:
"This value is a <newline>-separated list of names of programming environments supported by the implementation in which the widths of the blksize_t, cc_t, mode_t, nfds_t, pid_t, ptrdiff_t, size_t, speed_t, ssize_t, suseconds_t, tcflag_t, useconds_t, wchar_t, and wint_t types are no greater than the width of type long. The format of each name shall be suitable for use with the getconf -v option."

Change Number: XBD/TC2/D6/32 [XSH ERN 40,61,62]

On Page: 409-411 Line: 14525,14492,14610 Section: unistd.h (2003 Ed.)
On Page: 407-409 Line: 14438,14405,14520 Section: unistd.h (2001 Ed.)

Delete "_SC_FILE_LOCKING"

Delete "_SC_2_C_VERSION"

Delete "_SC_XOPEN_XCU_VERSION"

Change Number: XBD/TC2/D6/33 [XBD ERN 17]

On Page: 410 Line: 14564,14586,14588 Section: unistd.h (2003 Ed.)
On Page: 408 Line: 14475,14496,14498 Section: unistd.h (2001 Ed.)

Add to the list of symbolic constants for sysconf()

_SC_SS_REPL_MAX, _SC_TRACE_EVENT_NAME_MAX, _SC_TRACE_NAME_MAX,
_SC_TRACE_SYS_MAX _SC_TRACE_USER_EVENT_MAX

Rationale: These symbols were omitted.

Change Number: XBD/TC2/D6/34 [XBD ERN 10]

On Page: 413 Line: 14692 Section: unistd.h (2003 Ed.)
On Page: 411 Line: 14602 Section: unistd.h (2001 Ed.)

Remove the XSI margin marker and remove the shading
from the following function:
"int symlink(const char *, const char *);"

Rationale: This was inconsistent with the XSH volume.

Change Number: XBD/TC2/D6/35 [XBD ERN 11]

On Page: 413 Line: 14471 Section: unistd.h (2003 Ed.)
On Page: 404 Line: 14384 Section: unistd.h (2001 Ed.)

Insert at the start of the symbolic constants list
for pathconf() (before _PC_ALLOC_SIZE_MIN):
"_PC_2_SYMLINKS"

Rationale: This corresponds to the definition of POSIX2_SYMLINKS in
the Shell and Utilities volume.

3. Changes to System Interfaces

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

Change Number: XSH/TC2/D6/1 [XBD ERN 20]

On Page: xxx Line: "ISO/IEC 8859" Section: Referenced Documents (2003 Ed.)
On Page: xxvi Line: "ISO/IEC 8859" Section: Referenced Documents (2001 Ed.)

Add after line starting "Part 10":

"Part 11: Latin/Thai Alphabet"

Add after line starting "Part 15":

"Part 16: Latin Alphabet No. 10"

Change Number: XSH/TC2/D6/2 [XSH ERN 167,TC2d5 ERN 9]

On Page: 15,19 Line: 580-583,713-714 Section: 2.2.2 (2003 Ed.)
On Page: 15,19 Line: 571-574,699-700 Section: 2.2.2 (2001 Ed.)

On lines 580-583 (2003 Ed.); lines 571-574 (2001 Ed.)
Change From:
"Implementations may add symbols to the headers shown in the following table, provided the identifiers for those symbols begin with the corresponding reserved prefixes in the following table, and do not use the reserved prefixes posix_, POSIX_, or _POSIX_."

To:
"Implementations may add symbols to the headers shown in the following table, provided the identifiers for those symbols either:

1. begin with one of the corresponding reserved prefixes in the table; or

2. have one of the corresponding complete names in the table; or

3. end in the string indicated as a reserved suffix in the table and do not use the reserved prefixes posix_, POSIX_, or _POSIX_, as long as the reserved suffix is in that part of the name considered significant by the implementation.

Symbols that use the reserved prefix _POSIX_ may be made visible by implementations in any header defined by IEEE Std 1003.1-2001."

On lines 713-714 (2003 Ed.); lines 699-700 (2001 Ed.):
Change From:
"1. All identifiers that begin with an underscore and either an uppercase letter or another underscore are always reserved for any use by the implementation."

To:
"1. With the exception of identifiers beginning with the prefix _POSIX_, all identifiers that begin with an underscore and either an uppercase letter or another underscore are always reserved for any use by the implementation."

Rationale:
This change permits implementations to have symbols with the prefix _POSIX_ visible in any header. A change in TC1 had disallowed this.

Change Number: XSH/TC2/D6/3 [XSH ERN 1]

On Page: 18 Line: 668 Section: 2.2.2 (2003 Ed.)
On Page: 18 Line: 659 Section: 2.2.2 (2001 Ed.)

Add the following line into the table after line 668 (2003 Ed.);
after line 659 (2001 Ed.):
"<math.h> FP_[A-Z]"

Rationale:
This text is added for consistency with the math.h page in the
Base Definitions volume which states "Additional implementation-defined
floating-point classifications, with macro definitions beginning with FP_
and an uppercase letter, may also be specified by the implementation."

Change Number: XSH/TC2/D6/4 [XSH ERN 99]

On Page: 33 Line: 1337 Section: 2.4.3 (2003 Ed.)
On Page: 33 Line: 1339 Section: 2.4.3 (2001 Ed.)

Add "sockatmark()" to the table in 2.4.3 (the list of functions that shall
be either reentrant or non-interruptible by signals and shall be
async-signal-safe).

Rationale:
This was an omission when the sockatmark() function was added.

Change Number: XSH/TC2/D6/5 [XSH ERN 109]

On Page: 55 Line: 2267 Section: 2.9.5.2 (2003 Ed.)
On Page: 55 Line: 2268 Section: 2.9.5.2 (2001 Ed.)

Add "fdatasync()" into the table of functions after "fcntl()".

Rationale:
This was an omission. Since fdatasync() is often implemented
in terms of fsync() it should also be in this list.

Change Number: XSH/TC2/D6/6 [XSH ERN 76]

On Page: 56 Line: 2277 Section: 2.9.5.2 Cancellation Points (2003 Ed.)
On Page: 56 Line: 2278 Section: 2.9.5.2 Cancellation Points (2001 Ed.)

Insert into the list of functions that "may have" a cancellation point (the second list of functions in this section) in the appropriate order:

"access
asctime
asctime_r
ctime
ctime_r
fmtmsg
fpathconf
fstat
getaddrinfo
gethostid
getnameinfo
getopt [with footnote if opterr is nonzero]
link
localtime
localtime_r
lstat
mktime
pathconf
posix_openpt
stat
strerror_r
strftime
symlink
sync
tzset
wcsftime
wordexp"

Change Number: XSH/TC2/D6/7 [XSH ERN 77]

On Page: 57 Line: 2336-2353 Section: 2.9.5.3 (2003 Ed.)
On Page: 57 Line: 2337-2354 Section: 2.9.5.3 (2001 Ed.)

Change From:
"Each thread maintains a list of cancellation cleanup handlers. The programmer uses the pthread_cleanup_push() and pthread_cleanup_pop() functions to place routines on and remove routines from this list.

When a cancellation request is acted upon, the routines in the list are invoked one by one in LIFO sequence; that is, the last routine pushed onto the list (Last In) is the first to be invoked (First Out). The thread invokes the cancellation cleanup handler with cancellation disabled until the last cancellation cleanup handler returns. When the cancellation cleanup handler for a scope is invoked, the storage for that scope remains valid. If the last cancellation cleanup handler returns, thread execution is terminated and a status of PTHREAD_CANCELED is made available to any threads joining with the target. The symbolic constant PTHREAD_CANCELED expands to a constant expression of type (void *) whose value matches no pointer to an object in memory nor the value NULL.

The cancellation cleanup handlers are also invoked when the thread calls pthread_exit().

A side effect of acting upon a cancellation request while in a condition variable wait is that the mutex is re-acquired before calling the first cancellation cleanup handler. In addition, the thread is no longer considered to be waiting for the condition and the thread shall not have consumed any pending condition signals on the condition.

A cancellation cleanup handler cannot exit via longjmp() or siglongjmp()."

To:
Each thread maintains a list of cancellation cleanup handlers. The programmer uses the pthread_cleanup_push() and pthread_cleanup_pop() functions to place routines on and remove routines from this list.

When a cancellation request is acted upon, or when a thread calls pthread_exit(), the the thread first disables cancellation by setting its cancelability state to PTHREAD_CANCEL_DISABLE and its cancelability type to PTHREAD_CANCEL_DEFERRED. The cancelability state shall remain set to PTHREAD_CANCEL_DISABLE until the thread has terminated. The behavior is undefined if a cancellation cleanup handler or thread-specific data destructor routine changes the cancelability state to PTHREAD_CANCEL_ENABLE.

The routines in the thread's list of cancellation cleanup handlers are invoked one by one in LIFO sequence; that is, the last routine pushed onto the list (Last In) is the first to be invoked (First Out). When the cancellation cleanup handler for a scope is invoked, the storage for that scope remains valid. If the last cancellation cleanup handler returns, thread-specific data destructors (if any) associated with thread-specific data keys for which the thread has non-NULL values will be run, in unspecified order, as described for pthread_key_create().

After all cancellation cleanup handlers and thread-specific data destructors have returned, thread execution is terminated. If the thread has terminated because of a call to pthread_exit(), the value_ptr argument is made available to any threads joining with the target. If the thread has terminated by acting on a cancellation request, a status of PTHREAD_CANCELED is made available to any threads joining with the target. The symbolic constant PTHREAD_CANCELED expands to a constant expression of type (void *) whose value matches no pointer to an object in memory nor the value NULL.

A side effect of acting upon a cancellation request while in a condition variable wait is that the mutex is re-acquired before calling the first cancellation cleanup handler. In addition, the thread is no longer considered to be waiting for the condition and the thread shall not have consumed any pending condition signals on the condition.

A cancellation cleanup handler cannot exit via longjmp() or siglongjmp()."

Change Number: XSH/TC2/D6/8 [XSH ERN 89]

On Page: 58 Line: 2379 Section: Threads (2003 Ed.)
On Page: 58 Line: 2380 Section: Threads (2001 Ed.)

Insert new section:
"2.9.8 Use of Application-Managed Thread Stacks

An "application-managed thread stack" is a region of memory allocated by the application, for example, memory returned by the malloc() or mmap() functions, and designated as a stack through the act of passing an address related to that memory as the "stackaddr" argument to pthread_attr_setstackaddr() (obsolete) or by passing the address and size of the stack, respectively, as the stackaddr and stacksize arguments to pthread_attr_setstack(). Application-managed stacks allow the application to precisely control the placement and size of a stack.

The application grants to the implementation permanent ownership of and control over the application-managed stack when the attributes object in which the stack or stackaddr attribute has been set is used, either by presenting that attributes object as the 'attr' argument in a call to pthread_create() that completes successfully, or by storing a pointer to

the attributes object in the sigev_notify_attributes member of a 'struct sigevent' and passing that 'struct sigevent' to a function accepting such argument that completes successfully. The application may thereafter utilize the memory within the stack only within the normal context of stack usage within or properly synchronized with a thread that has been scheduled by the implementation with stack pointer value(s) that are within the range of that stack. In particular, the region of memory cannot be freed, nor can it be later specified as the stack for another thread.

When specifying an attributes object with an application-managed stack through the sigev_notify_attributes member of a 'struct sigevent', the results are undefined if the requested signal is generated multiple times (as for a repeating timer).

Until an attributes object in which the stack or stackaddr attribute has been set is used, the application retains ownership of and control over the memory allocated to the stack. It may free or reuse the memory as long as it either deletes the attributes object, or before using the attributes object replaces the stack by making an additional call to the same function, either pthread_attr_setstackaddr() or pthread_attr_setstack(), that was used originally to designate the stack. There is no mechanism to retract the reference to an application-managed stack by an existing attributes object.

Once an attributes object with an application-managed stack has been used, that attributes object cannot be used again by a subsequent call to pthread_create() or any function accepting a 'struct sigevent' with sigev_notify_attributes containing a pointer to the attributes object, without designating an unused application-managed stack by making an additional call to the function originally used to define the stack, pthread_attr_setstack() or pthread_attr_setstackaddr()."

Change Number: XSH/TC2/D6/9 [XSH ERN 138]

On Page: 92 Line: 3558-3559 Section: abort (2003 Ed.)
On Page: 92 Line: 3559-3560 Section: abort (2001 Ed.)

In the APPLICATION USAGE section
Change From:
"Catching the signal is intended to provide the application writer with a portable means to abort processing, free from possible interference from any implementation-defined functions."

To:
"Catching the signal is intended to provide the application writer with a portable means to abort processing, free from possible interference from any implementation-supplied functions."

Change Number: XSH/TC2/D6/10 [XSH ERN 2]

On Page: 105 Line: 3919 Section: aio_cancel (2003 Ed.)
On Page: 104 Line: 3898 Section: aio_cancel (2001 Ed.)

In the RETURN VALUE section
Change From:
"The aio_cancel() function shall return the value AIO_CANCELED to the calling process if the requested operation(s) were canceled."

To:
"The aio_cancel() function shall return the value AIO_CANCELED if the requested operation(s) were canceled."

Rationale:
The use of the term "to the calling process" was unnecessary and incorrect.

Change Number: XSH/TC2/D6/11 [XSH ERN 3]

On Page: 108 Line: 4022 Section: aio_fsync (2003 Ed.)
On Page: 107 Line: 4001 Section: aio_fsync (2001 Ed.)

In the RETURN VALUE section
Change From:
"The aio_fsync() function shall return the value 0 to the calling process if the I/O operation is successfully queued;"

To:
"The aio_fsync() function shall return the value 0 if the I/O operation is successfully queued;"

Rationale:
The use of the term "to the calling process" was unnecessary and incorrect.

Change Number: XSH/TC2/D6/12 [XSH ERN 4,5]

On Page: 110 Line: 4065,4088 Section: aio_read (2003 Ed.)
On Page: 109 Line: 4044,4067 Section: aio_read (2001 Ed.)

In the DESCRIPTION section
Change From:
"If prioritized I/O is supported for this file, then the asynchronous operation shall be submitted at a priority equal to the scheduling priority of the process minus aiocbp->aio_reqprio."

To:
"[PIO] If prioritized I/O is supported for this file, then the asynchronous operation shall be submitted at a priority equal to a base scheduling priority minus aiocbp->reqprio. If Thread Priority Scheduling is not supported then the base scheduling priority is that of the calling process, [PIO TPS] otherwise the base scheduling priority is that of the calling thread. [/end shading]"

Rationale:
The previous wording did not take threads into account.

In the RETURN VALUE section
Change From:
"The aio_read() function shall return the value zero to the calling process if the I/O operation is successfully queued;"

To:
"The aio_read() function shall return the value zero if the I/O operation is successfully queued;"

Rationale:
The use of the term "to the calling process" was unnecessary and incorrect.

Change Number: XSH/TC2/D6/13 [XSH ERN 83]

On Page: 111 Line: 4101-4103 Section: aio_read (2003 Ed.)
On Page: 110 Line: 4080-4082 Section: aio_read (2001 Ed.)

In the ERRORS section
Change From:
"[EINVAL] The file offset value implied by aiocbp->aio_offset would be invalid, aiocbp->aio_reqprio is not a valid value, or aiocbp->aio_nbytes is an invalid value."

To:
"[EINVAL] The file offset value implied by aiocbp->aio_offset would be invalid, [PIO]aiocbp->aio_reqprio is not a valid value[/PIO], or aiocbp->aio_nbytes is an invalid value."

Rationale:
Detection of an [EINVAL] error for a invalid value of aiocbp->aio_reqprio should only be required if _POSIX_PRIORITIZED_IO is supported.

Change Number: XSH/TC2/D6/14 [XSH ERN 6,7]

On Page: 116 Line: 4252,4276 Section: aio_write (2003 Ed.)
On Page: 115 Line: 4231,4255 Section: aio_write (2001 Ed.)

In the DESCRIPTION section
Change From:
"If prioritized I/O is supported for this file, then the asynchronous operation shall be submitted at a priority equal to the scheduling priority of the process minus aiocbp->aio_reqprio."

To:
"[PIO] If prioritized I/O is supported for this file, then the asynchronous operation shall be submitted at a priority equal to a base scheduling priority minus aiocbp->reqprio. If Thread Priority Scheduling is not supported then the base scheduling priority is that of the calling process, [PIO TPS] otherwise the base scheduling priority is that of the calling thread. [/end shading]"

Rationale:
The previous wording did not take threads into account.

In the RETURN VALUE section
Change From:
"The aio_write () function shall return the value zero to the calling process if the I/O operation is successfully queued;"

To:
"The aio_write () function shall return the value zero if the I/O operation is successfully queued;"

Rationale:
The use of the term "to the calling process" was unnecessary and incorrect.

Change Number: XSH/TC2/D6/15 [XSH ERN 83]

On Page: 117 Line: 4290-4292 Section: aio_write (2003 Ed.)
On Page: 116 Line: 4269-4271 Section: aio_write (2001 Ed.)

In the ERRORS section
Change From:
"[EINVAL] The file offset value implied by aiocbp->aio_offset would be invalid, aiocbp->aio_reqprio is not a valid value, or aiocbp->aio_nbytes is an invalid value."

To:
"[EINVAL] The file offset value implied by aiocbp->aio_offset would be invalid, [PIO]aiocbp->aio_reqprio is not a valid value[/PIO], or aiocbp->aio_nbytes is an invalid value."

Rationale:
Detection of an [EINVAL] error for a invalid value of aiocbp->aio_reqprio should only be required if _POSIX_PRIORITIZED_IO is supported.

Change Number: XSH/TC2/D6/16 [XSH ERN 139]

On Page: 119 Line: 4375-4376 Section: alarm (2003 Ed.)
On Page: 118 Line: 4353-4354 Section: alarm (2001 Ed.)

In the RATIONALE section
Change From:
"In some implementations, including 4.3 BSD, very large values of the seconds argument are silently rounded down to an implementation-defined maximum value"

To:
"In some implementations, including 4.3 BSD, very large values of the seconds argument are silently rounded down to an implementation-specific maximum value"

Change Number: XSH/TC2/D6/17 [XSH ERN 115]

On Page: 122 Line: 4440 Section: asctime (2003 Ed.)
On Page: 121 Line: 4419 Section: asctime (2001 Ed.)

In the RETURN VALUE section
Change From:
"Upon successful completion, asctime() shall return a pointer to the string."

To:
"Upon successful completion, asctime() shall return a pointer to the string. [CX] If the function is unsuccessful, it shall return NULL. [/CX]"

Change Number: XSH/TC2/D6/18 [XSH ERN 123]

On Page: 132 Line: 4738 Section: atan2 (2003 Ed.)
On Page: 131 Line: 4717 Section: atan2 (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Converting Cartesian to Polar coordinates system

The function below uses atan2() to convert a 2d vector expressed
in cartesian coordinates (x,y) to the polar coordinates (rho,theta).
There are other ways to compute the angle theta, using
asin(), acos() or atan(). However, atan2() presents here two
advantages:
    - the angle's quadrant is automatically determined.
    - the singular cases (0,y) are taken into account.
Finally, this example uses hypot() rather than sqrt() since
it is better for special cases, see hypot() for
more information.

```
#include <math.h>

void
cartesian_to_polar(const double x, const double y,
                   double *rho, double *theta
                   )
{
  *rho   = hypot (x,y); /* better than sqrt(x*x+y*y) */
  *theta = atan2 (y,x);
}"
```

Change Number: XSH/TC2/D6/19 [XSH ERN 8]

On Page: 137 Line: 4876 Section: atexit (2003 Ed.)
On Page: 136 Line: 4855 Section: atexit (2001 Ed.)

Add to end of the APPLICATION USAGE section:
"Since the behavior is undefined if the exit() function is called
more than once, portable applications calling atexit() must ensure
that the exit() function is not called at normal process termination
when all functions registered by the atexit() function are called.

All functions registered by the atexit() function are called at
normal process termination, which occurs by a call to the exit()
function or a return from main() or on the last thread termination,
when the behavior is as if the implementation called exit() with a
zero argument at thread termination time.

If, at normal process termination, a function registered by the
atexit() function is called and a portable application needs to stop
further exit() processing, it must call the _exit() function or
the _Exit() function or one of the functions which cause abnormal
process termination."

Rationale:
Additional clarifications for application usage are added.

Change Number: XSH/TC2/D6/20 [XSH ERN 140]

On Page: 142 Line: 5007-5009 Section: basename (2003 Ed.)
On Page: 141 Line: 4986-4988 Section: basename (2001 Ed.)

In the DESCRIPTION section
Change From:
"If the string consists entirely of the '/' character,
basename() shall return a pointer to the string "/". If the string
is exactly "//", it is implementation-defined whether '/'
or "//" is returned."

To:
"If the string pointed to by path consists entirely of the '/'
character, basename() shall return a pointer to the string "/". If the
string pointed to by path is exactly "//", it is implementation-defined
whether "/" or "//" is returned."

Change Number: XSH/TC2/D6/21 [XSH ERN 138]

On Page: 259 Line: 8533-8534 Section: dlopen (2003 Ed.)
On Page: 257 Line: 8488-8489 Section: dlopen (2001 Ed.)

In the DESCRIPTION section
Change From:
"If neither RTLD_GLOBAL nor RTLD_LOCAL are specified, then an
implementation-defined default behavior shall be applied."

To:
"If neither RTLD_GLOBAL nor RTLD_LOCAL are specified, then
the default behavior is unspecified."

Change Number: XSH/TC2/D6/22 [XSH ERN 120]

On Page: 289 Line: 9418 Section: erf (2003 Ed.)
On Page: 287 Line: 9360 Section: erf (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Computing the probability for a normal variate

This example shows how to use erf() to compute the probability
that a normal variate assumes a value in the range [x1,x2] with x1<=x2.
This example uses the constant M_SQRT1_2 which is an XSI extension.

```
 #include <math.h>

 double
 Phi(const double x1, const double x2)
 {
   return ( erf(x2*M_SQRT1_2) - erf(x1*M_SQRT1_2) ) / 2;
 }"
```

Change Number: XSH/TC2/D6/23 [XSH ERN 132]

On Page: 294 Line: 9526 Section: errno (2003 Ed.)
On Page: 292 Line: 9468 Section: errno (2001 Ed.)

In the DESCRIPTION section
Add at the end of the second paragraph:
"The setting of errno after a successful call to a function is unspecified unless the description of that function specifies that errno shall not be modified".

Change Number: XSH/TC2/D6/24 [XSH ERN 45,46,63]

On Page: 297-299 Line: 9637 Section: exec (2003 Ed.)
On Page: 295-297 Line: 9574 Section: exec (2001 Ed.)

In the DESCRIPTION section
On line 9637 (2003 Ed.); line 9574 (2001 Ed.)
Change From:
"The state of the floating-point environment in the new process image shall be set to the default."

To:
"The state of the floating-point environment in the new process image [THR]or in the initial thread of the new process image[/THR] shall be set to the default."

On line 9650 (2003 Ed.); line 9587 (2001 Ed.)
Change From:
"After a successful call to any of the exec functions, any functions previously registered by atexit() are no longer registered."

To:
"After a successful call to any of the exec functions, any functions previously registered by atexit() [THR]or pthread_atfork()[/THR] are no longer registered."

On lines 9675-9677 (2003 Ed.); lines 9612-9614 (2001 Ed.)
Change From:
"[PS]For the SCHED_FIFO and SCHED_RR scheduling policies, the policy and priority settings shall not be changed by a call to an exec function. For other scheduling policies, the policy and priority settings on exec are implementation-defined.[/PS]"

To:
"When the calling process image does not use the SCHED_FIFO, SCHED_RR, or SCHED_SPORADIC scheduling policies, the scheduling policy and parameters of the new process image and the initial thread in that new process image are implementation defined. [PS]When the calling process image uses the SCHED_FIFO, SCHED_RR, or SCHED_SPORADIC scheduling policies, the process policy and scheduling parameter settings shall not be changed by a call to an exec function.[/PS] [TPS]The initial thread in the new process image shall inherit the process scheduling policy and parameters. It shall have the default system contention scope, but shall inherit its allocation domain from the calling process image.[/TPS]"

Insert the following after line 9701 (2003 Ed.); line 9638 (2001 Ed.) (before the list that begins "The new process shall inherit at least..."):
"[THR]The thread id of the initial thread in the new process image is unspecified.

The size and location of the stack on which the initial thread in the new process image runs is unspecified.

The initial thread in the new process image shall have its cancellation type set to PTHREAD_CANCEL_DEFERRED and its cancellation state set to PTHREAD_CANCEL_ENABLED.

The initial thread in the new process image shall have all

thread-specific data values set to NULL and all thread-specific data keys shall be removed by the call to exec without running destructors.

The initial thread in the new process image shall be joinable, as if created with the detachstate attribute set to PTHREAD_CREATE_JOINABLE. [/THR]"

Add after line 9722 (2003 Ed.); line 9659 (2001 Ed.)
"The initial thread of the new process shall inherit at least the following attributes from the calling thread:

    * Signal mask (see sigprocmask() and pthread_sigmask())
    * Pending signals (see sigpending())"

On lines 9723-9725 (2003 Ed.); lines 9660-9662 (2001 Ed.)
Change From:
"All other process attributes defined in this volume of IEEE Std 1003.1-2001 shall be the same in the new and old process images. The inheritance of process attributes not defined by this volume of IEEE Std 1003.1-2001 is implementation-defined."

To:
"All other process attributes defined in this volume of IEEE Std 1003.1-2001 shall be inherited in the new process image from the old process image. All other thread attributes defined in this volume of IEEE Std 1003.1-2001 shall be inherited in the initial thread in the new process image from the calling thread in the old process image. The inheritance of process or thread attributes not defined by this volume of IEEE Std 1003.1-2001 is implementation-defined."

On lines 9726-9728 (2003 Ed.); lines 9663-9665 (2001 Ed.)
Change From:
"A call to any exec function from a process with more than one thread shall result in all threads being terminated and the new executable image being loaded and executed. No destructor functions shall be called."

To:
"[THR]A call to any exec function from a process with more than one thread shall result in all threads being terminated and the new executable image being loaded and executed. No destructor functions or cleanup handlers shall be called.[/THR]"

Change Number: XSH/TC2/D6/25 [XSH ERN 9]

On Page: 303 Line: 9890 Section: exec (2003 Ed.)
On Page: 301 Line: 9825 Section: exec (2001 Ed.)

In the RATIONALE section
Change From:
"SIG_IGN, and that the process signal mask be unchanged across an exec."

To:
"SIG_IGN, and that the new process image inherits the signal mask of the thread that called exec in the old process image"

Change Number: XSH/TC2/D6/26 [XSH ERN 119]

On Page: 313 Line: 10264 Section: exp (2003 Ed.)
On Page: 311 Line: 10192 Section: exp (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Computing the density of the standard normal distribution

This function shows an implementation for the density of the standard normal distribution using exp(). This example uses the constant M_PI which is an XSI extension.

```
#include <math.h>

double
normal_density (double  x
                )
{
  return exp(-x*x/2) / sqrt (2*M_PI);
}"
```

Change Number: XSH/TC2/D6/27 [XSH ERN 121]

On Page: 318 Line: 10437 Section: fabs (2003 Ed.)
On Page: 316 Line: 10365 Section: fabs (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
" Computing the 1-norm of a floating point vector

This example shows the use of fabs() to compute the 1-norm of a vector defined as follows:

norm1(v) = |v[0]| + |v[1]| + ... + |v[n-1]|

where |x| denotes the absolute value of x,
n denotes the vector's dimension
v[i] denotes the i-th component of v (0<=i<n)

```
 #include <math.h>

 double
 norm1(const double v[], const int n )
 {
   int     i;
   double  n1_v;  /* 1-norm of v */

   n1_v = 0;
   for (i=0; i<n; i++) {
     n1_v += fabs (v[i]);
   }

   return n1_v;
 }"
```

Change Number: XSH/TC2/D6/28 [XSH ERN 11]

On Page: 327 Line: 10726 Section: fclose (2003 Ed.)
On Page: 325 Line: 10655 Section: fclose (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the process would be delayed in the write operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the thread would be delayed in the write operation. [/CX]"

Rationale: It is the thread that is delayed not the process.

Change Number: XSH/TC2/D6/29 [XSH ERN 134]

On Page: 332 Line: 10935 Section: fcntl (2003 Ed.)
On Page: 330 Line: 10863 Section: fcntl (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Locking and unlocking a file

The following example demonstrates how to place a lock on bytes 100 to 109 of a file and then later remove it. F_SETLK is used to perform a non-blocking lock request so that the process does not have to wait if an incompatible lock is held by another process; instead the process can take some other action.

```
 #include <stdlib.h>
 #include <unistd.h>
 #include <fcntl.h>
 #include <errno.h>

 int
 main(int argc, char *argv[])
 {
     int fd;
     struct flock fl;

     fd = open("testfile", O_RDWR);
     if (fd == -1)
         /* Handle error */;

     /* Make a non-blocking request to place a write lock
        on bytes 100-109 of testfile */

     fl.l_type = F_WRLCK;
     fl.l_whence = SEEK_SET;
     fl.l_start = 100;
     fl.l_len = 10;

     if (fcntl(fd, F_SETLK, &fl) == -1) {
         if (errno == EACCES || errno == EAGAIN) {
             printf("Already locked by another process\n");

             /* We can't get the lock at the moment */

         } else {
             /* Handle unexpected error */;
         }
     } else { /* Lock was granted... */
```

```
1220          /* Perform I/O on bytes 100 to 109 of file */

1221          /* Unlock the locked bytes */

1222          fl.l_type = F_UNLCK;
1223          fl.l_whence = SEEK_SET;
1224          fl.l_start = 100;
1225          fl.l_len = 10;
1226          if (fcntl(fd, F_SETLK, &fl) == -1)
1227              /* Handle error */;
1228      }
1229      exit(EXIT_SUCCESS);
1230  } /* main */
```

1231 Setting the close-on-exec flag

1232 The following example demonstrates how to set the close on exec flag
1233 for the file descriptor fd.

```
1234  #include <unistd.h>
1235  #include <fcntl.h>
1236  ...
1237      int flags;

1238      flags = fcntl(fd, F_GETFD);
1239      if (flags == -1)
1240          /* Handle error */;
1241      flags |= FD_CLOEXEC;
1242      if (fcntl(fd, F_SETFD, flags) == -1)
1243          /* Handle error */;"
```

1244 **Change Number: XSH/TC2/D6/30** [XSH ERN 141]

1245 On Page: 343 Line: 11245-11247 Section: fdopen (2003 Ed.)
1246 On Page: 341 Line: 11173-11175 Section: fdopen (2001 Ed.)

1247 In the RATIONALE section
1248 Change From:
1249 "The file descriptor may have been obtained from open(), creat(), pipe(),
1250 dup(), or fcntl(); inherited through fork() or exec; or perhaps obtained
1251 by implementation-defined means, such as the 4.3 BSD socket() call."

1252 To:
1253 The file descriptor may have been obtained from open(), creat(), pipe(),
1254 dup(), fcntl() or socket(); inherited through fork(), posix_spawn()
1255 or exec; or perhaps obtained by other means.

1256 **Change Number: XSH/TC2/D6/31** [XSH ERN 12]

1257 On Page: 360 Line: 11686 Section: fflush (2003 Ed.)
1258 On Page: 358 Line: 11614 Section: fflush (2001 Ed.)

1259 In the ERRORS section
1260 Change From:
1261 "[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
1262 underlying stream and the process would be delayed in the write
1263 operation. [/CX]"

1264 To:
1265 "[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
1266 underlying stream and the thread would be delayed in the write
1267 operation. [/CX]"

Change Number: XSH/TC2/D6/32 [XSH ERN 13]

On Page: 364 Line: 11802 Section: fgetc (2003 Ed.)
On Page: 362 Line: 11730 Section: fgetc (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the process would be delayed in the fgetc() operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the thread would be delayed in the fgetc() operation. [/CX]"

Change Number: XSH/TC2/D6/33 [XSH ERN 14]

On Page: 370 Line: 11969 Section: fgetwc (2003 Ed.)
On Page: 368 Line: 11897 Section: fgetwc (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the process would be delayed in the fgetwc() operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor underlying stream and the thread would be delayed in the fgetwc() operation. [/CX]"

Change Number: XSH/TC2/D6/34 [XBD ERN 11]

On Page: 399 Line: 12942 Section: fpathconf (2003 Ed.)
On Page: 397 Line: 12860 Section: fpathconf (2001 Ed.)

In the DESCRIPTION section
Insert after the "{PIPE_BUF}" entry:

"{POSIX2_SYMLINKS} _PC_2_SYMLINKS 4"

Change Number: XSH/TC2/D6/35 [XSH ERN 15]

On Page: 399-402 Line: 12943-12947,12975,13054 Section: fpathconf (2003 Ed.)
On Page: 397-400 Line: 12861-12865,12893,12972 Section: fpathconf (2001 Ed.)

In the DESCRIPTION section
Insert the number "10" into the third column for the rows beginning {POSIX_ALLOC_SIZE_MIN}, {POSIX_REC_INCR_XFER_SIZE}, {POSIX_REC_MAX_XFER_SIZE}, {POSIX_REC_MIN_XFER_SIZE}, and {POSIX_REC_XFER_ALIGN}.

Add a new bullet item after Item 9 after line 12975 (2003 Ed.); line 12893 (2001 Ed.)
"10. If path or fildes does not refer to a regular file, it is unspecified whether an implementation supports an association of the variable name with the specified file. If an implementation supports such an association for other than a regular file, the value returned is unspecified."

Add to the RATIONALE section as a new paragraph, after line 13054 (2003 Ed.); line 12972 (2001 Ed.):
"It was the intention of 1003.1d that the variables {POSIX_ALLOC_SIZE_MIN}, {POSIX_REC_INCR_XFER_SIZE}, {POSIX_REC_MAX_XFER_SIZE}, {POSIX_REC_MIN_XFER_SIZE}, and {POSIX_REC_XFER_ALIGN} only applied to regular files, but note 10 also permits implementation of the advisory semantics on other file types

unique to an implementation (e.g. a character special device.) "

Change Number: XSH/TC2/D6/36 [XBD ERN 27]

On Page: 400 Line: 12991,13020 Section: fpathconf (2003 Ed.)
On Page: 398 Line: 12909,12938 Section: fpathconf (2001 Ed.)

Add to the end of the RETURN VALUE section as a new paragraph:
"If the variable corresponding to name is dependent on an unsupported
option the results are unspecified."

Add to the end of the APPLICATION USAGE section:
"Application writers should check whether an option, such as
_POSIX_ADVISORY_INFO is supported, prior to obtaining and using values
for related variables such as POSIX_ALLOC_SIZE_MIN."

Change Number: XSH/TC2/D6/37 [XSH ERN 16]

On Page: 416 Line: 13661 Section: fputc (2003 Ed.)
On Page: 414 Line: 13576 Section: fputc (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
underlying stream and the process would be delayed in the write
operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
underlying stream and the thread would be delayed in the write
operation. [/CX]"

Change Number: XSH/TC2/D6/38 [XSH ERN 17]

On Page: 420 Line: 13778 Section: fputwc (2003 Ed.)
On Page: 418 Line: 13693 Section: fputwc (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
underlying stream and the process would be delayed in the write
operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor
underlying stream and the thread would be delayed in the write
operation. [/CX]"

Change Number: XSH/TC2/D6/39 [XSH ERN 126]

On Page: 428 Line: 14076 Section: freeaddrinfo (2003 Ed.)
On Page: 426 Line: 13979 Section: freeaddrinfo (2001 Ed.)

In the ERRORS section
Change From:
"The getaddrinfo() function shall fail and return the corresponding value if:"

To:
"The getaddrinfo() function shall fail and return the corresponding
error value if:"

Change Number: XSH/TC2/D6/40 [XSH ERN 142]

On Page: 430 Line: 14141-14144 Section: freopen (2003 Ed.)
On Page: 428 Line: 14031-14034 Section: freopen (2001 Ed.)

In the DESCRIPTION section
Change From:
"If filename is a null pointer, the freopen() function attempt to change the mode of the stream to that by mode, as if the name of the file currently with the stream had been used. It is implementation-defined which changes of mode are permitted (if any) and under what circumstances".

To:
"If filename is a null pointer, the freopen() function shall attempt to change the mode of the stream to that specified by mode, as if the name of the file currently associated with the stream had been used. In this case, the file descriptor associated with the stream need not be closed if the call to freopen() succeeds. It is implementation-defined which changes of mode are permitted (if any) and under what circumstances".

Change Number: XSH/TC2/D6/41 [XSH ERN 142]

On Page: 430,431 Line: 14158,14179 Section: freopen (2003 Ed.)
On Page: 428,429 Line: 14048,14069 Section: freopen (2001 Ed.)

In the ERRORS section
Add after line 14158 (2003 Ed.); line 14048 (2001 Ed.):
"[CX][EBADF] The file descriptor underlying the stream is not a valid file descriptor when filename is a null pointer.[/CX]"

Add after line 14179 (2003 Ed.); line 14069 (2001 Ed.):
"[CX][EBADF] The mode with which the file descriptor underlying the stream was opened does not support the requested mode when filename is a null pointer.[/CX]"

Change Number: XSH/TC2/D6/42 [XSH ERN 18]

On Page: 443 Line: 14598 Section: fseek (2003 Ed.)
On Page: 441 Line: 14488 Section: fseek (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the process would be delayed in the write operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the thread would be delayed in the write operation. [/CX]"

Change Number: XSH/TC2/D6/43 [XSH ERN 19]

On Page: 445 Line: 14679 Section: fsetpos (2003 Ed.)
On Page: 443 Line: 14569 Section: fsetpos (2001 Ed.)

In the ERRORS section
Change From:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the process would be delayed in the write operation. [/CX]"

To:
"[CX][EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the thread would be delayed in the write operation. [/CX]"

Change Number: XSH/TC2/D6/44 [XSH ERN 143]

On Page: 452 Line: 14894-14896 Section: fsync (2003 Ed.)
On Page: 450 Line: 14784-14786 Section: fsync (2001 Ed.)

In the DESCRIPTION section
Change From:
"The fsync() function shall request that all data for the open file descriptor named by fildes is to be transferred to the storage device associated with the file described by fildes in an implementation-defined manner."

To:
"The fsync() function shall request that all data for the open file descriptor named by fildes is to be transferred to the storage device associated with the file described by fildes. The nature of the transfer is implementation-defined."

Rationale: This is an editorial wording change with no change intended in meaning.

Change Number: XSH/TC2/D6/45 [XSH ERN 53]

On Page: 491 Line: 16190,16229 Section: getcontext (2003 Ed.)
On Page: 489 Line: 16077,16116 Section: getcontext (2001 Ed.)

In the SYNOPSIS section
Change the margin marker from "XSI" to "OB XSI", to denote that the getcontext() and setcontext() functions are obsolescent.

In the APPLICATION USAGE section for getcontext add at the end:
"The obsolescent functions getcontext(), makecontext() and swapcontext() can be replaced using POSIX threads functions."

Change Number: XSH/TC2/D6/46 [XSH ERN 20]

On Page: 554,556 Line: 18219,18266 Section: getrlimit (2003 Ed.)
On Page: 551,553 Line: 18094,18139 Section: getrlimit (2001 Ed.)

In the DESCRIPTION section
Change From:
"RLIMIT_STACK This is the maximum size of a process stack, in bytes."

To:
"RLIMIT_STACK This is the maximum size of the initial thread's stack, in bytes."

In the RATIONALE section
Change From:
"None."

To:
"It should be noted that RLIMIT_STACK applies "at least" to the stack of the initial thread in the process, and not to the sum of all the stacks in the process, as that would be very limiting unless the value is so big as to provide no value at all with a single thread."

Change Number: XSH/TC2/D6/47 [XSH ERN 21,22,23]

On Page: 563 Line: 18492-18497 Section: getsockopt (2003 Ed.)
On Page: 560 Line: 18359-18364 Section: getsockopt (2001 Ed.)

Change From:
"If SO_LINGER is set, the system blocks the process during close() until it can transmit the data or until the end of the interval indicated by the l_linger member, whichever comes first. If SO_LINGER is not specified, and close() is issued, the system handles the call in a way that allows the process to continue as quickly as possible."

To:
"If SO_LINGER is set, the system shall block the calling thread during close() until it can transmit the data or until the end of the interval indicated by the l_linger member, whichever comes first. If SO_LINGER is not specified, and close() is issued, the system handles the call in a way that allows the calling thread to continue as quickly as possible."

Change Number: XSH/TC2/D6/48 [XSH ERN 48]

On Page: 580 Line: 19104-19108 Section: gmtime (2003 Ed.)
On Page: 577 Line: 18970-18974 Section: gmtime (2001 Ed.)

In the RETURN VALUE section
Change From:
"Upon successful completion, gmtime_r() shall return the address of the structure pointed to by the argument result. If an error is detected, gmtime_r() shall return a null pointer."

To:
"Upon successful completion, gmtime_r() shall return the address of the structure pointed to by the argument result. If an error is detected, gmtime_r() shall return a null pointer and set errno to indicate the error".

In the ERRORS section of the 2003 Ed.
Change From:
"The gmtime() function shall fail if:

[EOVERFLOW] The result cannot be represented."

To:
"The gmtime() and gmtime_r() function shall fail if:

[EOVERFLOW] The result cannot be represented."

with "and gmtime_r()" shaded and margin marked TSF.

In the ERRORS section of the 2001 Ed.
Change from:
"No errors are defined"

To:
"The gmtime() and gmtime_r() function shall fail if:

[EOVERFLOW] The result cannot be represented."

with "and gmtime_r()" shaded and margin marked TSF.

Change Number: XSH/TC2/D6/49 [TC2d5 ERN 3]

On Page: 589 Line: 19374 Section: hypot (2003 Ed.)
On Page: 586 Line: 19238 Section: hypot (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"See atan2() EXAMPLES."

Change Number: XSH/TC2/D6/50 [XSH ERN 24]

On Page: 648 Line: 21400 Section: isunordered (2003 Ed.)
On Page: 645 Line: 21256 Section: isunordered (2001 Ed.)

Change From:
"If x or y is NaN, 0 shall be returned."

To:
"If x or y is NaN, 1 shall be returned."

Rationale:
The previous statement is incorrect. If x is NaN, then isunordered()
must return 1. Likewise if y is NaN, then isunordered() must return 1.

Change Number: XSH/TC2/D6/51 [XSH ERN 144]

On Page: 669 Line: 22175-22176 Section: kill (2003 Ed.)
On Page: 666 Line: 22031-22032 Section: kill (2001 Ed.)

In the RATIONALE section
Change From:
"The implementation-defined processes to which a signal cannot be sent
may include the scheduler or init."

To:
"The unspecified processes to which a signal cannot be sent
may include the scheduler or init."

Rationale: The RATIONALE section conflicted with the normative
text in the DESCRIPTION section.

Change Number: XSH/TC2/D6/52 [XSH ERN 135]

On Page: 671 Line: 22533 Section: killpg (2003 Ed.)
On Page: 668 Line: 22109 Section: killpg (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Sending a signal to all other members of a process group

The following example shows how the calling process could send a
signal to all other members of its process group. To prevent
itself from receiving the signal it first makes itself immune to
the signal by ignoring it.

```
 #include <signal.h>
 #include <unistd.h>
 ...
     if (signal(SIGUSR1, SIG_IGN) == SIG_ERR)
         /* Handle error */;

     if (killpg(getpgrp(), SIGUSR1) == -1)
         /* Handle error */;"
```

Change Number: XSH/TC2/D6/53 [XSH ERN 78]

On Page: 686 Line: 22699 Section: lio_listio (2003 Ed.)
On Page: 683 Line: 22555 Section: lio_listio (2001 Ed.)

In the DESCRIPTION section
Add after line 22699 (2003 Ed); line 22555 (2001 Ed.):

"If the buffer pointed to by list or the aiocb structures pointed to by the elements of the array list become illegal addresses before all asynchronous I/O completed and, if necessary, the notification is sent, then the behavior is undefined. If the buffers pointed to by the aio_buf member of the aiocb structure pointed to by the elements of the array list become illegal addresses prior to the asynchronous I/O associated with that aiocb structure being completed, the behavior is undefined."

Rationale: This text is added for symmetry with the aio_read() and aio_write() functions.

Change Number: XSH/TC2/D6/54 [XSH ERN 79]

On Page: 686 Line: 22719 Section: lio_listio (2003 Ed.)
On Page: 683 Line: 22575 Section: lio_listio (2001 Ed.)

In the DESCRIPTION Section (after the last paragraph)
Add after line 22719 (2003 Ed.); line 22575 (2001 Ed.):

"If sig->sigev_notify is SIGEV_THREAD and sig->sigev_notify_attributes is a non-NULL pointer and the block pointed to by this pointer becomes an illegal address prior to all asynchronous I/O being completed, then the behavior is undefined."

Rationale:
This text is added to make it explicit that the user is required to keep the structure pointed to by sig->sigev_notify_attributes valid until the last asynchronous operation finished and the notification has been sent.

Change Number: XSH/TC2/D6/55 [XSH ERN 49]

On Page: 702 Line: 23186-23190 Section: localtime (2003 Ed.)
On Page: 699 Line: 23038-23041 Section: localtime (2001 Ed.)

In the RETURN VALUE
Change From:
"Upon successful completion, localtime_r() shall return a pointer to the structure pointed to by the argument result."

To:
"Upon successful completion, localtime_r() shall return a pointer to the structure pointed to by the argument result. If an error is detected, localtime_r() shall return a null pointer and set errno to indicate the error."

In the ERRORS section of the 2003 Ed.
Change From:
"The localtime() function shall fail if:

[EOVERFLOW] The result cannot be represented."

To:
"The localtime() and localtime_r() functions shall fail if:

[EOVERFLOW] The result cannot be represented."

with "and localtime_r()" shaded and margin marked TSF.

In the ERRORS section of the 2001 Ed.
Change From:

"No errors are defined."

To:
"The localtime() and localtime_r() functions shall fail if:

[EOVERFLOW] The result cannot be represented."

with "and localtime_r()" shaded and margin marked TSF.

Change Number: XSH/TC2/D6/56 [XSH ERN 84,85]

On Page: 702,704 Line: 23181,23245 Section: localtime (2003 Ed.)
On Page: 699,701 Line: 23035,23096 Section: localtime (2001 Ed.)

In the DESCRIPTION section
Add a new paragraph following the last paragraph
with TSF and XSI shading:

"If the reentrant version does not set tzname, it shall not set daylight
and shall not set timezone."

In the SEE ALSO section
Add "tzset()," before "utime(),".

Rationale:
On systems supporting XSI, the daylight, timezone, and tzname variables
should all be set to provide information for the same timezone.
This updates the description of localtime_r() to mention
daylight and timezone as well as tzname.

Change Number: XSH/TC2/D6/57 [XSH ERN 53]

On Page: 732,733 Line: 24081-24083,24143-24146 Section: makecontext (2003 Ed.)
On Page: 729,730 Line: 23930-23932,23993-23995 Section: makecontext (2001 Ed.)

In the SYNOPSIS section
Change the "XSI" margin marker to "OB XSI"

Change the function prototype from:
"void makecontext(ucontext_t *ucp, void (*func)(void), int argc, ...);"

To:
"void makecontext(ucontext_t *ucp, void (*func)(), int argc, ...);"

In the APPLICATION USAGE section
Change From:
"None."

To:
"The obsolescent functions getcontext(), makecontext() and
swapcontext() can be replaced using POSIX threads functions."

In the RATIONALE section
Change From:
"None."

To:
"With the incorporation of ISO/IEC 9899:1999 (C99) into this
specification it was found that the ISO C Standard (subclause 6.11.6)
specifies that the use of function declarators with empty parentheses
is an obsolescent feature. Therefore, using the function prototype:

    void makecontext(ucontext_t *ucp, void (*func)(), int argc, ...);

is making use of an obsolescent feature of ISO C. Therefore, a strictly
conforming POSIX application can't use this form. Therefore, use of
getcontext(), makecontext() and swapcontext() is marked obsolescent.

There is no way in ISO C to specify a non-obsolescent function prototype

indicating that a function will be called with an arbitrary number (including zero) of arguments of arbitrary types (including integers, pointers to data, pointers to functions, and composite types).

Replacing makecontext() with a number of ISO C compatible functions handing various numbers and types of arguments would have forced all existing uses of makecontext() to be rewritten for little or no gain. There are very few applications today that use the *context() routines. Those that do use them are almost always using them to implement co-routines. By maintaining the XSH5 specification for makecontext(), existing applications will continue to work, although they won't be able to be classified as strictly conforming applications.

There is no way in ISO C (without using obsolescent behavior) to specify functionality that was standard, strictly conforming behavior in XSH5 using the 1990 C Standard. Threads can be used to implement the functionality provided by makecontext(), getcontext() and swapcontext() but they are more complex to use. It was felt inventing new ISO C compatible interfaces that describe what can be done with the XSH5 functions and then converting applications to use them would cause more difficulty than just converting applications that use them to use threads instead."

Change Number: XSH/TC2/D6/58 [XSH ERN 50]

On Page: 768 Line: 25150-25152,25154 Section: mktime (2003 Ed.)
On Page: 765 Line: 24997-24999,25001 Section: mktime (2001 Ed.)

In the RETURN VALUE section
Change From:
"The mktime() function shall return the specified time since the Epoch encoded as a value of type time_t. If the time since the Epoch cannot be represented, the function shall return the value (time_t)-1."

To:
"The mktime() function shall return the specified time since the Epoch encoded as a value of type time_t. If the time since the Epoch cannot be represented, the function shall return the value (time_t)-1 [CX] and may set errno to indicate the error.[/CX]."

In the ERRORS section
Change From:
"None."

To:
(Shaded and marked with the CX margin code):
"The mktime() function may fail if:

[EOVERFLOW] The result cannot be represented."

Change Number: XSH/TC2/D6/59 [XSH ERN 86]

On Page: 769 Line: 25185 Section: mktime (2003 Ed.)
On Page: 766 Line: 25032 Section: mktime (2001 Ed.)

In the SEE ALSO section
Add "tzset()," before "utime(), "

Rationale: This is editorial.

Change Number: XSH/TC2/D6/60 [XSH ERN 72]

On Page: 774,777 Line: 25357,25469 Section: mmap (2003 Ed.)
On Page: 771,774 Line: 25204,25316 Section: mmap (2001 Ed.)

In the DESCRIPTION section
Add a new paragraph after line 25357 (2003 Ed.); line 25204 (2001 Ed.) :
"If len is zero mmap() shall fail and no mapping shall be
established."

In the ERRORS section (shall fail)
Add after line 25469 (2003 Ed.); line 25316 (2001 Ed.):
"[EINVAL] The value of len is zero."

Change Number: XSH/TC2/D6/61 [XSH ERN 47]

On Page: 786 Line: 25817 Section: mq_getattr (2003 Ed.)
On Page: 783 Line: 25660 Section: mq_getattr (2001 Ed.)

In the ERRORS section
Change From:
"The mq_getattr() function shall fail if:"

To:
"The mq_getattr() function may fail if:"

Change Number: XSH/TC2/D6/62 [XSH ERN 87]

On Page: 791 Line: 25933 Section: mq_open (2003 Ed.)
On Page: 788 Line: 25776 Section: mq_open (2001 Ed.)

In the DESCRIPTION section (O_CREAT mode)
Change From:
"The file permission bits shall be set to the value of mode. When
bits in mode other than file permission bits are set, the effect is
implementation-defined."

To:
"The permission bits of the message queue shall be set to the value of the
mode argument except those set in the file mode creation mask of the
process. When bits in mode other than the file permission bits are
specified, the effect is unspecified."

Rationale: Consistency with shm_open(), sem_open().

Change Number: XSH/TC2/D6/63 [XSH ERN 25]

On Page: 823 Line: 26879-26880 Section: nanosleep (2003 Ed.)
On Page: 820 Line: 26720-26721 Section: nanosleep (2001 Ed.)

In the RATIONALE section
Change From:
"It is common to suspend execution of a process for an interval in order
to poll the status of a non-interrupting function."

To:
"It is common to suspend execution of a thread for an interval in order
to poll the status of a non-interrupting function."

Change Number: XSH/TC2/D6/64 [XSH ERN 73]

On Page: 829-831 Line: 27026,27069,27102,27114 Section: nftw (2003 Ed.)
On Page: 826-828 Line: 26865,26908,26941,26953 Section: nftw (2001 Ed.)

In the SYNOPSIS section, line 27026 (2003 Ed.); line 26865 (2001 Ed.)
Change From:
"int nftw(const char *path, int (*fn)(const char *, const struct stat *,
int, struct FTW *), int depth, int flags);"

To:
"int nftw(const char *path, int (*fn)(const char *, const struct
stat *, int, struct FTW *), int fd_limit, int flags); "

In the DESCRIPTION section, line 27069 (2003 Ed.); line 26908 (2001 Ed.)
Change From:
"The argument depth sets the maximum number of file descriptors that
shall be used by nftw() while traversing the file tree."

To:
"The argument fd_limit sets the maximum number of file descriptors that
shall be used by nftw() while traversing the file tree."

In the EXAMPLES section, line 27102 (2003 Ed.); line 26941 (2001 Ed.)
Change From:
"The following example walks the /tmp directory and its subdirectories,
calling the nftw() function for every directory entry, to a maximum
of 5 levels deep."

To:
"The following example walks the /tmp directory and its subdirectories,
calling the nftw() function for every directory entry, using a maximum
of 5 file descriptors."

On line 27114 (2003 Ed.); line 26953 (2001 Ed.)
Change From:
"int depth = 5;
int flags = FTW_CHDIR | FTW_DEPTH | FTW_MOUNT;
int ret;
ret = nftw(startpath, nftwfunc, depth, flags);"

To:
"int fd_limit = 5;
int flags = FTW_CHDIR | FTW_DEPTH | FTW_MOUNT;
int ret;
ret = nftw(startpath, nftwfunc, fd_limit, flags);"

Change Number: XSH/TC2/D6/65 [XSH ERN 127]

On Page: 856 Line: 27866 Section: pipe (2003 Ed.)
On Page: 853 Line: 27705 Section: pipe (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Using a pipe to pass data between a parent process and a child process

The following example demonstrates the use of a pipe to transfer data
between a parent process and a child process. Error handling is excluded,
but otherwise this code demonstrates good practice when using pipes:
after the fork() the two processes close the unused ends of the pipe
before they commence transferring data.

```
#include <stdlib.h>
#include <unistd.h>
 ...

 int fildes[2];
```

```
1785  const int BSIZE = 100;
1786  char buf[BSIZE];
1787  ssize_t nbytes;
1788  int status;

1789  status = pipe(fildes);
1790  if (status == -1 ) {
1791     /* an error occurred */
1792     ....
1793  }

1794  switch (fork()) {
1795  case -1: /* Handle error */
1796      break;

1797  case 0:     /* Child  - reads from pipe */
1798      close(fildes[1]);                       /* Write end is unused */
1799      nbytes = read(fildes[0], buf, BSIZE);   /* Get data from pipe */
1800      /* At this point, a further read would see end of file... */
1801      close(fildes[0]);                       /* Finished with pipe */
1802      exit(EXIT_SUCCESS);

1803  default:    /* Parent - writes to pipe */
1804      close(fildes[0]);                       /* Read end is unused */
1805      write(fildes[1], "Hello world\n", 12);  /* Write data on pipe */
1806      close(fildes[1]);                       /* Child will see EOF */
1807      exit(EXIT_SUCCESS);
1808  }"
```

1809 **Change Number: XSH/TC2/D6/66** [XSH ERN 145]

1810 On Page: 860 Line: 27995-27999 Section: poll (2003 Ed.)
1811 On Page: 857 Line: 27834-27838 Section: poll (2001 Ed.)

1812 In the EXAMPLES section
1813 Change From:

```
1814 "/* Open STREAMS device. */
1815  fds[0].fd = open("/dev/dev0", ...);
1816  fds[1].fd = open("/dev/dev1", ...);
1817      fds[0].events = POLLOUT | POLLWRBAND;
1818      fds[1].events = POLLOUT | POLLWRBAND;"
```

1819 To:

```
1820 "/* Open STREAMS device. */
1821  fds[0].fd = open("/dev/dev0", ...);
1822  fds[1].fd = open("/dev/dev1", ...);
1823  fds[0].events = POLLOUT | POLLWRBAND;
1824  fds[1].events = POLLOUT | POLLWRBAND;"
```

Change Number: XSH/TC2/D6/67 [XSH ERN 128]

On Page: 863 Line: 28077 Section: popen (2003 Ed.)
On Page: 860 Line: 27916 Section: popen (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Using popen() to obtain a list of files from the ls utility

The following example demonstrates the use of popen()
and pclose() to execute the command "ls *" in order to
obtain a list of files in the current directory:

```
#include <stdio.h>
 ...

 FILE *fp;
 int status;
 char path[PATH_MAX];

 fp = popen("ls *", "r");
 if (fp == NULL)
     /* Handle error */;

 while (fgets(path, PATH_MAX, fp) != NULL)
     printf("%s", path);

 status = pclose(fp);
 if (status == -1) {
     /* Error reported by pclose() */
     ...
 } else {
     /* Use macros described under wait() to inspect 'status' in order
        to determine success/failure of command executed by popen() */
     ...
 }"
```

Change Number: XSH/TC2/D6/68 [XSH ERN 146]

On Page: 864 Line: 28122 Section: posix_fadvise (2003 Ed.)
On Page: 861 Line: 27961 Section: posix_fadvise (2001 Ed.)

In the SYNOPSIS section
Change From:
"int posix_fadvise(int fd, off_t offset, size_t len, int advice);"

To:
"int posix_fadvise(int fd, off_t offset, off_t len, int advice);"

Rationale: The previous prototype was not large-file aware, and the standard
developers felt it acceptable to make this change before
implementations of the functions become widespread."

Change Number: XSH/TC2/D6/69 [XSH ERN 146]

On Page: 866 Line: 28177 Section: posix_fallocate (2003 Ed.)
On Page: 863 Line: 28016 Section: posix_fallocate (2001 Ed.)

In the SYNOPSIS section
Change From:
"int posix_fallocate(int fd, off_t offset, size_t len);"

To:
"int posix_fallocate(int fd, off_t offset, off_t len);"

Rationale: The previous prototype was not large-file aware, and the standard developers felt it acceptable to make this change before implementations of the functions become widespread."

Change Number: XSH/TC2/D6/70 [XSH ERN 110]

On Page: 975 Line: 31069-31071 Section: pselect (2003 Ed.)
On Page: 972 Line: 30903-30905 Section: pselect (2001 Ed.)

In the DESCRIPTION section
Change From:
"If sigmask is not a null pointer, then the pselect() function shall replace the signal mask of the process by the set of signals pointed to by sigmask before examining the descriptors, and shall restore the signal mask of the process before returning."

To:
"If sigmask is not a null pointer, then the pselect() function shall replace the signal mask of the caller by the set of signals pointed to by sigmask before examining the descriptors, and shall restore the signal mask of the calling thread before returning."

Change Number: XSH/TC2/D6/71 [XSH ERN 147,148]

On Page: 981 Line: 31293 Section: pthread_attr_destroy (2003 Ed.)
On Page: 978 Line: 31127 Section: pthread_attr_destroy (2001 Ed.)

In the ERRORS section
Add
"The pthread_attr_destroy() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object.

The pthread_attr_init() may fail if:

[EBUSY] The implementation has detected an attempt to reinitialize the thread attribute referenced by attr, a previously initialized, but not yet destroyed, thread attribute."

Change Number: XSH/TC2/D6/72 [XSH ERN 125]

On Page: 984 Line: 31407 Section: pthread_attr_getdetachstate (2003 Ed.)
On Page: 981 Line: 31241 Section: pthread_attr_getdetachstate (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Retrieving the detachstate attribute.

This example shows how to obtain the detachstate attribute of a thread attribute object.

```
 #include <pthread.h>

 pthread_attr_t thread_attr;
 int            detachstate;
 int            rc;

 /* code initializing thread_attr */
 ...


 rc = pthread_attr_getdetachstate (&thread_attr, &detachstate);
 if (rc!=0) {
   /* handle error */
   ...
 }
 else {
   /* legal values for detachstate are:
    * PTHREAD_CREATE_DETACHED or PTHREAD_CREATE_JOINABLE
    */
    ...
 }"
```

Change Number: XSH/TC2/D6/73 [XSH ERN 149]

On Page: 984 Line: 31404 Section: pthread_attr_getdetachstate (2003 Ed.)
On Page: 981 Line: 31238 Section: pthread_attr_getdetachstate (2001 Ed.)

In the ERRORS section
Add
"The pthread_attr_getdetachstate() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object.

The pthread_attr_setdetachstate() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/74 [XSH ERN 150]

On Page: 986 Line: 31461,31462 Section: pthread_attr_getguardsize (2003 Ed.)
On Page: 983 Line: 31295,31296 Section: pthread_attr_getguardsize (2001 Ed.)

In the ERRORS section
On line 31461 (2003 Ed.); line 31295 (2001 Ed.) delete:
"[EINVAL] The attribute attr is invalid."

In the ERRORS section
Add:
"The pthread_attr_getguardsize() and pthread_attr_setguardsize may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/75 [XSH ERN 151,152]

On Page: 986 Line: 31500,31516,31517 Section: pthread_attr_getinheritsched (2003 Ed.)
On Page: 983 Line: 31334,31351,31352 Section: pthread_attr_getinheritsched (2001 Ed.)

In the DESCRIPTION section
After line 31500 (2003 Ed.); line 31334 (2001 Ed.) insert a new paragraph:
"The supported values of inheritsched shall be:"

In the ERRORS section
Insert before line 31516 (2003 Ed.); line 31350 (2001 Ed.)
"The pthread_attr_getinheritsched() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

In the ERRORS section
Add after line 31517 (2003 Ed.); 31351 (2001 Ed.):

"[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/76 [XSH ERN 124]

On Page: 987 Line: 31465 Section: pthread_attr_getguardsize (2003 Ed.)
On Page: 984 Line: 31299 Section: pthread_attr_getguardsize (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Retrieving the guardsize attribute.

This example shows how to obtain the guardsize attribute of a thread attribute object.

```
 #include <pthread.h>

 pthread_attr_t thread_attr;
 size_t  guardsize;
 int     rc;

 /* code initializing thread_attr */
 ...

 rc = pthread_attr_getguardsize (&thread_attr, &guardsize);
 if (rc != 0)  {
   /* handle error */
   ...
 }
 else {
   if (guardsize > 0) {
     /* a guard area of at least guardsize bytes is provided */
     ...
   }
   else {
     /* no guard area provided */
     ...
   }
 }"
```

Change Number: XSH/TC2/D6/77 [XSH ERN 55]

On Page: 988 Line: 31524 Section: pthread_attr_getinheritsched (2003 Ed.)
On Page: 985 Line: 31358 Section: pthread_attr_getinheritsched (2001 Ed.)

In the APPLICATION USAGE section
Add:
"See Section 2.9.4 for further details on thread scheduling attributes and their default settings."

Change Number: XSH/TC2/D6/78 [XSH ERN 154]

On Page: 990 Line: 31567,31569 Section: pthread_attr_getschedparam (2003 Ed.)
On Page: 987 Line: 31401,31403 Section: pthread_attr_getschedparam (2001 Ed.)

In the ERRORS section
Add after line 31567 (2003 Ed.); line 31401 (2001 Ed.):

"The pthread_attr_getschedparam() function may fail if:

[EINVAL] The value specified by attr does not refer to an initialized
thread attribute object."

On line 31569 (2003 Ed.);31403 (2001 Ed.)
Change From:
"[EINVAL] The value of param is not valid."

To:
"[EINVAL] The value of param is not valid, or the value specified by attr does
not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/79 [XSH ERN 56]

On Page: 992 Line: 31621 Section: pthread_attr_getschedpolicy (2003 Ed.)
On Page: 989 Line: 31455 Section: pthread_attr_getschedpolicy (2001 Ed.)

In the APPLICATION USAGE section
Add:
"See Section 2.9.4 for further details on thread scheduling
attributes and their default settings."

Change Number: XSH/TC2/D6/80 [XSH ERN 155]

On Page: 992 Line: 31612,31614 Section: pthread_attr_getschedpolicy (2003 Ed.)
On Page: 989 Line: 31446,31448 Section: pthread_attr_getschedpolicy (2001 Ed.)

In the ERRORS section
Add after line 31612 (2003 Ed.); line 31446 (2001 Ed.):

"The pthread_attr_getschedpolicy() function may fail if:

[EINVAL] The value specified by attr does not refer to an initialized
thread attribute object."

On line 31614 (2003 Ed.);31448 (2001 Ed.)
Change From:
"[EINVAL] The value of policy is not valid."

To:
"[EINVAL] The value of policy is not valid, or the value specified by
attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/81 [XSH ERN 57]

On Page: 994 Line: 31669 Section: pthread_attr_getscope (2003 Ed.)
On Page: 991 Line: 31503 Section: pthread_attr_getscope (2001 Ed.)

In the APPLICATION USAGE section
Add:
"See Section 2.9.4 for further details on thread scheduling
attributes and their default settings."

Change Number: XSH/TC2/D6/82 [XSH ERN 157]

On Page: 994 Line: 31660,31662 Section: pthread_attr_getscope (2003 Ed.)
On Page: 991 Line: 31494,31496 Section: pthread_attr_getscope (2001 Ed.)

In the ERRORS section
Add after line 31660 (2003 Ed.); line 31494 (2001 Ed.):

"The pthread_attr_getscope() function may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

On line 31662 (2003 Ed.);31496 (2001 Ed.)
Change From:
"[EINVAL] The value of contentionscope is not valid."

To:
"[EINVAL] The value of contentionscope is not valid, or the value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/83 [XSH ERN 89]

On Page: 996 Line: 31731 Section: pthread_attr_getstack (2003 Ed.)
On Page: 993 Line: 31565 Section: pthread_attr_getstack (2001 Ed.)

In the APPLICATION USAGE
Add at the end:

"After a successful call to pthread_attr_setstack(), the storage area specified by the stackaddr parameter is under the control of the implementation as described in Section 2.9.8."

Change Number: XSH/TC2/D6/84 [XSH ERN 158]

On Page: 996 Line: 31714,31716 Section: pthread_attr_getstack (2003 Ed.)
On Page: 993 Line: 31548,31550 Section: pthread_attr_getstack (2001 Ed.)

In the ERRORS section
Add after line 31774 (2003 Ed.); line 31716 (2001 Ed.):

"The pthread_attr_getstack() function may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

On line 31716 (2003 Ed.);31550 (2001 Ed.)
Change From:
"[EINVAL] The value of stackaddr does not have proper alignment to be used as a stack, or if (stackaddr + stacksize) lacks proper alignment."

To:
"[EINVAL] The value of stackaddr does not have proper alignment to be used as a stack, or (stackaddr + stacksize) lacks proper alignment, or the value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/85 [XSH ERN 89]

On Page: 998 Line: 31777 Section: pthread_attr_getstackaddr (2003 Ed.)
On Page: 995 Line: 31611 Section: pthread_attr_getstackaddr (2001 Ed.)

In the APPLICATION USAGE
Add at the end:

"After a successful call to pthread_attr_setstackaddr(), the storage area specified by the stackaddr parameter is under the control of the implementation as described in Section 2.9.8."

Change Number: XSH/TC2/D6/86 [XSH ERN 159]

On Page: 998 Line: 31761 Section: pthread_attr_getstackaddr (2003 Ed.)
On Page: 995 Line: 31595 Section: pthread_attr_getstackaddr (2001 Ed.)

In the ERRORS section
Change From:
"No errors are defined."

To:
"The pthread_attr_getstackaddr() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object.

The pthread_attr_setstackaddr() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/87 [XSH ERN 160]

On Page: 1000 Line: 31815 Section: pthread_attr_getstacksize (2003 Ed.)
On Page: 997 Line: 31649 Section: pthread_attr_getstacksize (2001 Ed.)

In the ERRORS section
Add:
"The pthread_attr_getstacksize() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object.

The pthread_attr_setstacksize() may fail if:

[EINVAL] The value specified by attr does not refer to an initialized thread attribute object."

Change Number: XSH/TC2/D6/88 [XSH ERN 64]

On Page: 1024 Line: 32233 Section: pthread_cleanup_pop (2003 Ed.)
On Page: 1020 Line: 32064 Section: pthread_cleanup_pop (2001 Ed.)

In the DESCRIPTION section
Add a new final paragraph:
"The effect of the use of 'return', 'break', 'continue', and 'goto' to prematurely leave a code block described by a pair of pthread_cleanup_push() and pthread_cleanup_pop() functions calls is undefined."

Change Number: XSH/TC2/D6/89 [XSH ERN 90]

On Page: 1036 Line: 32652-32655 Section: pthread_cond_timedwait (2003 Ed.)
On Page: 1033 Line: 32483-32486 Section: pthread_cond_timedwait (2001 Ed.)

In the DESCRIPTION section
Change From:
"The effect of using more than one mutex for concurrent pthread_cond_timedwait() or pthread_cond_wait() operations on the same condition variable is undefined; that is, a condition variable becomes bound to a unique mutex when a thread waits on the condition variable, and this (dynamic) binding shall end when the wait returns."

To:
"When a thread waits on a condition variable, having specified a particular mutex to either the pthread_cond_timedwait() or the pthread_cond_wait() operation, a dynamic binding is formed between that mutex and condition variable that remains in effect as long as at least one thread is blocked on the condition variable. During this time, the effect of an attempt by any thread to wait on that condition variable using a different mutex is undefined. Once all waiting threads have been unblocked (as by the pthread_cond_broadcast() operation), the next wait operation on that condition variable shall form a new dynamic binding with the mutex specified by that wait operation. Even though the dynamic binding between condition variable and mutex may be removed or replaced between the time a thread is unblocked from a wait on the condition variable and the time that it returns to the caller or begins cancellation cleanup, the unblocked thread shall always re-acquire the mutex specified in the condition wait operation call from which it is returning."

Rationale:
This change is for consistency with pthread_cond_destroy() which says it is safe to destroy an initialized condition variable upon which no threads are currently blocked.

Change Number: XSH/TC2/D6/90 [XSH ERN 58]

On Page: 1036 Line: 32656-32659 Section: pthread_cond_timedwait (2003 Ed.)
On Page: 1032 Line: 32487-32490 Section: pthread_cond_timedwait (2001 Ed.)

In the DESCRIPTION section
Change From:
"When the cancelability enable state of a thread is set to PTHREAD_CANCEL_DEFERRED, a side effect of acting upon a cancellation request while in a condition wait is that the mutex is (in effect) re-acquired before calling the first cancellation cleanup handler."

To:
"When the cancelability type of a thread is set to PTHREAD_CANCEL_DEFERRED, a side effect of acting upon a cancellation request while in a condition wait is that the mutex is (in effect) re-acquired before calling the first cancellation cleanup handler."

Change Number: XSH/TC2/D6/91 [XSH ERN 26]

On Page: 1037 Line: 32690-32693 Section: pthread_cond_timedwait (2003 Ed.)
On Page: 1033 Line: 32521-32524 Section: pthread_cond_timedwait (2001 Ed.)

In the ERRORS section
Change From:
"The pthread_cond_timedwait() and pthread_cond_wait() functions may fail if:

[EINVAL] The value specified by cond, mutex,or abstime is invalid.

[EINVAL] Different mutexes were supplied for concurrent
pthread_cond_timedwait() or pthread_cond_wait() operations on the same
condition variable."

To:
"The pthread_cond_timedwait() function shall fail if:

[ETIMEDOUT] The time specified by abstime to pthread_cond_timedwait()
has passed.

[EINVAL] The value specified by abstime is invalid.

The pthread_cond_timedwait() and pthread_cond_wait() functions may
fail if:

[EINVAL] The value specified by cond or mutex is invalid."

Change Number: XSH/TC2/D6/92 [XSH ERN 118]

On Page: 1037 Line: 32705 Section: pthread_cond_timedwait (2003 Ed.)
On Page: 1033 Line: 32536 Section: pthread_cond_timedwait (2001 Ed.)

In the RATIONALE section
Add a new second paragraph:
"The application needs to recheck the predicate on any return because it
cannot be sure there is another thread waiting on the thread to handle
the signal, and if there is not then the signal is lost. The burden
is on the application to check the predicate."

Change Number: XSH/TC2/D6/93 [XSH ERN 136]

On Page: 1050 Line: 33036,33038 Section: pthread_create (2003 Ed.)
On Page: 1046 Line: 32866,32868 Section: pthread_create (2001 Ed.)

In the ERROR RETURN section
Delete the following from the "shall fail" section:
"[EINVAL] The value specified by attr is invalid."

Add after line 33038 (2003 Ed.); 32868 (2001 Ed.):
"The pthread_create() function may fail if:

[EINVAL] The attributes specified by attr are invalid."

Change Number: XSH/TC2/D6/94 [XSH ERN 91]

On Page: 1051 Line: 33043 Section: pthread_create (2003 Ed.)
On Page: 1047 Line: 32873 Section: pthread_create (2001 Ed.)

In the APPLICATION USAGE section
Change From:
"None."

To:
"There is no requirement on the implementation that the ID of the created thread be available before the newly created thread starts executing. The calling thread can obtain the ID of the created thread through the return value of the pthread_create() function, and the newly created thread can obtain its ID by a call to pthread_self()."

Change Number: XSH/TC2/D6/95 [XSH ERN 27]

On Page: 1053 Line: 33123-33127 Section: pthread_detach (2003 Ed.)
On Page: 1049 Line: 32951-32954 Section: pthread_detach (2001 Ed.)

In the ERRORS section
Change From:
"The pthread_detach() function shall fail if:

[EINVAL] The implementation has detected that the value specified by thread does not refer to a joinable thread.

[ESRCH] No thread could be found corresponding to that specified by the given thread ID."

To:
"The pthread_detach() function may fail if:

[EINVAL] The implementation has detected that the value specified by thread does not refer to a joinable thread.

[ESRCH] No thread could be found corresponding to that specified by the given thread ID."

Change Number: XSH/TC2/D6/96 [XSH ERN 100]

On Page: 1064 Line: 33451 Section: pthread_getspecific (2003 Ed.)
On Page: 1060 Line: 33279 Section: pthread_getspecific (2001 Ed.)

In the ERRORS Section
Change From:
"[ENOMEM] Insufficient memory exists to associate the value with the key."

To:
"[ENOMEM] Insufficient memory exists to associate the non-NULL value with the key."

Change Number: XSH/TC2/D6/97 [XSH ERN 28]

On Page: 1066 Line: 33494-33500 Section: pthread_join (2003 Ed.)
On Page: 1062 Line: 33322-33328 Section: pthread_join (2001 Ed.)

Change From:
"The pthread_join() function shall fail if:

[EINVAL] The implementation has detected that the value specified by thread does not refer to a joinable thread.

[ESRCH] No thread could be found corresponding to that specified by the given thread ID.

The pthread_join() function may fail if:

[EDEADLK] A deadlock was detected or the value of thread specifies the calling thread."

To:
"The pthread_join() function shall fail if:

[ESRCH] No thread could be found corresponding to that specified by the given thread ID.

The pthread_join() function may fail if:

[EDEADLK] A deadlock was detected or the value of thread specifies the calling thread.

[EINVAL] The value specified by thread does not refer to a joinable thread."

Change Number: XSH/TC2/D6/98 [XSH ERN 59,65]

On Page: 1085 Line: 34135,34150 Section: pthread_mutex_lock (2003 Ed.)
On Page: 1081 Line: 33962,33977 Section: pthread_mutex_lock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The current thread already owns the mutex."

To:
"[EDEADLK] A deadlock condition was detected or the current thread already owns the mutex."

In the RATIONALE section
Change From:
"For example, deadlocking on a double-lock is explicitly allowed behavior in order to avoid requiring more overhead in the basic mechanism than is absolutely necessary."

To:
"For example, on systems not supporting the XSI extended mutex types, deadlocking on a double-lock is explicitly allowed behavior in order to avoid requiring more overhead in the basic mechanism than is absolutely necessary."

Add to the end of the RATIONALE section:
"For further rationale on the XSI extended mutex types please see the Rationale Volume."

Change Number: XSH/TC2/D6/99 [XSH ERN 60]

On Page: 1088 Line: 34211-34214 Section: pthread_mutex_timedlock (2003 Ed.)
On Page: 1084 Line: 34038-34041 Section: pthread_mutex_timedlock (2001 Ed.)

In the DESCRIPTION section (last paragraph)
Margin mark the following text TPI and shade:
"As a consequence of the priority inheritance rules (for mutexes initialized with the PRIO_INHERIT protocol), if a timed mutex wait is terminated because its timeout expires, the priority of the owner of the mutex shall be adjusted as necessary to reflect the fact that this thread is no longer among the threads waiting for the mutex."

Change Number: XSH/TC2/D6/100 [XSH ERN 65]

On Page: 1089 Line: 34231 Section: pthread_mutex_timedlock (2003 Ed.)
On Page: 1085 Line: 34058 Section: pthread_mutex_timedlock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The current thread already owns the mutex."

To:
"[EDEADLK] A deadlock condition was detected or the current thread already owns the mutex."

Change Number: XSH/TC2/D6/101 [XSH ERN 65]

On Page: 1114 Line: 34988 Section: pthread_rwlock_rdlock (2003 Ed.)
On Page: 1110 Line: 34811 Section: pthread_rwlock_rdlock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The current thread already owns the read-write lock for writing."

To:
"[EDEADLK] A deadlock condition was detected or the current thread already owns the read-write lock for writing."

Change Number: XSH/TC2/D6/102 [XSH ERN 65]

On Page: 1115 Line: 35054 Section: pthread_rwlock_timedrdlock (2003 Ed.)
On Page: 1111 Line: 34877 Section: pthread_rwlock_timedrdlock (2003 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The calling thread already holds a write lock on rwlock."

To:
"[EDEADLK] A deadlock condition was detected or the calling thread already holds a write lock on rwlock."

Change Number: XSH/TC2/D6/103 [XSH ERN 65]

On Page: 1117 Line: 35112 Section: pthread_rwlock_timedwrlock (2003 Ed.)
On Page: 1113 Line: 34935 Section: pthread_rwlock_timedwrlock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The calling thread already holds the rwlock."

To:
"[EDEADLK] A deadlock condition was detected or the calling thread already holds the rwlock."

Change Number: XSH/TC2/D6/104 [XSH ERN 65]

On Page: 1120 Line: 35178 Section: pthread_rwlock_trywrlock (2003 Ed.)
On Page: 1116 Line: 35001 Section: pthread_rwlock_trywrlock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The current thread already owns the read-write lock for writing or reading."

To:
"[EDEADLK] A deadlock condition was detected or the current thread already owns the read-write lock for writing or reading."

Change Number: XSH/TC2/D6/105 [XSH ERN 30]

On Page: 1139-1140 Line: 35589,35615 Section: pthread_sigmask (2003 Ed.)
On Page: 1135-1136 Line: 35412,35438 Section: pthread_sigmask (2001 Ed.)

In the DESCRIPTION section, line 35589 (2003 Ed.); line 35412 (2001 Ed.)
Change From:
"If set is a null pointer, the value of the argument how is not significant and the process signal mask shall be unchanged; thus the call can be used to enquire about currently blocked signals."

To:
"If set is a null pointer, the value of the argument how is not significant and the thread signal mask shall be unchanged; thus the call can be used to enquire about currently blocked signals."

In the RATIONALE section, line 35615 (2003 Ed.); line 35438 (2001 Ed.)
Change From:
"When a process signal mask is changed in a signal-catching function that is installed by sigaction(), the restoration of the signal mask on return from the signal-catching function overrides that change (see sigaction())."

To:
"When a thread's signal mask is changed in a signal-catching function that is installed by sigaction(), the restoration of the signal mask on return from the signal-catching function overrides that change (see sigaction())."

Change Number: XSH/TC2/D6/106 [XSH ERN 114]

On Page: 1140 Line: 35611 Section: pthread_sigmask (2003 Ed.)
On Page: 1136 Line: 35434 Section: pthread_sigmask (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Signalling in multi-threaded process.

This example shows the use of pthread_sigmask() in order to deal with signals in a multi-threaded process. It provides a fairly general framework that could be easily adapted/extended by the reader.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
...

    static sigset_t   signal_mask;  /* signals to block         */

    int main (int argc, char *argv[])
    {
        pthread_t  sig_thr_id;      /* signal handler thread ID */
        int        rc;              /* return code              */

        sigemptyset (&signal_mask);
        sigaddset (&signal_mask, SIGINT);
        sigaddset (&signal_mask, SIGTERM);
        rc = pthread_sigmask (SIG_BLOCK, &signal_mask, NULL);
        if (rc != 0) {
            /* handle error */
         ...
        }
        /* any newly created threads inherit the signal mask */

        rc = pthread_create (&sig_thr_id, NULL, signal_thread, NULL);
        if (rc != 0) {
            /* handle error */
            ...
        }

        /* APPLICATION CODE */
        ...
    }


    void *signal_thread (void *arg)
    {
        int       sig_caught;    /* signal caught       */
        int       rc;            /* returned code       */

        rc = sigwait (&signal_mask, &sig_caught);
        if (rc != 0) {
            /* handle error */
        }
        switch (sig_caught)
        {
        case SIGINT:     /* process SIGINT  */
            ...
            break;
        case SIGTERM:    /* process SIGTERM */
            ...
            break;
        default:         /* should normally not happen */
            fprintf (stderr, "\nUnexpected signal %d\n", sig_caught);
```

```
            break;
        }
    }
```

Change Number: XSH/TC2/D6/107 [XSH ERN 65]

On Page: 1143 Line: 35723 Section: pthread_spin_lock (2003 Ed.)
On Page: 1139 Line: 35546 Section: pthread_spin_lock (2001 Ed.)

In the ERRORS section
Change From:
"[EDEADLK] The calling thread already holds the lock."

To:
"[EDEADLK] A deadlock condition was detected or the calling thread already holds the lock."

Change Number: XSH/TC2/D6/108 [XSH ERN 43]

On Page: 1174 Line: 36533 Section: read (2003 Ed.)
On Page: 1169 Line: 36335 Section: read (2001 Ed.)

In the ERRORS section
Change From:
"[EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the process would be delayed."

To:
"[EAGAIN] The O_NONBLOCK flag is set for the file descriptor and the thread would be delayed."

Change Number: XSH/TC2/D6/109 [XSH ERN 165]

On Page: 1177 Line: 36652-36654 Section: read (2003 Ed.)
On Page: 1172 Line: 36454-36456 Section: read (2001 Ed.)

In the RATIONALE section
Change From:
"IEEE Std 1003.1-2001 does not specify when an implementation that buffers read()ss actually moves the data into the user-supplied buffer, so an implementation may chose to do this at the latest possible moment."

To:
"IEEE Std 1003.1-2001 does not specify when an implementation that buffers read()s actually moves the data into the user-supplied buffer, so an implementation may chose to do this at the latest possible moment."

Change Number: XSH/TC2/D6/110 [XSH ERN 111]

On Page: 1190 Line: 37082,37091 Section: realpath() (2003 Ed.)
On Page: 1185 Line: 36862,36872 Section: realpath() (2001 Ed.)

Change From:
"[ELOOP] A loop exists in symbolic links encountered during resolution of the path argument."

To:
"[ELOOP] A loop exists in symbolic links encountered during resolution of the file_name argument."

Change From:
"[ELOOP] More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the path argument.

To:
"[ELOOP] More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the file_name argument.

Rationale:
These ERROR descriptions erroneously referred to a non-existent
path argument.

Change Number: XSH/TC2/D6/111 [XSH Errata 2003-1]

On Page: 1247 Line: 38887 Section: select (2003 Ed.)
On Page: 1242 Line: 38658 Section: select (2001 Ed.)

In the SYNOPSIS section
Change from:
"#include <sys/time.h>"

To:
"#include <sys/select.h>"

Rationale: This is an editorial correction to the pointer page
to match the actual manual page.

Change Number: XSH/TC2/D6/112 [Errata Item 1]

On Page: 1247 Line: 38887 Section: select (2003 Ed.)
On Page: 1242 Line: 38656 Section: select (2001 Ed.)

In the SYNOPIS section

Change From:
"#include <sys/time.h>"

To:
"#include <sys/select.h>"

Rationale: This corrects the pointer page to be consistent with the
pselect() manual page.

Change Number: XSH/TC2/D6/113 [XSH ERN 66]

On Page: 1248 Line: 38915 Section: sem_close (2003 Ed.)
On Page: 1243 Line: 38686 Section: sem_close (2001 Ed.)

In the ERRORS section
Change From:
"The sem_close() function shall fail if:"

To:
"The sem_close() function may fail if:"

Change Number: XSH/TC2/D6/114 [XSH ERN 67]

On Page: 1249 Line: 38952-38955 Section: sem_destroy (2003 Ed.)
On Page: 1244 Line: 38723-38726 Section: sem_destroy (2001 Ed.)

In the ERRORS section
Change From:
"The sem_destroy() function shall fail if:

[EINVAL] The sem argument is not a valid semaphore.

The sem_destroy() function may fail if:

[EBUSY] There are currently processes blocked on the semaphore."

To:
"The sem_destroy() function may fail if:

[EINVAL] The sem argument is not a valid semaphore.

[EBUSY] There are currently processes blocked on the semaphore."

Change Number: XSH/TC2/D6/115 [XSH ERN 68]

On Page: 1250 Line: 38993 Section: sem_getvalue (2003 Ed.)
On Page: 1245 Line: 38764 Section: sem_getvalue (2001 Ed.)

In the ERRORS section
Change From:
"The sem_getvalue() function shall fail if:"

To:
"The sem_getvalue() function may fail if:"

Change Number: XSH/TC2/D6/116 [XSH ERN 102]

On Page: 1252 Line: 39025-39036 Section: sem_init (2003 Ed.)
On Page: 1246 Line: 38795-38806 Section: sem_init (2001 Ed.)

In the DESCRIPTION section
Change From:
"The sem_init() function shall initialize the unnamed semaphore referred to by sem. The value of the initialized semaphore shall be value. Following a successful call to sem_init(), the semaphore may be used in subsequent calls to sem_wait(), sem_trywait(), sem_post(), and sem_destroy(). This semaphore shall remain usable until the semaphore is destroyed.

If the pshared argument has a non-zero value, then the semaphore is shared between processes; in this case, any process that can access the semaphore sem can use sem for performing sem_wait(),sem_trywait(),sem_post(), and sem_destroy() operations.

Only sem itself may be used for performing synchronization. The result of referring to copies of sem in calls to sem_wait(),sem_trywait(),sem_post(), and sem_destroy() is undefined.

If the pshared argument is zero, then the semaphore is shared between threads of the process; any thread in this process can use sem for performing sem_wait(), sem_trywait(), sem_post(), and sem_destroy() operations. "

To:
"The sem_init() function shall initialize the unnamed semaphore referred to by sem. The value of the initialized semaphore shall be value. Following a successful call to sem_init(), the semaphore may be used in subsequent calls to sem_wait(), [TMO]sem_timedwait(),[/TMO] sem_trywait(), sem_post(), and sem_destroy(). This semaphore shall remain usable until the semaphore is destroyed.

If the pshared argument has a non-zero value, then the semaphore is shared between processes; in this case, any process that can access the semaphore sem can use sem for performing sem_wait(), [TMO]sem_timedwait(),[/TMO]sem_trywait(), sem_post(), and sem_destroy() operations.

Only sem itself may be used for performing synchronization. The result of referring to copies of sem in calls to sem_wait(), [TMO]sem_timedwait(),[/TMO]sem_trywait(), sem_post(), and sem_destroy() is undefined.

If the pshared argument is zero, then the semaphore is shared between threads of the process; any thread in this process can use sem for performing sem_wait(), [TMO]sem_timedwait(),[/TMO]sem_trywait(), sem_post(), and sem_destroy() operations."

Rationale: The sem_timedwait() function is added to the DESCRIPTION for alignment with 1003.1d-1999.

Change Number: XSH/TC2/D6/117 [XSH ERN 103]

On Page: 1254 Line: 39079-39080 Section: sem_open (2003 Ed.)
On Page: 1248 Line: 38849-38850 Section: sem_open (2001 Ed.)

In the DESCRIPTION section
Change From:
"This semaphore may be used in subsequent calls to sem_wait(),
sem_trywait(), sem_post(), and sem_close(). "

To:
"This semaphore may be used in subsequent calls to sem_wait(),
[TMO]sem_timedwait(),[/TMO] sem_trywait(), sem_post(), and sem_close(). "

Rationale: The sem_timedwait() function is added to the DESCRIPTION
for alignment with 1003.1d-1999.

Change Number: XSH/TC2/D6/118 [XSH ERN 97]

On Page: 1254-1255 Line: 39115-39117 Section: sem_open (2003 Ed.)
On Page: 1248-1249 Line: 38885-38887 Section: sem_open (2001 Ed.)

In the DESCRIPTION section
Change From:
"If a process makes multiple successful calls to sem_open() with the same
value for name, the same semaphore address shall be returned for each such
successful call, provided that there have been no calls to sem_unlink()
for this semaphore."

To:
"If a process makes multiple successful calls to sem_open() with the same
value for name, the same semaphore address shall be returned for each such
successful call, provided that there have been no calls to sem_unlink()
for this semaphore and at least one previous successful sem_open()
call for this semaphore has not been matched with sem_close() call."

Change Number: XSH/TC2/D6/119 [XSH ERN 69]

On Page: 1257 Line: 39192 Section: sem_post (2003 Ed.)
On Page: 1251 Line: 38962 Section: sem_post (2001 Ed.)

In the ERRORS section
Change From:
"The sem_post() function shall fail if:"

To:
"The sem_post() function may fail if:"

Change Number: XSH/TC2/D6/120 [XSH ERN 70]

On Page: 1259 Line: 39247 Section: sem_timedwait (2001 Ed.)
On Page: 1253 Line: 39017 Section: sem_timedwait (2001 Ed.)

In the ERRORS section
Move the following text on line 39247 from the "shall fail"
to the "may fail" section:

"[EINVAL] The sem argument does not refer to a valid semaphore."

Change Number: XSH/TC2/D6/121 [XSH ERN 71]

On Page: 1261 Line: 39298 Section: sem_trywait (2003 Ed.)
On Page: 1255 Line: 39068 Section: sem_trywait (2001 Ed.)

In the ERRORS section
Move the following text on line 39298 from the "shall fail" to the
"may fail" section:

"[EINVAL] The sem argument does not refer to a valid semaphore."

Change Number: XSH/TC2/D6/122 [XSH ERN 133]

On Page: 1269 Line: 39510 Section: semget (2003 Ed.)
On Page: 1263 Line: 39280 Section: semget (2001 Ed.)

In the DESCRIPTION section
Change From:
"The data structure associated with each semaphore in the set shall not be initialized."

To:
"The data structure associated with each semaphore in the set need not be initialized."

Change Number: XSH/TC2/D6/123 [XSH ERN 44]

On Page: 1290 Line: 40250-40251 Section: seteuid (2003 Ed.)
On Page: 1284 Line: 40018-40019 Section: seteuid (2001 Ed.)

In the RATIONALE section
Change From:
"[EPERM] The process does not have appropriate privileges and uid does not match the real group ID or the saved set-group-ID."

To:
"[EPERM] The process does not have appropriate privileges and uid does not match the real user ID or the saved set-user-ID."

Rationale. This is a correction of an editorial error.

Change Number: XSH/TC2/D6/124 [XSH ERN 92]

On Page: 1299 Line: 40449,40456 Section: setlocale (2003 Ed.)
On Page: 1293 Line: 40217,40224 Section: setlocale (2001 Ed.)

In the DESCRIPTION section
Change From:
" "" Specifies an implementation-defined native environment. [CX]
     This corresponds to the value of the associated environment
     variables, LC_* and LANG ; see the Base Definitions volume of IEEE Std
     1003.1-2001, Chapter 7, Locale and the Base Definitions volume of IEEE
     Std 1003.1-2001, Chapter 8, Environment Variables. [/CX] "

To:
" "" Specifies an implementation-defined native environment. [CX]
     The determination of the name of the new locale for the specified
     category depends on the value of the associated environment
     variables, LC_* and LANG ; see the Base Definitions volume of IEEE Std
     1003.1-2001, Chapter 7, Locale and the Base Definitions volume of IEEE
     Std 1003.1-2001, Chapter 8, Environment Variables. [/CX] "

In the DESCRIPTION section
Insert before the last paragraph:
"[CX] Setting all of the categories of the locale of the process is similar to successively setting each individual category of the locale of the process, except that all error checking is done before any actions are performed. To set all the categories of the locale of the process, setlocale() is invoked as:

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

In this case, setlocale() shall first verify that the values of all the environment variables it needs according to the precedence rules, described in the Base Definitions Volume of IEEE Std 1003.1-8, Chapter 8, Environment Variables, indicate supported locales. If the value of any of

these environment-variable searches yields a locale that is not supported (and nonnull), setlocale() shall return a null pointer and the locale of the process shall not be changed. If all environment variables name supported locales, setlocale() shall proceed as if it had been called for each category, using the appropriate value from the associated environment variable or from the implementation-defined default if there is no such value. [/CX]"

Change Number: XSH/TC2/D6/125 [XSH ERN 31,32]

On Page: 1319 Line: 40905-40910 Section: setsockopt (2003 Ed.)
On Page: 1313 Line: 40671-40674 Section: setsockopt (2001 Ed.)

Change From:
"If SO_LINGER is set, the system shall block the process during close() until it can transmit the data or until the time expires. If SO_LINGER is not specified, and close() is issued, the system handles the call in a way that allows the process to continue as quickly as possible."

To:
"If SO_LINGER is set, the system shall block the calling thread during close() until it can transmit the data or until the time expires. If SO_LINGER is not specified, and close() is issued, the system handles the call in a way that allows the calling thread to continue as quickly as possible."

Change Number: XSH/TC2/D6/126 [XSH ERN 129]

On Page: 1330 Line: 41250 Section: shm_open (2003 Ed.)
On Page: 1324 Line: 41014 Section: shm_open (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Creating and mapping a shared memory object

The following code segment demonstrates the use of shm_open() to create a shared memory object which is then sized using ftruncate() before being mapped into the process address space using mmap():

```
#include <unistd.h>
#include <sys/mman.h>
  ...

 #define MAX_LEN 10000
 struct region {        /* Defines "structure" of shared memory */
     int len;
     char buf[MAX_LEN];
 };
 struct region *rptr;
 int fd;

 /* Create shared memory object and set its size */

 fd = shm_open("/myregion", O_CREAT | O_RDWR, S_IRUSR | S_IWUSR);
 if (fd == -1)
     /* Handle error */;

 if (ftruncate(fd, sizeof(struct region)) == -1)
     /* Handle error */;

 /* Map shared memory object */

 rptr = mmap(NULL, sizeof(struct region),
         PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
```

```
2726  if (rptr == MAP_FAILED)
2727      /* Handle error */;

2728  /* Now we can refer to mapped region using fields of rptr,
2729     for example, rptr->len */
2730   ..."
```

2731 **Change Number: XSH/TC2/D6/127** [XSH ERN 93]

2732 On Page: 1344 Line: 41688-41696 Section: sigaction (2003 Ed.)
2733 On Page: 1338 Line: 41449-41458 Section: sigaction (2001 Ed.)

2734 In the DESCRIPTION
2735 Change From:
2736 "If the SA_SIGINFO flag (see below) is cleared in the sa_flags field of
2737 the sigaction structure, the sa_handler field identifies the
2738 action to be associated with the specified signal. [XSI|RTS] If the SA_SIGINFO
2739 flag is set in the sa_flags field, and the implementation supports the
2740 Realtime Signals Extension option or the XSI Extension option,
2741 the sa_sigaction field specifies a signal-catching function. [/XSI|RTS]
2742 If the SA_SIGINFO bit is cleared and the sa_handler field specifies a
2743 signal-catching function, or if the SA_SIGINFO bit is set, the sa_mask
2744 field identifies a set of signals that shall be added to the signal
2745 mask of the thread before the signal-catching function is invoked. If
2746 the sa_handler field specifies a signal-catching function, the sa_mask
2747 field identifies a set of signals that shall be added to the process
2748 signal mask before the signal-catching function is invoked."

2749 To:
2750 "If the SA_SIGINFO flag (see below) is cleared in the sa_flags field of
2751 the sigaction structure, the sa_handler field identifies the
2752 action to be associated with the specified signal. [XSI|RTS] If the SA_SIGINFO
2753 flag is set in the sa_flags field, and the implementation supports the
2754 Realtime Signals Extension option or the XSI Extension option,
2755 the sa_sigaction field specifies a signal-catching function. [/XSI|RTS]

2756 Rationale:
2757 The removed text repeats requirements stated later in the
2758 DESCRIPTION section on page 1346 line 41765 (2003 Ed.);
2759 page 1340 line 41527 (2001 Ed.).

2760 **Change Number: XSH/TC2/D6/128** [XSH ERN 33,34,35,36,37]

2761 On Page: 1345-1348 Line: 41736,41761,41843 Section: sigaction (2003 Ed.)
2762 On Page: 1339-1342 Line: 41498,41523,41605 Section: sigaction (2001 Ed.)

2763 In the DESCRIPTION of SA_SIGINFO, line 41736 (2003 Ed.); line 41498 (2001 Ed.)
2764 Change From:
2765 "The second argument shall point to an object of type siginfo_t explaining
2766 the reason why the signal was generated; the third argument can be cast
2767 to a pointer to an object of type ucontext_t to refer to the receiving
2768 process context that was interrupted when the signal was delivered."

2769 To:
2770 "The second argument shall point to an object of type siginfo_t explaining
2771 the reason why the signal was generated; the third argument can be cast
2772 to a pointer to an object of type ucontext_t to refer to the receiving
2773 thread's context that was interrupted when the signal was delivered."

2774 In the DESCRIPTION of SA_NODEFER, line 41761 (2003 Ed.); line 41523 (2001 Ed.)
2775 Change From:
2776 "If set and sig is caught, sig shall not be added to the process
2777 signal mask on entry to the signal handler unless it is included in
2778 sa_mask.Otherwise, sig shall always be added to the process signal mask
2779 on entry to the signal handler."

2780 To:
2781 "If set and sig is caught, sig shall not be added to the thread's

signal mask on entry to the signal handler unless it is included in sa_mask.Otherwise, sig shall always be added to the thread's signal mask on entry to the signal handler."

In the APPLICATION USAGE , line 41843 (2003 Ed.); line 41605 (2001 Ed.)
Change From:
"When the signal handler returns, the receiving process resumes execution at the point it was interrupted unless the signal handler makes other arrangements. If longjmp() or _longjmp() is used to leave the signal handler, then the signal mask must be explicitly restored by the process."

To:
"When the signal handler returns, the receiving thread resumes execution at the point it was interrupted unless the signal handler makes other arrangements. If longjmp() or _longjmp() is used to leave the signal handler, then the signal mask must be explicitly restored."

Change Number: XSH/TC2/D6/129 [XSH ERN 130]

On Page: 1347 Line: 41807 Section: sigaction (2003 Ed.)
On Page: 1341 Line: 41569 Section: sigaction (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Establishing a signal handler

The following example demonstrates the use of sigaction to establish a handler for the SIGINT signal.

```
#include <signal.h>

 static void handler(int signum)
 {
     /* Take appropriate actions for signal delivery */
 }

 int main()
 {
     struct sigaction sa;

     sa.sa_handler = handler;
     sigemptyset(&sa.sa_mask);
     sa.sa_flags = SA_RESTART;     /* Restart functions if
                                      interrupted by handler */
     if (sigaction(SIGINT, &sa, NULL) == -1)
         /* Handle error */;

     /* Further code */
 }"
```

Change Number: XSH/TC2/D6/130 [XSH ERN 38]

On Page: 1377 Line: 42765-42766 Section: sigtimedwait (2003 Ed.)
On Page: 1371 Line: 42520-42521 Section: sigtimedwait (2001 Ed.)

Change From:
"An implementation only checks for this error if no signal is pending in set and it is necessary to wait."

To:
"An implementation should only check for this error if no signal is pending in set and it is necessary to wait."

Rationale: This restores the wording from the original base document.

Change Number: XSH/TC2/D6/131 [XSH ERN 94]

On Page: 1380 Line: 42881-42883 Section: sigwait (2003 Edition)
On Page: 1374 Line: 42636-42638 Section: sigwait (2001 Edition)

In the DESCRIPTION section
Change From:
"If more than one thread is using sigwait() to wait for the same signal, no more than one of these threads shall return from sigwait() with the signal number. Which thread returns from sigwait() if more than a single thread is waiting is unspecified."

To:
"If more than one thread is using sigwait() to wait for the same signal, no more than one of these threads shall return from sigwait() with the signal number. If more than a single thread is blocked in sigwait() for a signal when that signal is generated for the process, it is unspecified which of the waiting threads returns from sigwait(). If the signal is generated for a specific thread, as by pthread_kill(), only that thread shall return."

Change Number: XSH/TC2/D6/132 [XSH ERN 38]

On Page: 1388 Line: 43101-43103 Section: sleep (2003 Ed.)
On Page: 1382 Line: 42856-42858 Section: sleep (2001 Ed.)

In the RATIONALE section
Change From:
"One is to use the alarm() function to schedule a SIGALRM signal and then suspend the process waiting for that signal."

To:
"One is to use the alarm() function to schedule a SIGALRM signal and then suspend the calling thread waiting for that signal."

Change Number: XSH/TC2/D6/133 [XSH ERN 131]

On Page: 1443 Line: 44720 Section: strptime (2003 Ed.)
On Page: 1436 Line: 44467 Section: strptime (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Convert a data-plus-time string to broken-down time and then into seconds

The following example demonstrates the use strptime() to convert a
string into broken-down time. The broken-down time is then converted
into seconds since the Epoch using mktime().

```
#include <time.h>
 ...

 struct tm tm;
 time_t t;

 if (strptime("6 Dec 2001 12:33:45", "%d %b %Y %H:%M:%S", &tm) == NULL)
     /* Handle error */;

 printf("year: %d; month: %d; day: %d;\n",
         tm.tm_year, tm.tm_mon, tm.tm_mday);
 printf("hour: %d; minute: %d; second: %d\n",
         tm.tm_hour, tm.tm_min, tm.tm_sec);
 printf("week day: %d; year day: %d\n", tm.tm_wday, tm.tm_yday);

 tm.tm_isdst = -1;           /* Not set by strptime(); tells mktime()
                                to determine if daylight saving time
                                is in effect */
 t = mktime(&tm);
 if (t == -1)
     /* Handle error */;
 printf("seconds since the Epoch: %ld\n", (long) t);"
```

Change Number: XSH/TC2/D6/134 [XSH ERN 105]

On Page: 1474 Line: 45617 Section: sysconf (2003 Ed.)
(not present in the 2001 Edition)

In the DESCRIPTION section
Delete the following entry from the table

"_POSIX_SYMLOOP_MAX _SC_SYMLOOP_MAX"

Rationale:
There is an entry for SYMLOOP_MAX in this table, there thus does not
need to be an entry for _POSIX_SYMLOOP_MAX (the minimum maximum) here.
This corrects an error in Technical Corrigendum 1.

Change Number: XSH/TC2/D6/135 [XSH ERN 40,61,62]

On Page: 1474-1476 Line: 45593,45603,45645,45691,45711-45718 Section: sysconf (2003 Ed.)
On Page: 1467-1469 Line: 45341,45350,45390,45434,45452-45459 Section: sysconf (2001 Ed.)

In the DESCRIPTION, line 45593 (2003 Ed.); line 45341 (2001 Ed.)
Delete the following entry from the table
"_POSIX_FILE_LOCKING _SC_FILE_LOCKING"

In the DESCRIPTION, line 45603 (2003 Ed.); line 45350 (2001 Ed.)
Delete the following entry from the table

"_POSIX_MULTI_PROCESS _SC_MULTI_PROCESS"

In the DESCRIPTION, line 45645 (2003 Ed.); line 45390 (2001 Ed.)
Delete the following entry from the table

"_POSIX2_C_VERSION _SC_2_C_VERSION"

In the DESCRIPTION, line 45691 (2003 Ed.); line 45434 (2001 Ed.)
Delete the following entry from the table
"_XOPEN_XCU_VERSION _SC_XOPEN_XCU_VERSION"

In the APPLICATION USAGE, lines 45711-45718 (2003 Ed.); lines 45452-45459
(2001 Ed.)
Delete the paragraph

"If the value of sysconf(_SC_2_VERSION) is not equal to the value of
the _POSIX2_VERSION symbolic constant, the utilities available via
system() or popen() might not behave as described in the Shell and
Utilities volume of IEEE Std 1003.1-2001. This would mean that the
application is not running in an environment that conforms to the Shell
and Utilities volume of IEEE Std 1003.1-2001. Some applications might be
able to deal with this, others might not. However, the functions defined
in this volume of IEEE Std 1003.1-2001 continue to operate as specified,
even if sysconf(_SC_2_VERSION) reports that the utilities no longer
perform as specified."

Rationale: _POSIX_FILE_LOCKING and _POSIX_MULTI_PROCESS were erroneous
entries. _POSIX2_C_VERSION was removed by P1003.1a/P1003.2b and should
have been removed here too. _XOPEN_XCU_VERSION should have been removed
as per The Open Group Base Working Group resolution bwg2001-006.

Change Number: XSH/TC2/D6/136 [XBD ERN 27]

On Page: 1474 Line: 45616,45632,45635,45701,45718 Section: sysconf (2003 Ed.)
On Page: 1467 Line: 45362,45377,45380,45442,45459 Section: sysconf (2001 Ed.)

In the DESCRIPTION section:
Add into the table after the _POSIX_SPORADIC_SERVER entry:
"_POSIX_SS_REPL_MAX _SC_SS_REPL_MAX"

Add into the table after the _POSIX_TRACE_EVENT_FILTER entry:
"_POSIX_TRACE_EVENT_NAME_MAX _SC_TRACE_EVENT_NAME_MAX"

Add into the table after the _POSIX_TRACE_MEMORY_OBJECTS entry:
"_POSIX_TRACE_NAME_MAX _SC_TRACE_NAME_MAX
 _POSIX_TRACE_SYS_MAX _SC_TRACE_SYS_MAX
 _POSIX_TRACE_USER_EVENT_MAX _SC_TRACE_USER_EVENT_MAX"

Add to the end of the RETURN VALUE section as a new paragraph:
"If the variable corresponding to name is dependent on an unsupported
option the results are unspecified."

Add to the end of the APPLICATION USAGE section:
"Application writers should check whether an option, such as _POSIX_TRACE
is supported, prior to obtaining and using values for related
variables such as _POSIX_TRACE_NAME_MAX."

Change Number: XSH/TC2/D6/137 [XSH ERN 107]

On Page: 1475 Line: 45659 Section: sysconf (2003 Ed.)
On Page: 1468 Line: 45403 Section: sysconf (2001 Ed.)

In the DESCRIPTION section
Delete the following entry from the table:
"_REGEX_VERSION _SC_REGEX_VERSION"

Rationale:
The text in P1003.1a indicates that this was to be used for versioning information related to the _SC_REGEXP_VERSION (sic)

(.1a D17): "172 POSIX.2RE: The version of ISO/IEC 9945-2 that defines _POSIX2_VERSION to be the value indicated by the return value of sysconf(_SC_REGEXP_VERSION)."

Given that the approach for "_SC_2_C_VERSION" and "_SC_XOPEN_XCU_VERSION" has been to remove them since they are unnecessary this is removed for consistency.

Change Number: XSH/TC2/D6/138 [XSH ERN 95]

On Page: 1521,1522 Line: 47034,47050 Section: timer_create (2003 Ed.)
On Page: 1514,1515 Line: 46762,46780 Section: timer_create (2001 Ed.)

In the DESCRIPTION section
Add as a new last paragraph, shaded and margin marked "TSA":
"If evp->sigev_sigev_notify is SIGEV_THREAD and sev->sigev_notify_attributes is not NULL, if the attribute pointed to by sev->sigev_notify_attributes has a thread stack address specified by a call to pthread_attr_setstack() or pthread_attr_setstackaddr() the results are unspecified if the signal is generated more than once."

In the APPLICATION USAGE section:
Change From:
"None."

To:
"If a timer is created which has evp->sigev_sigev_notify set to SIGEV_THREAD and the attribute pointed to by evp->sigev_notify_attributes has a thread stack address specified by a call to either pthread_attr_setstack() or pthread_attr_setstackaddr(), the memory dedicated as a thread stack can not be recovered. The reason for this is that the threads created in response to a timer expiration are created detached, or in an unspecified way if the thread attribute's detachstate is PTHREAD_CREATE_JOINABLE. In neither case is it valid to call pthread_join(), which makes it impossible to determine the lifetime of the created thread which thus means the stack memory can not be reused."

Change Number: XSH/TC2/D6/139 [XSH ERN 74]

On Page: 1524 Line: 47131 Section: timer_delete (2003 Ed.)
On Page: 1517 Line: 46859 Section: timer_delete (2001 Ed.)

In the ERRORS section
Change From:
"The timer_delete() function shall fail if:"

To:
"The timer_delete() function may fail if:"

Change Number: XSH/TC2/D6/140 [XSH ERN 75]

On Page: 1526 Line: 47211-47213 Section: timer_getoverrun (2003 Ed.)
On Page: 1519 Line: 46939-46941 Section: timer_getoverrun (2001 Ed.)

Change From:
"The timer_getoverrun(),timer_gettime(), and timer_settime() functions shall fail if:

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

To:
"The timer_getoverrun(),timer_gettime(), and timer_settime() 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()."

Notes to Editors:
Move the revised text section after the current following [EINVAL] error so that the "shall fail" case occurs before the "may fail" cases.

Change Number: XSH/TC2/D6/141 [XSH ERN 95]

On Page: 1526 Line: 47217,47221 Section: timer_getoverrun (2003 Ed.)
On Page: 1519 Line: 46945,46949 Section: timer_getoverrun (2001 Ed.)

In the ERRORS section
Add the following additional text

"The timer_settime() function may fail if:

[EINVAL] The it_interval member of value is not zero and the timer was created with notification by creation of a new thread (sigev_sigev_notify was SIGEV_THREAD) and a fixed stack address has been set in the thread attribute pointed to by sigev_notify_attributes."

In the APPLICATION USAGE section
Change From:
"None."

To:
"Using fixed stack addresses is problematic when timer expiration is signalled by the creation of a new thread. Since it can not be assumed that the thread created for one expiration is finished before the next expiration of the timer, it could happen that two threads use the same memory as stack at the same time. This is invalid and produces undefined results."

Change Number: XSH/TC2/D6/142 [XSH ERN 137]

On Page: 1533 Line: 47433-47434 Section: tmpnam (2003 Ed.)
On Page: 1526 Line: 47161-47162 Section: tmpnam (2001 Ed.)

In the DESCRIPTION section
Change From:
"The implementation shall behave as if no function defined in this volume of IEEE Std 1003.1-2001 calls tmpnam()."

To:
"The implementation shall behave as if no function defined in this volume of IEEE Std 1003.1-2001 except tempnam() calls tmpnam()."

Change Number: XSH/TC2/D6/143 [XSH ERN 52,98]

On Page: 1555 Line: 48027-48028,48051 Section: umask (2003 Ed.)
On Page: 1548 Line: 47755-47756,47779 Section: umask (2001 Ed.)

In the DESCRIPTION section
Change From:
"The process' file mode creation mask is used during open(), creat(), mkdir(), and mkfifo() to turn off permission bits in the mode

argument supplied."

To:
"The process' file mode creation mask is used during open(), creat(), mkdir(), mkfifo(), mknod(), mq_open() and sem_open() to turn off permission bits in the mode argument supplied."

(Ed note:mq_open is shaded and margin marked MSG, sem_open is shaded and margin marked SEM, mknod is shaded and margin marked XSI)

In the SEE ALSO section
Add "mknod(), mq_open()" and "sem_open()" in alphabetical order.

Change Number: XSH/TC2/D6/144 [XSH ERN 41]

On Page: 1568 Line: 48450-48452,48465 Section: usleep (2003 Ed.)
On Page: 1561 Line: 48178-48180,48193 Section: usleep (2001 Ed.)

Change From:
"It is also unspecified whether the SIGALRM signal is blocked, unless the process signal mask is restored as part of the environment."

To:
"It is also unspecified whether the SIGALRM signal is blocked, unless the thread's signal mask is restored as part of the environment."

Add as a new final paragraph to the DESCRIPTION:

"The usleep() function need not be reentrant. A function that is not required to be reentrant is not required to be thread-safe."

Change Number: XSH/TC2/D6/145 [XSH ERN 116]

On Page: 1592 Line: 49061 Section: wait (2003 Ed.)
On Page: 1585 Line: 48789 Section: wait (2001 Ed.)

In the EXAMPLES section
Change From:
"None."

To:
"Waiting for a child process and then checking its status

The following example demonstrates the use of waitpid(), fork(), and the macros used to interpret the status value returned by waitpid() (and wait()). The code segment creates a child process which does some unspecified work. Meanwhile the parent loops performing calls to waitpid() to monitor the status of the child. The loop terminates when child termination is detected.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
 ...

 pid_t child_pid, wpid;
 int status;

 child_pid = fork();
 if (child_pid == -1) {      /* fork() failed */
     perror("fork");
     exit(EXIT_FAILURE);
 }

 if (child_pid == 0) {       /* This is the child */
     /* Child does some work and then terminates */
     ...
```

```
3100  } else {                   /* This is the parent */
3101      do {
3102          wpid = waitpid(child_pid, &status, WUNTRACED
3103  #ifdef WCONTINUED       /* Not all implementations support this */
3104                                  | WCONTINUED
3105  #endif
3106                          );
3107          if (wpid == -1) {
3108              perror("waitpid");
3109              exit(EXIT_FAILURE);
3110          }

3111          if (WIFEXITED(status)) {
3112              printf("child exited, status=%d\n", WEXITSTATUS(status));

3113          } else if (WIFSIGNALED(status)) {
3114              printf("child killed (signal %d)\n", WTERMSIG(status));

3115          } else if (WIFSTOPPED(status)) {
3116              printf("child stopped (signal %d)\n", WSTOPSIG(status));

3117  #ifdef WIFCONTINUED     /* Not all implementations support this */
3118          } else if (WIFCONTINUED(status)) {
3119              printf("child continued\n");
3120  #endif
3121          } else {    /* Non-standard case -- may never happen */
3122              printf("Unexpected status (0x%x)\n", status);
3123          }
3124      } while (!WIFEXITED(status) && !WIFSIGNALED(status));
3125  }
```

3126 **Change Number: XSH/TC2/D6/146** [XSH ERN 166]

3127 On Page: 1659 Line: 51172-51173 Section: write (2003 Ed.)
3128 On Page: 1652 Line: 50898-50899 Section: write (2001 Ed.)

3129 In the ERRORS section
3130 Change From:
3131 "In the latter case, if the socket is of type SOCK_STREAM, the SIGPIPE
3132 signal is generated to the calling process."

3133 To:
3134 "In the latter case, if the socket is of type SOCK_STREAM, a SIGPIPE
3135 signal shall also be sent to the thread."

3136 **Change Number: XSH/TC2/D6/147** [XSH ERN 42]

3137 On Page: 1660 Line: 51223 Section: write (2003 Ed.)
3138 On Page: 1653 Line: 50949 Section: write (2001 Ed.)

3139 In the RATIONALE section
3140 Change From:
3141 "Otherwise, the process may block; that is, pause until enough space is
3142 available for writing."

3143 To:
3144 "Otherwise, the calling thread may block; that is, pause until enough space is
3145 available for writing."

3146 **Change Number: XSH/TC2/D6/148** [XSH ERN 108]

3147 On Page: 1667 Line: 51442,51453 Section: y0 (2003 Ed.)
3148 On Page: 1660 Line: 51168,51179 Section: y0 (2001 Ed.)

3149 In the RETURN VALUE section
3150 Change From:
3151 "If x is 0.0, -HUGE_VAL shall be returned and a range error
3152 may occur".

3153 To:
3154 "If x is 0.0, -HUGE_VAL shall be returned and a pole error may occur."

3155 In the ERRORS section
3156 Insert before line 51453 (2003 Ed.); line 51179 (2001 Ed.):
3157 "Pole error   The value of x is zero.

3158   If the integer expression (math_errhandling & MATH_ERRNO) is
3159   non-zero, then errno shall be set to [ERANGE]. If the integer
3160   expression (math_errhandling & MATH_ERREXCEPT) is non-zero,
3161   then the divide-by-zero floating-point exception shall be raised."

3162 In the ERRORS section, line 51453 (2003 Ed.); line 51179 (2001 Ed.)
3163 Change From:
3164 "Range Error The value of x is 0.0, or the correct result would cause
3165 overflow."

3166 To:
3167 "Range Error The correct result would cause overflow."

3168 Rationale:
3169 The changes are made for consistency with the general rules stated in
3170 "Treatment of Error Conditions for Mathematical Functions" in XBD.

4. Changes to Shell and Utilities

This section contains the set of changes to the text of the Shell and Utilities.

Change Number: XCU/TC2/D6/1 [XBD ERN 20]

On Page: xxxii Line: "ISO/IEC 8859" Section: Referenced Documents (2003 Ed.)
On Page: xxviii Line: "ISO/IEC 8859" Section: Referenced Documents (2001 Ed.)

Add after line starting "Part 10":

"Part 11: Latin/Thai Alphabet"

Add after line starting "Part 15":

"Part 16: Latin Alphabet No. 10"

Change Number: XCU/TC2/D6/2 [XCU ERN 29]

On Page: 5 Line: 139-150,170 Section: 1.7.1.4 (2003 Ed.)
On Page: 5 Line: 139-150,170 Section: 1.7.1.4 (2001 Ed.)

In Table 1-1, change the dash (-) entries to letter "U" on lines
140, 141 and 150 (rows for types B, C and T respectively).

Change all the dash (-) entries in the T column to letter "U".

Add after line 170
"U The effect is unspecified unless specified by the utility description."

Change Number: XCU/TC2/D6/3 [XRAT ERN 2]

On Page: 41 Line: 1704 Section: 2.6.4 (2003 Ed.)
On Page: 41 Line: 1700 Section: 2.6.4 (2001 Ed.)

Add a new paragraph before the existing paragraph that
commences "As an extension, the shell may recognize arithmetic expressions
beyond those listed...":

"All changes to variables in an arithmetic expression shall be in effect
after the arithmetic expansion, as in the parameter expansion
${x=value}.

If the shell variable x contains a value that forms a valid
integer constant then the arithmetic expansions $((x)) and $(($x))
shall return the same value."

Change Number: XCU/TC2/D6/4 [XCU ERN 17]

On Page: 48 Line: 1986-1990 Section: 2.9.1.1 (2003 Ed.)
On Page: 48 Line: 1982-1986 Section: 2.9.1.1 (2001 Ed.)

In Section 2.9.1.1 Command Search and Execution
Change From:
"If the execve() function fails due to an error equivalent to the
[ENOEXEC] error defined in the System Interfaces volume of IEEE Std
1003.1-2001, the shell shall execute a command equivalent to having
a shell invoked with the command name as its first operand, with any
remaining arguments passed to the new shell."

To:
"If the execve() function fails due to an error equivalent to the
[ENOEXEC] error defined in the System Interfaces volume of IEEE Std
1003.1-2001, the shell shall execute a command equivalent to having a
shell invoked with the pathname resulting from the search as its first
operand, with any remaining arguments passed to the new shell, except

that the value of $0 in the new shell may be set to the command name."

Change Number: XCU/TC2/D6/5 [XCU ERN 30]

On Page: 212 Line: 8323-8324 Section: c99 (2003 Ed.)
On Page: 212 Line: 8243-8244 Section: c99 (2001 Ed.)

In the OPTIONS section
Change From:
"If a directory specified by the -L option contains files named libc.a,
libm.a, libl.a, or liby.a, the results are unspecified."

To:
"If a directory specified by the -L option contains files with names
starting with any of the strings "libc.", "libl.", "libpthread.",
"libm.", "librt.", "libtrace.", "libxnet.", or "liby.", the results
are unspecified."

Change Number: XCU/TC2/D6/6 [XCU ERN 19]

On Page: 215,218 Line: 8466-8467,8583-8588 Section: c99 (2003 Ed.)
On Page: 215,218 Line: 8385-8386,8502-8507 Section: c99 (2001 Ed.)

In the EXTENDED DESCRIPTION section, Programming Environments
Change From:
"shall be output by a getconf command using the
_POSIX_V6_WIDTH_RESTRICTED_ENVS argument."

To:
"shall be output by a getconf command using the
POSIX_V6_WIDTH_RESTRICTED_ENVS argument, as a <newline>-separated list
of names suitable for use with the getconf -v option."

(Note the removed leading underscore on POSIX_V6_WIDTH_RESTRICTED_ENVS.)

In the EXAMPLES section
Change From:
"# ... if there are no additional constraints, the first one will do:
CENV=$(getconf _POSIX_V6_WIDTH_RESTRICTED_ENVS | head -n l)

... or, if an environment that supports large files is preferred,
look for names that contain "OFF64" or "OFFBIG". (This chooses
the last one in the list if none match.)
for CENV in $(getconf _POSIX_V6_WIDTH_RESTRICTED_ENVS)"

To:
"# ... if there are no additional constraints, the first one will do:
CENV=$(getconf POSIX_V6_WIDTH_RESTRICTED_ENVS | head -n l)

... or, if an environment that supports large files is preferred,
look for names that contain "OFF64" or "OFFBIG". (This chooses
the last one in the list if none match.)
for CENV in $(getconf POSIX_V6_WIDTH_RESTRICTED_ENVS) "

Rationale:
This change corrects the name of the variable not to include the
underscore.

Change Number: XCU/TC2/D6/7 [XCU ERN 33]

On Page: 343 Line: 13238-13240,13451 Section: ed (2003 Ed.)
On Page: 341 Line: 13113-13115,13326 Section: ed (2001 Ed.)

In the EXTENDED DESCRIPTION section (List command)
Delete the sentence:
"If the size of a byte on the system is greater than nine bits, the format used for non-printable characters is implementation-defined."

In the RATIONALE section:
Change From:
"Systems with bytes too large to fit into three octal digits must devise other means of displaying non-printable characters. Consideration was given to requiring that the number of octal digits be large enough to hold a byte, but this seemed to be too confusing for applications on the vast majority of systems where three digits are adequate. It would be theoretically possible for the application to use the getconf utility to find out the CHAR_BIT value and deal with such an algorithm; however, there is really no portable way that an application can use the octal values of the bytes across various coded character sets, so the additional specification was not worthwhile."

To:
"Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/8 [XCU ERN 31]

On Page: 370 Line: 14333-14334 Section: ex (2003 Ed.)
On Page: 368 Line: 14205-14206 Section: ex (2001 Ed.)

In the EXTENDED DESCRIPTION section
Change From:
"Implementations may restrict the set of characters accepted in lhs or rh, except that printable characters and <blank>s shall not be restricted. Additional restrictions shall be implementation-defined."

To:
"Implementations may restrict the set of characters accepted in lhs or rhs, except that printable characters and <blank>s shall not be restricted. Additional restrictions shall be implementation-defined."

Note to editor: The change is from "rh" to "rhs".

Change Number: XCU/TC2/D6/9 [XCU ERN 33]

On Page: 380 Line: 14705-14709,16154 Section: ex (2003 Ed.)
On Page: 378 Line: 14577-14581,16025 Section: ex (2001 Ed.)

In the EXTENDED DESCRIPTION section (Print command)
Delete the sentence:
"If the size of a byte on the system is greater than 9 bits, the format used for non-printable characters is implementation-defined."

Add to the RATIONALE section:
"Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/10 [XCU ERN 26,28]

On Page: 447-448 Line: 17290-17292,17297-17300,17317-17322 Section: file (2003 Ed.)
On Page: 445-446 Line: 17135-17137,17142-17147,17162-17167 Section: file (2001 Ed.)

On lines 17290-17292 (2003 Ed.); lines 17135-17137 (2001 Ed.)
Change From:
"The type shall consist of the type specification characters c, d, f, s, and u, specifying character, signed decimal, floating point, string, and unsigned decimal, respectively."

To:
"The type shall consist of the type specification characters d, s, and u, specifying signed decimal, string, and unsigned decimal, respectively."

On lines 17297-17300 (2003 Ed.); lines 17142-17147 (2001 Ed.)
Change From:
"The type specification characters d, f, and u can be followed by an optional unsigned decimal integer that specifies the number of bytes represented by the type. The type specification character f can be followed by an optional F, D, or L, indicating that the value is of type float, double, or long double, respectively."

To:
"The type specification characters d and u can be followed by an optional unsigned decimal integer that specifies the number of bytes represented by the type."

On lines 17317-17322 (2003 Ed.); lines 17162-17167 (2001 Ed.)
Delete the paragraph:
"For the type specifier f , the default number of bytes shall correspond to the number of bytes in the basic double precision floating-point data type of the underlying implementation. The implementation shall support values of the optional number of bytes to be converted corresponding to the number of bytes in the C-language types float, double, and long double. These numbers can also be specified by an application as the characters F , D , and L , respectively."

Change Number: XCU/TC2/D6/11 [XCU ERN 27]

On Page: 447-448 Line: 17280 Section: file (2003 Ed.)
On Page: 445-446 Line: 17125 Section: file (2001 Ed.)

In the EXTENDED DESCRIPTION section
On line 17280 (2003 Ed.); line 17125 (2001 Ed.)
Change From:
"Each line shall be composed of the following four <blank>-separated fields:"

To:
"Each line shall be composed of the following four <tab>-separated* fields:

Footnote:* Implementations may allow any combination of one or more white space characters other than <newline> to act as field separators."

On lines 17336-17341 (2003 Ed.); lines 17181-17186 (2001 Ed.)
Change From:
"\character
The backslash-escape sequences as specified in the Base Definitions volume of IEEE Std 1003.1-2001, Table 5-1, Escape Sequences and Associated Actions ('\\' , '\a' , '\b' , '\f' , '\n' , '\r' , '\t' , '\v'). The results of using any other character, other than an octal digit, following the backslash are unspecified."

To:
"\character

The backslash-escape sequences as specified in the Base Definitions volume of IEEE Std 1003.1-2001, Table 5-1, Escape Sequences and Associated Actions ('\\' , '\a' , '\b' , '\f' , '\n' , '\r' , '\t' , '\v'). In addition, the escape sequence '\ ' (the <backslash> character followed by the <space> character) shall be recognized to represent a <space> character. The results of using any other character, other than an octal digit, following the backslash are unspecified."

Change Number: XCU/TC2/D6/12 [XCU ERN 33]

On Page: 448 Line: 17345-17347,17462 Section: file (2003 Ed.)
On Page: 446 Line: 17190-17192,17278 Section: file (2001 Ed.)

In the EXTENDED DESCRIPTION section
Delete the sentence:
"If the size of a byte on the system is greater than 9 bits, the valid escape sequence used to represent a byte implementation-defined."

In the RATIONALE section
Add the following text at the end:
"Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/13 [XCU ERN 4]

On Page: 458 Line: 17761-17762,17765-17766 Section: find (2003 Ed.)
On Page: 455 Line: 17572-17573,17576-17577 Section: find (2001 Ed.)

In the RATIONALE section
Change From:
"The descriptions of -atime, -ctime, and -mtime were changed from the SVID descrption of n ''days'' to ''24-hour period''."

To:
"The descriptions of -atime, -ctime, and -mtime were changed from the SVID descrption of n ''days'' to n being the result of the integer division of the time difference in seconds by 86400".

Change From:
"For example, -atime 3 is true if the file was accessed any time in the period from 72 hours to 48 hours ago."

To:
"For example, -atime 2 is true if the file was accessed any time in the period from 72 hours to 48 hours ago."

Rationale:
The rationale for -atime/-mtime/-ctime needs updating to match changes made in the normative text since POSIX.2-1992.

Change Number: XCU/TC2/D6/14 [XCU ERN 33]

On Page: 540 Line: 20802-20806,21020-21023 Section: lex (2003 Ed.)
On Page: 537 Line: 20593-20598,20811-20814 Section: lex (2001 Ed.)

In the EXTENDED DESCRIPTION section (Table 4-10)
Delete the sentence:
"If the size of a byte on the system is greater than nine bits, the escape sequence used to represent a byte is implementation-defined."

In the RATIONALE section
Change from:
"The description of octal and hexadecimal-digit escape sequences agrees with the ISO C standard use of escape sequences. See the RATIONALE for ed for a discussion of bytes larger than 9 bits being represented by octal values. Hexadecimal values can represent larger bytes and multi-byte characters directly, using as many digits as required."

To:
"The description of octal and hexadecimal-digit escape sequences agrees with the ISO C standard use of escape sequences.

Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/15 [XCU ERN 18]

On Page: 559 Line: 21535 Section: localedef (2003 Ed.)
On Page: 556 Line: 21324 Section: localedef (2001 Ed.)

In the OPERANDS section
Change From:
"This capability may be restricted to users with appropriate privileges."

To:
"The ability to create public locales in this way may be restricted to users with appropriate privileges."

Change Number: XCU/TC2/D6/16 [XCU ERN 13]

On Page: 560-561 Line: 21593-21594,21611 Section: localedef (2003 Ed.)
On Page: 557-558 Line: 21382-21383,21400 Section: localedef (2001 Ed.)

In the EXTENDED DESCRIPTION section
Change From:
"If a non-printable character in the charmap has a width specified that is not -1, localedef shall generate a warning."

To:
"If a non-printable character in the charmap has a width specified that is not -1 the result will be undefined."

In the CONSEQUENCES OF ERRORS section
Delete the fourth bullet item:

"* If a non-printable character has a width specified other than -1."

Rationale:
The previous text was conflicting with the descriptions of WIDTH and WIDTH_DEFAULT in XBD.

Change Number: XCU/TC2/D6/17 [XCU ERN 34]

On Page: 593 Line: 22946-22947 Section: mailx (2003 Ed.)
On Page: 590 Line: 22721-22722 Section: mailx (2001 Ed.)

In the EXTENDED DESCRIPTION section (Internal Variables in Mailx)
Change From:
"The defaults specified here may be changed by the implementation-defined system start-up file unless the user specifies the -n option."

To:
"The defaults specified here may be changed by the unspecified system start-up file unless the user specifies the -n option."

Rationale: The change is made for consistency with elsewhere in the EXTENDED DESCRIPTION.

Change Number: XCU/TC2/D6/18 [XCU ERN 35]

On Page: 666 Line: 25769-25772 Section: nice (2003 Ed.)
On Page: 663 Line: 25539-25542 Section: nice (2001 Ed.)

In the RATIONALE section
Delete the paragraph:
"Due to the text about the limits of the nice value being implementation-defined, nice is not actually required to change the nice value of the executed command; the limits could be zero differences from the system default, although the implementor is required to document this fact in the conformance document."

Change Number: XCU/TC2/D6/19 [XCU ERN 33]

On Page: 682 Line: 26379-26380,26539 Section: od (2003 Ed.)
On Page: 679 Line: 26149-26150,26309 Section: od (2001 Ed.)

In the EXTENDED DESCRIPTION section
Delete the sentence:
"If the size of a byte on the system is greater than nine bits, the format used for non-printable characters is implementation-defined."

Add to the RATIONALE section:
"Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/20 [XCU ERN 21]

On Page: 701 Line: 27113 Section: pax (2003 Ed.)
On Page: 698 Line: 26876 Section: pax (2001 Ed.)

In the SYNOPSIS section
Change From:
"pax -r -w[-diklntuvX][-H|-L][-p string]...[-s replstr]..."

To:
"pax -r -w[-diklntuvX][-H|-L][-o options]...[-p string]...[-s replstr]..."

Change From:
"pax [-cdnv][-H|-L][-f archive][-s replstr]...[pattern...]"

To:
"pax [-cdnv][-H|-L][-o options][-f archive][-s replstr]...[pattern...]"

Rationale:
The SYNOPSIS was inconsistent with the normative text.

Change Number: XCU/TC2/D6/21 [XCU ERN 24]

On Page: 702 Line: 27167-27168 Section: pax (2003 Ed.)
On Page: 699 Line: 26930-26931 Section: pax (2001 Ed.)

Change From:
"If the selected archive format supports the specification of linked files, it shall be an error if these files cannot be linked when the archive is extracted."

To:
"If the selected archive format supports the specification of linked files, it shall be an error if these files cannot be linked when the archive is extracted, except that if the files to be linked are symbolic links and the system is not capable of making hard links to symbolic links then separate copies of the symbolic link shall be created instead."

Change Number: XCU/TC2/D6/22 [XCU ERN 22]

On Page: 704 Line: 27266 Section: pax (2003 Ed.)
On Page: 701 Line: 27029 Section: pax (2001 Ed.)

In the OPTIONS section,
Add new text to the end of the subsection for "delete=pattern"
"When multiple -odelete=pattern options are specified, the patterns shall be additive; all keywords matching the specified string patterns shall be omitted from extended header records that pax produces."

Change Number: XCU/TC2/D6/23 [XCU ERN 9]

On Page: 706 Line: 27336-27339 Section: pax (2003 Ed.)
On Page: 703 Line: 27097-27100 Section: pax (2001 Ed.)

In the OPTIONS section
Change From:
"write In read or copy mode, pax shall write the file, translating or truncating the name, regardless of whether this may overwrite an existing file with a valid name. In list mode, pax shall behave identically to the bypass action."

To:
"write In read or copy mode, pax shall write the file, translating the name, regardless of whether this may overwrite an existing file with a valid name. In list mode, pax shall behave identically to the bypass action."

Rationale:
The text currently written in the standard allows the pax utility to truncate pathnames of files being extracted from an archive or copied from one place to another in arbitrary ways, even if this action overwrites existing files. We agree that translation may cause this to occur, but the pax utility should not truncate file names on its own. It should be capable of restoring files even if the pathname in the archive is longer than {PATH_MAX}. (This may require the pax utility to perform one or more change directory operations to get to a point where the file being extracted can be created using a pathname shorter than {PATH_MAX}.) If the underlying file system performs filename truncation (which is not allowed on file systems conforming to this standard), the pax utility need not attempt to prevent or detect the action. But, the pax utility itself should never truncate a pathname component.

Change Number: XCU/TC2/D6/24 [XCU ERN 7]

On Page: 708 Line: 27469 Section: pax (2003 Ed.)
On Page: 705 Line: 27230 Section: pax (2001 Ed.)

In the OPTIONS section
On line 27469 (2003 Ed.); line 27230 (2001 Ed.)
add as a new paragraph before the paragraph beginning
"The options that operate on the names of files..."

"Specifying more than one of the mutually-exclusive options -H and -L shall not be considered an error and the last option specified shall determine the behavior of the utility."

Change Number: XCU/TC2/D6/25 [XCU ERN 8]

On Page: 715 Line: 27723-27727 Section: pax (2003 Ed.)
On Page: 712 Line: 27484-27488 Section: pax (2001 Ed.)

In the EXTENDED DESCRIPTION section (Pax Extended Header)
Delete the paragraph at line 27723 (2003 Ed.); line 27484 (2001 Ed.):

"ctime The file creation time for the following file(s), equivalent

to the value of the st_ctime member of the stat structure for a file, as described by the stat() function. The creation time shall be restored if the process has the appropriate privilege required to do so. The format of the <value> shall be as described in pax Extended Header File Times (on page 717)."

Rationale:
There is a contradiction in the definition of the ctime keyword for pax's extended header, in that the st_ctime member of the stat structure does not refer to a file creation time. No field in the standard stat structure from <sys/stat.h> includes a file creation time.

Change Number: XCU/TC2/D6/26 [XCU ERN 23]

On Page: 720 Line: 27939-27940,27961 Section: pax (2003 Ed.)
On Page: 717 Line: 27700-27701,27722 Section: pax (2001 Ed.)

In the EXTENDED DESCRIPTION section (ustar Interchange Format)
On lines 27939-27940 (2003 Ed.); lines 27700-27701 (2001 Ed.)

Change From:
"Such files are identified by each file having the same device and file serial number."

To:
"Such files are identified by having the same device and file serial numbers, and pathnames that refer to different directory entries. All such files shall be archived as linked files."

Before line 27961 (2003 Ed.); line 27722 (2001 Ed.)
Insert the following paragraph:
"It is unspecified whether files with pathnames that refer to the same directory entry are archived as linked files or as separate files. If they are archived as linked files this means that attempting to extract both pathnames from the resulting archive will always cause an error (unless the -u option is used) because the link cannot be created.

It is unspecified whether files with the same device and file serial numbers being appended to an archive are treated as linked files to members that were in the archive before the append."

Change Number: XCU/TC2/D6/27 [XCU ERN 10]

On Page: 722,724 Line: 28039-28040,28099 Section: pax (2003 Ed.)
On Page: 719,721 Line: 27800-27801,27860 Section: pax (2001 Ed.)

In the EXTENDED DESCRIPTION section (cpio header)
Change From:
"c_nlink Contains the number of links referencing the file at the time the archive was created."

To:
"c_nlink Contains a number greater than or equal to the number of links in the archive referencing the file. If the -a option is used to append to a cpio archive, then the pax utility need not account for the files in the existing part of the archive when calculating the c_nlink values for the appended part of the archive, and need not alter the c_nlink values in the existing part of the archive if additional files with the same c_dev and c_ino values are appended to the archive."

In the APPLICATION USAGE section
Insert before line 28099 (2003 Ed.); line 27860 (2001 Ed.)

"Caution is advised when using the -a option to append to a cpio format archive. If any of the files being appended happen to be given the same c_dev and c_ino values as a file in the existing part of the archive, then they may be treated as links to that file on extraction. Thus it is risky to use -a with cpio

format except when it is done on the same system that the original archive was created on, and with the same pax utility, and in the knowledge that there has been little or no file system activity since the original archive was created that could lead to any of the files being appended being given the same c_dev and c_ino values as an unrelated file in the existing part of the archive. Also, when (intentionally) appending additional links to a file in the existing part of the archive, the c_nlink values in the modified archive can be smaller than the number of links to the file in the archive, which may mean that the links are not preserved on extraction."

Change Number: XCU/TC2/D6/28 [XCU ERN 33]

On Page: 845 Line: 32699-32701,32892 Section: sed (2003 Ed.)
On Page: 842 Line: 32450-32452,32643 Section: sed (2001 Ed.)

In the EXTENDED DESCRIPTION section
Delete the sentence:
"If the size of a byte on the system is greater than nine bits, the format used for non-printable characters is implementation-defined."

Add to the RATIONALE section:
"Prior versions of this standard allowed for implementations with bytes other than eight bits, but this has been modified in this version."

Change Number: XCU/TC2/D6/29 [XCU ERN 11]

On Page: 846 Line: 32770-32771 Section: sed (2003 Ed.)
On Page: 843 Line: 32521-32522 Section: sed (2001 Ed.)

Change From:
"If the delimiter is not n, within string1 and string2, the delimiter itself can be used as a literal character if it preceeded by a backslash."

To:
"If the delimiter is not 'n', within string1 and string2, the delimiter itself can be used as a literal character if it preceeded by a backslash."

Note to editor the change is from italic n to a quoted character n.

Change Number: XCU/TC2/D6/30 [XCU ERN 14]

On Page: 907-905 Line: 35124-35169 Section: test (2003 Ed.)
On Page: 904-905 Line: 34872-34917 Section: test (2001 Ed.)

In the OPERANDS section
Change From:
"-b file
    True if file exists and is a block special file.

-c file
    True if file exists and is a character special file.

-d file
    True if file exists and is a directory.

-e file
    True if file exists.

-f file
    True if file exists and is a regular file.

-g file
    True if file exists and its set-group-ID flag is set.

```
3662 -h  file
3663     True if file exists and is a symbolic link.

3664 -L  file
3665     True if file exists and is a symbolic link.

3666 -n  string
3667     True if the length of string is non-zero.

3668 -p  file
3669     True if file is a FIFO.

3670 -r  file
3671     True if file exists and is readable. True shall indicate that
3672     permission to read from file will be granted, as defined in Section
3673     1.7.1.4 (on page 4).

3674 -S  file
3675     True if file exists and is a socket.

3676 -s  file
3677     True if file exists and has a size greater than zero.

3678 -t  file_descriptor
3679     True if the file whose file descriptor number is file_descriptor is
3680     open and is associated with a terminal.

3681 -u  file
3682     True if file exists and its set-user-ID flag is set.

3683 -w  file
3684     True if file exists and is writable. True shall indicate that
3685     permission to write from file will be granted, as defined in
3686     Section 1.7.1.4 (on page 4).

3687 -x  file
3688     True if file exists and is executable. True shall indicate that
3689     permission to execute file will be granted, as defined in Section
3690     1.7.4 (on page 4).  If file is a directory, true shall indicate that
3691     permission to search file will be granted.

3692 -z  string
3693     True if the length of string string is zero.

3694 string
3695     True if the string string is not the null string.

3696 s1 =  s2
3697     True if the strings s1 and s2 are identical.

3698 s1 !=  s2
3699     True if the strings s1 and s2 are not identical.

3700 n1 -eq  n2
3701     True if the integers n1 and n2 are algebraically equal.

3702 n1 -ne  n2
3703     True if the integers n1 and n2 are not algebraically equal.

3704 n1 -gt  n2
3705     True if the integer n1 is algebraically greater than the integer n2.

3706 n1 -ge  n2
3707     True if the integer n1 is algebraically greater than or equal to
3708     the integer n2.

3709 n1 -lt  n2
3710     True if the integer n1 is algebraically less than the integer n2.

3711 n1 -le  n2
```

```
3712     True if the integer n1 is algebraically less than or equal to the
3713     integer n2.

3714 expression1 -a  expression2
3715     [XSI] True if both expression1 and expression2 are true. The -a
3716     binary primary is left associative. It has a higher precedence than
3717     -o. [/XSI]

3718 expression1 -o  expression2
3719     [XSI] True if either expression1 or expression2 is
3720     true. The -o binary primary is left associative. [/XSI]

3721 With the exception of the -h file and -L file primaries, if a file
3722 argument is a symbolic link, test shall evaluate the expression by
3723 resolving the symbolic link and using the file referenced by the link.

3724 These primaries can be combined with the following operators:

3725 !  expression
3726     True if expression is false.
3727 (  expression  )
3728     [XSI] True if expression is true. The parentheses can be used to
3729     alter the normal precedence and associativity.  [/XSI]"

3730 To:
3731 "-b pathname
3732   True if pathname resolves to a file that exists and is a block special
3733   file.  False if pathname cannot be resolved, or if pathname resolves
3734   to a file that exists but is not a block special file.

3735 -c pathname
3736   True if pathname resolves to a file that exists and is a character
3737   special file.  False if pathname cannot be resolved, or if pathname
3738   resolves to a file that exists but is not a character special file.

3739 -d pathname
3740   True if pathname resolves to a file that exists and is a directory.
3741   False if pathname cannot be resolved, or if pathname resolves to a
3742   file that exists but is not a directory.

3743 -e pathname
3744   True if pathname resolves to a file that exists.  False if pathname
3745   cannot be resolved.

3746 -f pathname
3747   True if pathname resolves to a file that exists and is a regular file.
3748   False if pathname cannot be resolved, or if pathname resolves to a
3749   file that exists but is not a regular file.

3750 -g pathname
3751   True if pathname resolves to a file that exists and has its set-group-ID
3752   flag set.  False if pathname cannot be resolved, or if pathname resolves
3753   to a file that exists but does not have its set-group-ID flag set.

3754 -h pathname
3755   True if pathname resolves to a file that exists and is a symbolic link.
3756   False if pathname cannot be resolved, or if pathname resolves to a file
3757   that exists but is not a symbolic link.  If the final component of
3758   pathname is a symlink that symlink is not followed.

3759 -L pathname
3760   True if pathname resolves to a file that exists and is a symbolic link.
3761   False if pathname cannot be resolved, or if pathname resolves to a
3762   file that exists but is not a symbolic link.  If the final component
3763   of pathname is a symlink that symlink is not followed.

3764 -n string
3765   True if the length of string is non-zero, otherwise false.

3766 -p pathname
3767   True if pathname resolves to a file that exists and is a FIFO.
```

```
3768   False if pathname cannot be resolved, or if pathname resolves to a
3769   file that exists but is not a FIFO.

3770 -r pathname
3771   True if pathname resolves to a file that exists and for which permission
3772   to read from the file will be granted, as defined in Section 1.7.1.4
3773   (on page 4).  False if pathname cannot be resolved, or if pathname
3774   resolves to a file for which permission to read from the file will
3775   not be granted.

3776 -S pathname
3777   True if pathname resolves to a file that exists and is a socket.
3778   False if pathname cannot be resolved, or if pathname resolves to a
3779   file that exists but is not a socket.

3780 -s pathname
3781   True if pathname resolves to a file that exists and has a size greater
3782   than zero.  False if pathname cannot be resolved, or if pathname
3783   resolves to a file that exists but does not have a size greater
3784   than zero.

3785 -t file_descriptor
3786   True if file descriptor number file_descriptor is open and is
3787   associated with a terminal.  False if file_descriptor is not a valid
3788   file descriptor number, or if file descriptor number file_descriptor
3789   is not open, or if it is open but is not associated with a terminal.

3790 -u pathname
3791   True if pathname resolves to a file that exists and has its set-user-ID
3792   flag set.  False if pathname cannot be resolved, or if pathname resolves
3793   to a file that exists but does not have its set-user-ID flag set.

3794 -w pathname
3795   True if pathname resolves to a file that exists and for which permission
3796   to write to the file will be granted, as defined in Section 1.7.1.4 (on
3797   page 4).  False if pathname cannot be resolved, or if pathname resolves
3798   to a file for which permission to write to the file will not be granted.

3799 -x pathname
3800   True if pathname resolves to a file that exists and for which
3801   permission to execute the file (or search it, if it is a directory)
3802   will be granted, as defined in Section 1.7.1.4 (on page 4).  False if
3803   pathname cannot be resolved, or if pathname resolves to a file for
3804   which permission to execute (or search) the file will not be granted.

3805 -z  string
3806     True if the length of string string is zero, otherwise false.

3807 string
3808     True if the string string is not the null string, otherwise false.

3809 s1 =  s2
3810     True if the strings s1 and s2 are identical, otherwise false.

3811 s1 !=  s2
3812     True if the strings s1 and s2 are not identical, otherwise false.

3813 n1 -eq  n2
3814     True if the integers n1 and n2 are algebraically equal, otherwise
3815     false.

3816 n1 -ne  n2
3817     True if the integers n1 and n2 are not algebraically equal,
3818     otherwise false.

3819 n1 -gt  n2
3820     True if the integer n1 is algebraically greater than the integer n2,
3821     otherwise false.

3822 n1 -ge  n2
3823     True if the integer n1 is algebraically greater than or equal to
```

the integer n2, otherwise false.

n1 -lt n2
    True if the integer n1 is algebraically less than the integer n2,
    otherwise false.

n1 -le n2
    True if the integer n1 is algebraically less than or equal to the
    integer n2, otherwise false.

expression1 -a expression2
    [XSI] True if both expression1 and expression2 are true, otherwise
    false. The -a binary primary is left associative. It has a higher
    precedence than -o. [/XSI]

expression1 -o expression2
    [XSI] True if either expression1 or expression2 is true, otherwise
    false. The -o binary primary is left associative. [/XSI]

With the exception of the -h file and -L file primaries, if a file
argument is a symbolic link, test shall evaluate the expression by
following the symbolic link and using the file referenced by the link.

These primaries can be combined with the following operators:

! expression
    True if expression is false. False if expression is true.
(expression)
    [XSI] True if expression is true. False if expression is false. The
    parentheses can be used to alter the normal precedence and
    associativity. [/XSI]

Rationale:
The previous descriptions of the file existence primaries for the test utility
have been clarified.

Change Number: XCU/TC2/D6/31 [XCU ERN 33]

On Page: 927 Line: 35871-35872,36041 Section: tr (2003 Ed.)
On Page: 924 Line: 35616-35617,35786 Section: tr (2001 Ed.)

In the EXTENDED DESCRIPTION section
Delete the sentence:
"If the size of a byte on the system is greater than nine bits,
the valid escape sequence to represent a byte is
implementation-defined."

Add to the RATIONALE section:
"Prior versions of this standard allowed for implementations with
bytes other than eight bits, but this has been modified in this
version."

Change Number: XCU/TC2/D6/32 [XCU ERN 12]

On Page: 930 Line: 35992 Section: tr (2003 Ed.)
On Page: 927 Line: 35737 Section: tr (2001 Ed.)

In the EXAMPLES section
Change From:
"tr "[=e=]" e <file1 >file2"

To:
"tr "[=e=]" "[e*]" <file1 >file2"

Rationale:
The previous example depended on unspecified behavior.

Change Number: XCU/TC2/D6/33 [XCU ERN 20]

On Page: 931 Line: 36041 Section: tr (2003 Ed.)
On Page: 928 Line: 35786 Section: tr (2001 Ed.)

In the RATIONALE section,
Change From:
"IEEE Std 1003.1-2001 specifies that octal sequences always refer to
single byte binary values."

To:
"IEEE Std 1003.1-2001 specifies that octal sequences always refer to
single byte binary values when used to specify an endpoint of a range
of collating elements."

Rationale:
This clarification is for consistency with the normative text.

Change Number: XCU/TC2/D6/34 [XCU ERN 35]

On Page: 945 Line: 36539-36540 Section: umask (2003 Ed.)
On Page: 663 Line: 25539-25542 Section: umask (2001 Ed.)

In the RATIONALE section
Change From:
"The default output style is implementation-defined to permit implementors
to provide migration to the new symbolic style at the time most
appropriate to their users."

To:
"The default output style is unspecified to permit implementors
to provide migration to the new symbolic style at the time most
appropriate to their users."

Rationale: The change is made for consistency with the normative text.

Change Number: XCU/TC2/D6/35 [XCU ERN 16]

On Page: 969 Line: 37351 Section: uudecode (2003 Ed.)
On Page: 966 Line: 37089 Section: uudecode (2003 Ed.)

In the DESCRIPTION section
Add at the end of the first paragraph:

"If either of the op characters '+' and '-' (see chmod) are specified
in symbolic mode, the initial mode on which those operations are based
is unspecified."

Note to editors: "op" is in italics

5. Changes to Rationale

This section contains the set of changes to the text of the Rationale.

Change Number: XRAT/TC2/D6/1 [XBD ERN 20]

On Page: xxxiii Line: "ISO/IEC 8859" Section: Referenced Documents (2003 Ed.)
On Page: xxix Line: "ISO/IEC 8859" Section: Referenced Documents (2001 Ed.)

Add after line starting "Part 10":

"Part 11: Latin/Thai Alphabet"

Add after line starting "Part 15":

"Part 16: Latin Alphabet No. 10"

Change Number: XRAT/TC2/D6/2 [XBD ERN 35]

On Page: 7 Line: 191 Section: A.1.4 (2003 Ed.)
On Page: 7 Line: 190 Section: A.1.4 (2001 Ed.)

Add to the end of the paragaph:
"In some places the text refers to facilities supplied by the implementation that are outside of the standard as implementation-supplied or implementation-provided. This is not intended to imply a requirement for documentation. If it were the term "implementation-defined" would have been used."

Change Number: XRAT/TC2/D6/3 [XBD ERN 17]

On Page: 42 Line: 1652 Section: A.6.3 (2003 Ed.)
On Page: 42 Line: 1648 Section: A.6.3 (2001 Ed.)

Change From:
"There is no additional rationale provided for this section."

To:
"The standard does not specify how wide characters are encoded or provide a method for defining wide characters in a charmap. It specifies ways of translating between wide characters and multibyte characters . The standard does not prevent an extension from providing a method to define wide characters."

Change Number: XRAT/TC2/D6/4 [XBD ERN 25]

On Page: 49 Line: 1962-1966 Section: A.7.3.3 (2003 Ed.)
On Page: 49 Line: 1958-1962 Section: A.7.3.3 (2001 Ed.)

Change From:
"The locale definition is an extension of the ISO C standard localeconv() specification. In particular, rules on how currency_symbol is treated are extended to also cover int_curr_symbol, and p_set_by_space and n_sep_by_space have been augmented with the value 2, which places a <space> between the sign and the symbol (if they are adjacent; otherwise, it should be treated as a 0). The following table shows the result of various combinations:"

To:
"The locale definition is an extension of the ISO C standard localeconv() specification. In particular, rules on how currency_symbol is treated are extended to also cover int_curr_symbol, and p_set_by_space and n_sep_by_space have been augmented with the value 2, which places a <space> between the sign and the symbol. This has been updated to match the C99 requirements and is an incompatible change from UNIX 98 and POSIX.2-1992 and POSIX.1-1996 requirements. The following table shows

the result of various combinations:"

Change Number: XRAT/TC2/D6/5 [XSH ERN 92]

On Page: 55 Line: 2254-2257,2258-2259 Section: 8.2 (2003 Ed.)
On Page: 56 Line: 2244-2247,2248-2249 Section: 8.2 (2001 Ed.)

In Section 8.2 Internationalization Variables
Change From:
"The text about locale implies that any utilities written in standard
C and conforming to IEEE Std 1003.1-2001 must issue the following call:

setlocale(LC_ALL, "")

If this were omitted, the ISO C standard specifies that the C locale
would be used."

To:
"Utilities conforming to the Shell and Utilities volume of IEEE Std
1003.1-2001 and written in standard C can access the locale variables
by issuing the following call:

setlocale(LC_ALL, "")

If this were omitted, the ISO C standard specifies that the C locale
would be used."

Change From:
"If any of the environment variables are invalid, it makes sense to
default to an implementation-defined, consistent locale environment."

To:
"The DESCRIPTION of setlocale() requires that when setting all
categories of a locale,that if the value of any of
the environment-variable searches yields a locale that is not supported
(and nonnull), the setlocale() function returns a null pointer and the
locale of the process is unchanged.

For the standard utilities, if any of the environment variables
are invalid, it makes sense to default to an implementation-defined,
consistent locale environment."

Change Number: XRAT/TC2/D6/6 [XRAT ERN 1]

On Page: 72 Line: 2960-2961 Section: A.12.2 (2003 Ed.)
On Page: 73 Line: 2948-2949 Section: A.12.2 (2001 Ed.)

Change From:
"Guidelines 1 and 2 are offered as guidance for locales using Latin
alphabets. No recommendations are made by IEEE Std 1003.1-2001 concerning
utility naming in other locales."

To:
"Guidelines 1 and 2 encourage utility writers to use only characters from
the portable character set because use of locale specific characters
may make the utility inaccessible from other locales. Use of uppercase
letters is discouraged due to problems associated with porting utilities
to systems that don't distinguish between uppercase and lowercase
characters in file names. Use of non-alphanumeric characters is
discouraged due to the number of utilities that treat non-alphanumeric
characters in "special" ways depending on context (such as the shell
using whitespace characters to delimit arguments, various quote characters
for quoting, <dollar-sign> to introduce variable expansion,...)"

Change Number: XRAT/TC2/D6/7 [XSH ERN 89]

On Page: 177 Line: 7479 Section: B.2 Threads (2003 Ed.)
On Page: 178 Line: 7442 Section: B.2 Threads (2001 Ed.)

Insert new section before B.2.10:
"B.2.9.8 Use of Application-Managed Thread Stacks

IEEE Std 1003.1-2001/Cor 2-200x, item XSH/TC2/Dx/x is applied, adding this new section. It was added to make it clear that the current standard does not allow an application to determine when a stack can be reclaimed. This may be addressed in a future revision."

Notes to Editors: The reference should be updated to reflect the final draft.

Change Number: XRAT/TC2/D6/8 [XBD ERN 11]

On Page: 224 Line: 9264-9265 Section: C.1.9 (2003 Ed.)
On Page: 224 Line: 9226-9227 Section: C.1.9 (2001 Ed.)

Delete the paragraph:
" {POSIX2_SYMLINKS} was developed even though there is no comparable configuration value for the system interfaces."

Rationale: The pathconf() symbolic constant _PC_2_SYMLINKS has been introduced as a result of this defect report.

Change Number: XRAT/TC2/D6/9 [XRAT ERN 2]

On Page: 240,241 Line: 9903-9911,9921,9925,9958 Section: C.2.6.4 (2003 Ed.)
On Page: 239,240 Line: 9852-9860,9870,9874,9907 Section: C.2.6.4 (2001 Ed.)

Change From:
"The "(())" form of KornShell arithmetic in early proposals was omitted. The standard developers concluded that there was a strong desire for some kind of arithmetic evaluator to replace expr, and that relating it to '$' makes it work well with the standard shell language, and it provides access to arithmetic evaluation in places where accessing a utility would be inconvenient.

The syntax and semantics for arithmetic were changed for the ISO/IEC 9945-2:1993 standard. The language is essentially a pure arithmetic evaluator of constants and operators (excluding assignment) and represents a simple subset of the previous arithmetic language (which was derived from the KornShell "(())" construct). "

To:
"The standard developers agreed that there was a strong desire for some kind of arithmetic evaluator to provide functionality similar to expr, that relating it to '$' makes it work well with the standard shell language and provides access to arithmetic evaluation in places where accessing a utility would be inconvenient.

The syntax and semantics for arithmetic were revised for the ISO/IEC 9945-2:1993 standard. The language represents a simple subset of the previous arithmetic language (which was derived from the KornShell "(())" construct). "

On line 9921 (2003 Ed.); line 9870 (2001 Ed.):
Add before the paragraph commencing "In early proposals"

"The standard requires assignment operators to be supported (as listed in Section 2.7.1), and since arithmetic expansions are not specified to be evaluated in a subshell environment, changes to variables there have to be in effect after the arithmetic expansion, just as in the parameter expansion ${x=value}.

Note, however, that $((x=5)) need not be equivalent to $(($x=5)).

4054 If the value of the environment variable x is the string "y=", the
4055 expansion of $((x=5)) would set x to 5 and output 5, but $(($x=5))
4056 output 0 if the value of the environment variable y is not 5 and would
4057 output 1 if the environment variable y is 5. Similarly, if the value of
4058 the environment variable is 4, the expansion of $((x=5)) would still
4059 set x to 5 and output 5, but $(($x=5)) (which would be equivalent to
4060 $((4=5))) would yield a syntax error. "

4061 In the paragraph on line 9925 (2003 Ed.); line 9874, (2001 Ed.)
4062 Change From:
4063 "The portion of the ISO C standard arithmetic operations selected
4064 corresponds to the operations historically supported in the KornShell."

4065 To:
4066 "The portion of the ISO C standard arithmetic operations selected
4067 corresponds to the operations historically supported in the KornShell.
4068 In addition to the exceptions listed in section 2.6.4, the use of the
4069 following are explicitly outside the scope of the rules defined in
4070 section 1.7.2.1:

4071 * The prefix operator & and the [], -> and . operators.

4072 * Casts."

4073 On line 9958 (2003 Ed.); line 9907 (2001 Ed.)
4074 Add before the paragraph commencing: "Although the ISO/IEC 9899:1999
4075 Standard now..." :

4076 "The standard is intentionally silent about how a variable's numeric
4077 value in an expression is determined from its normal "sequence of bytes"
4078 value. It could be done as a text substitution, as a conversion like
4079 that performed by strtol(), or even recursive evaluation. Therefore,
4080 the only cases for which the standard is clear are those for which
4081 both conversions produce the same result. The cases where they give
4082 the same result are those where the sequence of bytes form a valid
4083 integer constant. Therefore, if a variable does not contain a valid
4084 integer constant, the behavior is unspecified.

4085 For the commands:
4086   x=010; echo $((x += 1))
4087 the output must be 9.

4088 For the commands:
4089   x=' 1'; echo $((x += 1))
4090 the results are unspecified.

4091 For the commands:
4092   x=1+1; echo $((x += 1))
4093 the results are unspecified."