

Recognized as an

American National Standard (ANSI)

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

Approved 31 May 2003

American National Standards Institute



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)

All rights reserved. No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission from both the IEEE and The Open Group.

Portions of this standard are derived with permission from copyrighted material owned by Hewlett-Packard Company, International Business Machines Corporation, Novell Inc., The Open Software Foundation, and Sun Microsystems, Inc.

Permissions

Authorization to photocopy portions of this standard for internal or personal use is granted provided that the appropriate fee is paid to the Copyright Clearance Center or the equivalent body outside of the U.S. Permission to make multiple copies for educational purposes in the U.S. requires agreement and a license fee to be paid to the Copyright Clearance Center.

Beyond these provisions, permission to reproduce all or any part of this standard must be with the consent of both copyright holders and may be subject to a license fee. Both copyright holders will need to be satisfied that the other has granted permission. Requests to the copyright holders should be sent by email to austin-group-permissions@opengroup.org.

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

Austin Group Working Group Members

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

H. Stephen Berger

Mark D. Bowman

Joseph A. Bruder

Bob Davis

Roberto de Boisson

Julian Forster*

Arnold M. Greenspan

Mark S. Halpin

Raymond Hapeman

Richard J. Holleman

Richard H. Hulett

Lowell G. Johnson

Hermann Koch

Joseph L. Koepfinger*

Thomas J. McGean

Steve M. Mills

Daleep C. Mohla

Paul Nikolich

T. W. Olsen

Ronald C. Petersen

Gary S. Robinson

Frank Stone

Malcolm V. Thaden

Doug Topping

Joe D. Watson

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 Execution 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.

Technical Corrigendum 2

0026 **Change Number:** XBD/TC2/D6/3 [XBD ERN 4]

0027 On Page: 48 Line: 1631-1635 Section: 3.93 Child Process (2003 Ed.)

0028 On Page: 46 Line: 1619-1623 Section: 3.93 Child Process (2001 Ed.)

0029 Change From:

0030 "3.93 Child Process

0031 A new process created (by `fork()`, `posix_spawn()`, or `posix_spawnp()`)

0032 by a given process. A child process remains the child of the creating

0033 process as long as both processes continue to exist.

0034 Note: The `fork()`, `posix_spawn()`, and `posix_spawnp()` functions

0035 are defined in detail in the System Interfaces volume of IEEE Std

0036 1003.1-2001."

0037 To:

0038 "3.93 Child Process

0039 A new process created (by `fork()`, `posix_spawn()`, `posix_spawnp()`, or

0040 `vfork()`) by a given process. A child process remains the child of the

0041 creating process as long as both processes continue to exist.

0042 Note: The `fork()`, `posix_spawn()`, `posix_spawnp()` and `vfork()` functions

0043 are defined in detail in the System Interfaces volume of IEEE Std

0044 1003.1-2001."

0045 **Change Number:** XBD/TC2/D6/4 [XBD ERN 30]

0046 On Page: 61 Line: 2005-2006 Section: 3.187 Group Database (2003 Ed.)

0047 On Page: 59 Line: 1993-1994 Section: 3.187 Group Database (2001 Ed.)

0048 Change From:

0049 "A system database of implementation-defined format that contains at

0050 least the following information for each group ID:"

0051 To:

0052 "A system database that contains at least the following information for

0053 each group ID:"

0054 **Change Number:** XBD/TC2/D6/5 [XBD ERN 7]

0055 On Page: 76 Line: 2433,2440 Section: 3.295 Process Lifetime (2003 Ed.)

0056 On Page: 74 Line: 2421,2428 Section: 3.295 Process Lifetime(2001 Ed.)

0057 In the Section 3.295 Process Lifetime

0058 Change From:

0059 "After a process is created with a `fork()` function, it is considered

0060 active."

0061 To:

0062 "After a process is created by `fork()`, `posix_spawn()`, `posix_spawnp()`

0063 or `vfork()`, it is considered active."

0064 Change From:

0065 "Note: The `fork()`, `wait()`, `waitid()`, and `waitpid()` functions

0066 are defined in detail in the System Interfaces volume of IEEE Std

0067 1003.1-2001."

0068 To:

0069 "Note: The `fork()`, `posix_spawn()`, `posix_spawnp()`, `vfork()`, `wait()`,

0070 `waitid()`, and `waitpid()` functions are defined in detail in the System

0071 Interfaces volume of IEEE Std 1003.1-2001."

0072 **Change Number:** XBD/TC2/D6/6 [XBD ERN 5]

0073 On Page: 76 Line: 2448-2454 Section: 3.297 Process Termination (2003 Ed.)

0074 On Page: 74 Line: 2436-2442 Section: 3.297 Process Termination (2001 Ed.)

0075 Change From:

0076 "There are two kinds of process termination:

0077 1. Normal termination occurs by a return from main() or when requested
0078 with the exit() or _exit() functions.

0079 2. Abnormal termination occurs when requested by the abort() function
0080 or when some signals are received.

0081 Note: The _exit(), abort(), and exit() functions are defined in
0082 detail in the System Interfaces volume of IEEE Std 1003.1-2001."

0083 To:

0084 "There are two kinds of process termination:

0085 1. Normal termination occurs by a return from main(), when requested
0086 with the exit(), _exit(), or _Exit() functions; or when the last thread
0087 in the process terminates by returning from its start function, by
0088 calling the pthread_exit() function, or through cancellation.

0089 2. Abnormal termination occurs when requested by the abort() function
0090 or when some signals are received.

0091 Note: The _exit(), _Exit(), abort(), and exit() functions are defined in
0092 detail in the System Interfaces volume of IEEE Std 1003.1-2001."

0093 Rationale:

0094 The definition does not mention the 'passive exit' on termination of
0095 the last thread or the _Exit() function.

0096 **Change Number:** XBD/TC2/D6/7 [XBD ERN 31,32]

0097 On Page: 88 Line: 2778-2779 Section: 3.382 System Console (2003 Ed.)

0098 On Page: 86 Line: 2766-2767 Section: 3.382 System Console (2001 Ed.)

0099 Change From:

0100 "An implementation-defined device that receives messages sent by the
0101 syslog() function, and the fmtmsg() function when the MM_CONSOLE flat
0102 is set."

0103 To:

0104 "A device that receives messages sent by the syslog() function, and the
0105 fmtmsg() function when the MM_CONSOLE flag is set."

0106 **Change Number:** XBD/TC2/D6/8 [XBD ERN 33]

0107 On Page: 88 Line: 2802-2803 Section: 3.385 System Process (2003 Ed.)

0108 On Page: 86 Line: 2790-2791 Section: 3.385 System Process (2001 Ed.)

0109 Change From:

0110 "An implementation-defined object, other than a process executing an
0111 application, that has a process ID."

0112 To:

0113 "An object other than a process executing an application, that is
0114 provided by the system and has a process ID."

Technical Corrigendum 2

0115 **Change Number:** XBD/TC2/D6/9 [TC2d5 ERN 4]

0116 On Page: 88 Line: 2783-2796 Section: 3.383 System Databases (2003 Ed.)

0117 On Page: 86 Line: 2771-2784 Section: 3.383 System Databases (2001 Ed.)

0118 Change From:

0119 "An implementation provides two system databases.

0120 The "group database" contains the following information for each group:

0121 1. Group name

0122 2. Numerical group ID

0123 3. List of all users allowed in the group

0124 The "user database" contains the following information for each user:

0125 1. User name

0126 2. Numerical user ID

0127 3. Numerical group ID

0128 4. Initial working directory

0129 5. Initial user program

0130 If the initial user program field is null, the system default is used. If

0131 the initial working directory field is null, the interpretation of that

0132 field is implementation-defined. These databases may contain other fields

0133 that are unspecified by IEEE Std 1003.1-2001."

0134 To:

0135 "An implementation provides two system databases, the "group database"

0136 (see also Section 3.187) and the "user database" (see also Section

0137 3.424)."

0138 **Change Number:** XBD/TC2/D6/10 [TC2d5 ERN 5]

0139 On Page: 89 Line: 2805-2806 Section: 3.386 System Reboot (2003 Ed.)

0140 On Page: 87 Line: 2793-2794 Section: 3.386 System Reboot (2001 Ed.)

0141 Change From:

0142 "An implementation-defined sequence of events that may result in the loss

0143 of transitory data; that is, data that is not saved in permanent storage."

0144 To:

0145 "An unspecified sequence of events that may result in the loss

0146 of transitory data; that is, data that is not saved in permanent storage."

0147 **Change Number:** XBD/TC2/D6/11 [XBD ERN 36]

0148 On Page: 94 Line: 2939-2940 Section: 3.424 User Database (2003 Ed.)

0149 On Page: 92 Line: 2927-2928 Section: 3.424 User Database (2001 Ed.)

0150 Change From:

0151 "A system database of implementation-defined format that contains at

0152 least the following information for each user ID:"

0153 To:

0154 "A system database that contains at least the following information for

0155 each user ID:"

0156 **Change Number:** XBD/TC2/D6/12 [XBD ERN 28]

0157 On Page: 104 Line: 3222-3223 Section: 4.14 (2003 Ed.)

0158 On Page: 102 Line: 3207-3208 Section: 4.14 (2001 Ed.)

0159 Change From:

0160 "How any changes to the value of seconds since the Epoch
0161 are made to align to a desired relationship with the current
0162 actual time are made is implementation-defined."

0163 To:

0164 "How any changes to the value of seconds since the Epoch
0165 are made to align to a desired relationship with the current
0166 actual time is implementation-defined."

0167 Rationale:

0168 This is an editorial change.

0169 **Change Number:** XBD/TC2/D6/13 [XBD ERN 17]

0170 On Page: 119 Line: 3758 Section: 6.3 (2003 Ed.)

0171 On Page: 117 Line: 3742 Section: 6.3 (2001 Ed.)

0172 Add to the end of the paragraph:

0173 "This standard provides no means of defining a wide character codeset."

0174 **Change Number:** XBD/TC2/D6/14 [XBD ERN 17]

0175 On Page: 121 Line: 3860-3862,3867-3868,3869 Section: 6.4 (2003 Ed.)

0176 On Page: 119 Line: 3844-3846,3851-3852,3853 Section: 6.4 (2001 Ed.)

0177 Change From:

0178 "Bytes shall be treated as unsigned octets, and carry shall be propagated
0179 between the bytes as necessary to represent the range. For example,
0180 the line:"

0181 To:

0182 "Bytes shall be treated as unsigned octets, and carry shall be propagated
0183 between the bytes as necessary to represent the range. However, because
0184 this causes a null byte in the second or subsequent bytes of a character,
0185 such a declaration should not be specified. For example, the line:"

0186 Change From:

0187 <j0103> \d130\d0
0188 <j0104> \d130\d1

0189 To:

0190 <j0103> \d130\d00
0191 <j0104> \d130\d01

0192 Change From:

0193 "The comment is optional."

0194 To:

0195 "The expanded declaration of the symbol <j0103> in the above
0196 example is an invalid specification, because it contains a null
0197 byte in the second byte of a character.

0198 The comment is optional."

0199 Add to the end of the paragraph:

0200 "This standard provides no means of defining a wide character codeset."

Technical Corrigendum 2

0201 **Change Number:** XBD/TC2/D6/15 [XBD ERN 24]

0202 On Page: 121-122 Line: 3873-3885 Section: 6.4 (2003 Ed.)

0203 On Page: 119-120 Line: 3857-3859 Section: 6.4 (2001 Ed.)

0204 Change From:

0205 "WIDTH An unsigned positive integer value defining the column width
0206 (see Section 3.103 (on page 49)) for the printable characters in the
0207 coded character set specified in Table 6-1 (on page 115) and Table 6-2
0208 (on page 120)."

0209 To:

0210 "WIDTH A non-negative integer value defining the column width (see
0211 Section 3.103 (on page 49)) for the printable characters in the coded
0212 character set specified in Table 6-1 (on page 115) and Table 6-2 (on
0213 page 120)."

0214 Change From:

0215 "WIDTH_DEFAULT An unsigned positive integer value defining the default
0216 column width for any printable character not listed by one of the WIDTH
0217 keywords."

0218 To:

0219 "WIDTH_DEFAULT A non-negative integer value defining the default
0220 column width for any printable character not listed by one of the WIDTH
0221 keywords."

0222 Rationale: This change allows the value zero for the width value of WIDTH
0223 and WIDTH_DEFAULT. This is required to cover some existing locales

0224 **Change Number:** XBD/TC2/D6/16 [XBD ERN 25]

0225 On Page: 143 Line: 4836-4839 Section: 7.3.3 (2003 Ed.)

0226 On Page: 141 Line: 4820-4823 Section: 7.3.3 (2001 Ed.)

0227 On page 143 lines 4836-4839 (2003 Ed.), page 141 lines 4820-4823 (2001 Ed.)

0228 Change From:

0229 "p_sep_by_space An integer set to 0 if no space separates the
0230 currency_symbol from the value for a monetary quantity with a non-negative
0231 value, set to 1 if a space separates the symbol from the value, and
0232 set to 2 if a space separates the symbol and the sign string, if adjacent."

0233 To:

0234 "p_sep_by_space Set to a value indicating the separation of the
0235 currency_symbol, the sign string, and the |
0236 value for a non-negative formatted monetary quantity.

0237 The values of p_sep_by_space, n_sep_by_space, int_p_sep_by_space, and
0238 int_n_sep_by_space are interpreted according to the following:

0239 0 No space separates the currency symbol and value.

0240 1 If the currency symbol and sign string are adjacent, a
0241 space separates them from the value; otherwise, a space
0242 separates the currency symbol from the value.

0243 2 If the currency symbol and sign string are adjacent, a
0244 space separates them; otherwise, a space separates the
0245 sign string from the value."

0246 On page 143 lines 4843-4846 (2003 Ed.), page 141 lines 4827-2830 (2001 Ed.)

0247 Change From:

0248 "n_sep_by_space An integer set to 0 if no space separates the
0249 currency_symbol from the value for a monetary quantity with a negative
0250 value, set to 1 if a space separates the symbol from the value, and set
0251 to 2 if a space separates the symbol and the sign string, if adjacent."

0252 To:

0253 "n_sep_by_space Set to a value indicating the separation of
0254 the currency_symbol, the sign string, and the value for a negative

0255 formatted monetary quantity."

0256 On page 144 lines 4864-4871 (2003 Ed.), page 142 lines 4848-4855 (2001 Ed.)

0257 Change From:

0258 "int_p_sep_by_space An integer set to 0 if no space separates the

0259 int_curr_symbol from the value for a monetary quantity with a non-negative

0260 value, set to 1 if a space separates the symbol from the value, and set

0261 to 2 if a space separates the symbol and the sign string, if adjacent.

0262 int_n_sep_by_space An integer set to 0 if no space separates the

0263 int_curr_symbol from the value for a monetary quantity with a negative

0264 value, set to 1 if a space separates the symbol from the value, and set

0265 to 2 if a space separates the symbol and the sign string, if adjacent."

0266 To:

0267 "int_p_sep_by_space Set to a value indicating the separation

0268 of the int_curr_symbol, the sign string, and the value for a

0269 non-negative internationally formatted monetary quantity."

0270 int_n_sep_by_space Set to a value indicating the separation of

0271 the int_curr_symbol, the sign string, and the value for a negative

0272 internationally formatted monetary quantity."

0273 Rationale:

0274 The descriptions of p_sep_by_space, n_sep_by_space, int_p_sep_by_space,

0275 and int_n_sep_by_space need to be updated to match the description of

0276 these keywords in C99 and System Interfaces Volume, localeconv().

0277 **Change Number:** XBD/TC2/D6/17 [XBD ERN 37]

0278 On Page: 185 Line: 6540-6541 Section: 10.2 (2003 Ed.)

0279 On Page: 183 Line: 6508-6509 Section: 10.2 (2001 Ed.)

0280 Change From:

0281 "The implementation shall document which terminal types it supports and

0282 which of these features and utilities are not supported by each terminal."

0283 To:

0284 "The implementation shall document in the system documentation which

0285 terminal types it supports and which of these features and utilities

0286 are not supported by each terminal."

0287 **Change Number:** XBD/TC2/D6/18 [XSH ERN 146]

0288 On Page: 224 Line: 7897-7899 Section: fcntl.h (2003 Ed.)

0289 On Page: 222 Line: 7864-7865 Section: fcntl.h (2001 Ed.)

0290 In the DESCRIPTION section

0291 Change From:

0292 "[ADV] int posix_fadvise(int, off_t, size_t, int);

0293 int posix_fallocate(int, off_t, size_t);"

0294 To:

0295 "[ADV] int posix_fadvise(int, off_t, off_t, int);

0296 int posix_fallocate(int, off_t, off_t);"

0297 Rationale: The previous prototype was not large-file aware, and the standard

0298 developers felt it acceptable to make this change before

0299 implementations of the functions become widespread."

Technical Corrigendum 2

0300 **Change Number:** XBD/TC2/D6/19 [XBD ERN 12]

0301 On Page: 259-260 Line: 9132-9169 Section: limits.h (2003 Ed.)

0302 On Page: 257-258 Line: 9093-9131 Section: limits.h (2001 Ed.)

0303 On line 9132-9134 (2003 Ed.); line 9093-9095 (2001 Ed.)

0304 Change From:

0305 "{INT_MAX}

0306 Maximum value of an int.

0307 Minimum Acceptable Value: 2 147 483 647"

0308 To:

0309 "{INT_MAX}

0310 Maximum value of an int.

0311 [CX]Minimum Acceptable Value: 2 147 483 647[/CX]"

0312 On line 9156-9158 (2003 Ed.); line 9117-9119 (2001 Ed.)

0313 Change From:

0314 "{UINT_MAX}

0315 Maximum value of type unsigned.

0316 Minimum Acceptable Value: 4 294 967 295"

0317 To:

0318 "{UINT_MAX}

0319 Maximum value of type unsigned.

0320 [CX]Minimum Acceptable Value: 4 294 967 295[/CX]"

0321 On line 9165-9167 (2003 Ed.); line 9126-9128 (2001 Ed.)

0322 Change From:

0323 "[XSI] {WORD_BIT}

0324 Number of bits in a word or type int.

0325 Minimum Acceptable Value: 16[/XSI]"

0326 To:

0327 "[XSI] {WORD_BIT}

0328 Number of bits in a type int.

0329 Minimum Acceptable Value: 32[/XSI]"

0330 On Line 9168-9170 (2003 Ed.); line 9129-9131 (2001 Ed.)

0331 Change From:

0332 "{INT_MIN}

0333 Minimum value of type int.

0334 Maximum Acceptable Value: -2 147 483 647"

0335 To:

0336 "{INT_MIN}

0337 Minimum value of type int.

0338 [CX]Maximum Acceptable Value: -2 147 483 647[/CX]"

0339 Rationale: There was conflicting information about the size

0340 of an integer.

0341 **Change Number:** XBD/TC2/D6/20 [XBD ERN 26]

0342 On Page: 260 Line: 9191-9193 Section: limits.h (2003 Ed.)

0343 On Page: 258 Line: 9152-9154 Section: limits.h (2001 Ed.)

0344 In the DESCRIPTION section

0345 Under "Other Invariant Values"

0346 Remove the lines:

0347 "XSI {CHARCLASS_NAME_MAX}

0348 Maximum number of bytes in a character class name.

0349 Minimum Acceptable Value: 14"

0350 Rationale:

0351 CHARCLASS_NAME_MAX was defined under "Runtime Inceasable Values"

0352 and also "Other Invarient Values". This change corrects

0353 an integration issue with the base specifications.

0354 **Change Number:** XBD/TC2/D6/21 [XBD ERN 21]

0355 On Page: 266 Line: 9440-9447 Section: math.h (2003 Ed.)
 0356 On Page: 264 Line: 9399-9406 Section: math.h (2001 Ed.)

0357 Change From:

0358 "The following optional macros indicate whether the fma() family of
 0359 functions are fast compared with direct code:

0360 FP_FAST_FMA
 0361 FP_FAST_FMAF
 0362 FP_FAST_FMAL

0363 The FP_FAST_FMA macro shall be defined to indicate that the fma() function
 0364 generally executes about as fast as, or faster than, a multiply and an
 0365 add of double operands. The other macros have the equivalent meaning
 0366 for the float and long double versions."

0367 To:

0368 "The following optional macros indicate whether the fma()
 0369 family of functions are fast compared with direct code:

0370 FP_FAST_FMA
 0371 FP_FAST_FMAF
 0372 FP_FAST_FMAL

0373 If defined, the FP_FAST_FMA macro shall indicate that the fma() function
 0374 generally executes about as fast as, or faster than, a multiply and an
 0375 add of double operands. If undefined it is unspecified what the speed
 0376 of execution is. The other macros have the equivalent meaning for the
 0377 float and long double versions."

0378 **Change Number:** XBD/TC2/D6/22 [XBD ERN 15]

0379 On Page: 281 Line: 10000 Section: netinet/in.h (2003 Ed.)
 0380 On Page: 279 Line: 9953 Section: netinet/in.h (2001 Ed.)

0381 Change From:

0382 "The <netinet/in.h> header shall define the sockaddr_in structure
 0383 that includes at least the following members (all in network
 0384 byte order):"

0385 To:

0386 "The <netinet/in.h> header shall define the sockaddr_in structure
 0387 that includes at least the following members listed below:"

0388 Add after line 10004 (2003 Ed.); line 9957 (2001 Ed.):
 0389 "The sin_port and sin_addr members shall be in network byte order."

0390 On line 10011 (2003 Ed.); line 9964 (2001 Ed.).

0391 Change From:

0392 "The <netinet/in.h> header shall define the sockaddr_in6 structure that
 0393 includes at least the following members (all in network byte order):"

0394 To:

0395 "The <netinet/in.h> header shall define the sockaddr_in6 structure that
 0396 includes at least the following members:"

0397 Add after line 10017 (2003 Ed.); line 9970 (2001 Ed.):
 0398 "The sin6_port and sin6_addr members shall be in network byte order."



Technical Corrigendum 2

0399 **Change Number:** XBD/TC2/D6/23 [XBD ERN 8]

0400 On Page: 297 Line: 10588,10600 Section: <sched.h> (2003 Ed.)

0401 On Page: 295 Line: 10537,10549 Section: <sched.h> (2001 Ed.)

0402 In the DESCRIPTION section

0403 Change From:

0404 "int sched_priority Process execution scheduling priority."

0405 To:

0406 "int sched_priority Process [THR]or thread[/THR] execution scheduling
0407 priority."

0408 Change From:

0409 "Each process is controlled by an associated scheduling policy and
0410 priority."

0411 To:

0412 "Each process [THR]or thread[/THR] is controlled by an associated
0413 scheduling policy and priority."

0414 **Change Number:** XBD/TC2/D6/24 [XBD ERN 18]

0415 On Page: 306 Line: 10907-10918 Section: signal.h (2003 Ed.)

0416 On Page: 304 Line: 10852-10863 Section: signal.h (2001 Ed.)

0417 Change From:

0418 "CX The <signal.h> header shall define the siginfo_t type as a structure
0419 that includes at least the following members:

0420 [CX]	int si_signo	Signal number.
0421 [XSI]	int si_errno	If non-zero, an errno value associated with
0422		this signal, as defined in <errno.h>.
0423 [CX]	int si_code	Signal code.
0424 [XSI]	pid_t si_pid	Sending process ID.
0425	uid_t si_uid	Real user ID of sending process.
0426	void *si_addr	Address of faulting instruction.
0427	int si_status	Exit value or signal.
0428	long si_band	Band event for SIGPOLL.
0429 [RTS]	union sigval si_value	Signal value. "

0430 To:

0431 "[CX] The <signal.h> header shall define the siginfo_t type as a structure
0432 that includes at least the following members:

0433 [CX]	int si_signo	Signal number.
0434	int si_code	Signal code.
0435 [XSI]	int si_errno	If non-zero, an errno value associated with
0436		this signal, as defined in <errno.h>.
0437	pid_t si_pid	Sending process ID.
0438	uid_t si_uid	Real user ID of sending process.
0439	void *si_addr	Address of faulting instruction.
0440	int si_status	Exit value or signal.
0441	long si_band	Band event for SIGPOLL.
0442 [RTS]	union sigval si_value	Signal value. "

0443 Rationale: This is an editorial change to the shading and no

0444 normative change is intended.

0445 **Change Number:** XBD/TC2/D6/25 [XBD ERN 29]

0446 On Page: 358 Line: 12708 Section: <sys/stat.h> (2003 Ed.)

0447 On Page: 356 Line: 12643 Section: <sys/stat.h> (2001 Ed.)

0448 Change From:

0449 "Times shall be given in seconds since the Epoch."

0450

0451 To:

0452 "The timespec structure may be defined as described in <time.h>.

0453 Times shall be given in seconds since the Epoch."

0454 **Change Number:** XBD/TC2/D6/26 [XBD ERN 9]

0455 On Page: 369 Line: 13064 Section: sys/types.h (2003 Ed.)

0456 On Page: 367 Line: 12988 Section: sys/types.h (2001 Ed.)

0457 Add "pthread_t" to the list of types in lines 13052-13068 (2003 Ed.)

0458 and lines 12976-12992 (2001 Ed.) with it margin marked as THR and shaded.

0459 Rationale: The intent is to allow pthread_t to be a structure.

0460 **Change Number:** XBD/TC2/D6/27 [XBD ERN 38]

0461 On Page: 373 Line: 13189-13190 Section: sys/utsname.h (2003 Ed.)

0462 On Page: 371 Line: 13113-13114 Section: sys/utsname.h (2001 Ed.)

0463 Change From:

0464 "char nodename[] Name of this node within an implementation-defined

0465 communications network. "

0466 To:

0467 "char nodename[] Name of this node within the communications

0468 network to which this node is attached, if any."

0469 **Change Number:** XBD/TC2/D6/28 [XSH ERN 53]

0470 On Page: 398 Line: 14041-14044 Section: ucontext.h (2003 Ed.)

0471 On Page: 396 Line: 13960-13963 Section: ucontext.h (2001 Ed.)

0472 Margin mark OB (obsolescent) and shade the getcontext(), makecontext(),

0473 setcontext(), swapcontext() functions.

Technical Corrigendum 2

0474 **Change Number:** XBD/TC2/D6/29 [XBD ERN 16]

0475 On Page: 401-404 Line: 14130-14133 Section: unistd.h (2003 Ed.)

0476 On Page: 399-403 Line: 14049-14052 Section: unistd.h (2001 Ed.)

0477 Change From:

0478 `"_POSIX_CHOWN_RESTRICTED`

0479 The use of `chown()` and `fchown()` is restricted to a process with
0480 appropriate privileges, and to changing the group ID of a file only to
0481 the effective group ID of the process or to one of its supplementary
0482 group IDs. This symbol shall always be set to a value other than `-1`."

0483 To:

0484 `"_POSIX_CHOWN_RESTRICTED`

0485 The use of `chown()` and `fchown()` is restricted to a process with
0486 appropriate privileges, and to changing the group ID of a file only to
0487 the effective group ID of the process or to one of its supplementary
0488 group IDs. This symbol shall either be undefined or defined with a value
0489 other than `-1`."

0490 On lines 14167-14169 (2003 Ed.); lines 14083-14085 (2001 Ed.)

0491 Change from:

0492 `"_POSIX_NO_TRUNC`

0493 Pathname components longer than `{NAME_MAX}` generate an
0494 error. This symbol shall always be set to a value other than `-1`."

0495 To:

0496 `"_POSIX_NO_TRUNC`

0497 Pathname components longer than `{NAME_MAX}` generate an error. This symbol
0498 shall either be undefined or defined with a value other than `-1`."

0499 For the Constants for Options and Option Groups on pages 401-406 lines

0500 14121-14350 (2003 Ed.); pages 399-404 lines 14040-14266 (2001 Ed.):

0501 Change occurrences of the text From:

0502 "If this symbol has a value other than `-1` or `0`, it shall have the
0503 value `200112L`."

0504 To:

0505 "If this symbol is defined in `<unistd.h>`, it shall be defined to be `-1`,
0506 `0`, or `200112L`. The value of this symbol reported by `sysconf()` shall
0507 either be `-1` or `200112L`."

0508 Rationale:

0509 This further clarifies the requirements on when constants in `unistd.h`
0510 can be defined or undefined.

0511 **Change Number:** XBD/TC2/D6/30 [XSH ERN 106]

0512 On Page: 405 Line: 14309-14320 Section: unistd.h (2003 Ed.)

0513 On Page: 403 Line: 14225-14236 Section: unistd.h (2003 Ed.)

0514 In the DESCRIPTION section, line 14309 (2003 Ed.); line 14225 (2001 Ed.)

0515 Change From:

0516 "_V6_ILP32_OFF32"

0517 To:

0518 "_POSIX_V6_ILP32_OFF32"

0519 In the DESCRIPTION section, line 14312 (2003 Ed.); line 14228 (2001 Ed.)

0520 Change From:

0521 "_V6_ILP32_OFFBIG"

0522 To:

0523 "_POSIX_V6_ILP32_OFFBIG"

0524 In the DESCRIPTION section, line 14315 (2003 Ed.); line 14231 (2001 Ed.)

0525 Change From:

0526 "_V6_LP64_OFF64"

0527 To:

0528 "_POSIX_V6_LP64_OFF64"

0529 In the DESCRIPTION section, line 14318 (2003 Ed.); line 14234 (2001 Ed.)

0530 Change From:

0531 "_V6_LPBIG_OFFBIG"

0532 To:

0533 "_POSIX_V6_LPBIG_OFFBIG"

0534 Rationale: Consistency with the sysconf() and c99 man pages.

0535 **Change Number:** XBD/TC2/D6/31 [XCU ERN 19]

0536 On Page: 408 Line: 14437-14440 Section: unistd.h (2003 Ed.)

0537 On Page: 406 Line: 14350-14353 Section: unistd.h (2001 Ed.)

0538 Change From:

0539 "This value is a <newline>-separated list of names of programming

0540 environments supported by the implementation in which the widths of

0541 the blksize_t, cc_t, mode_t, nfd_t, pid_t, ptrdiff_t, size_t, speed_t,

0542 ssize_t, suseconds_t, tcflag_t, useconds_t, wchar_t, and wint_t types

0543 are no greater than the width of type long."

0544 To:

0545 "This value is a <newline>-separated list of names of programming

0546 environments supported by the implementation in which the widths of

0547 the blksize_t, cc_t, mode_t, nfd_t, pid_t, ptrdiff_t, size_t, speed_t,

0548 ssize_t, suseconds_t, tcflag_t, useconds_t, wchar_t, and wint_t types

0549 are no greater than the width of type long. The format of each name

0550 shall be suitable for use with the getconf -v option."

0551 **Change Number:** XBD/TC2/D6/32 [XSH ERN 40,61,62]

0552 On Page: 409-411 Line: 14525,14492,14610 Section: unistd.h (2003 Ed.)

0553 On Page: 407-409 Line: 14438,14405,14520 Section: unistd.h (2001 Ed.)

0554 Delete "_SC_FILE_LOCKING"

0555 Delete "_SC_2_C_VERSION"

0556 Delete "_SC_XOPEN_XCU_VERSION"

Technical Corrigendum 2

0557 **Change Number:** XBD/TC2/D6/33 [XBD ERN 17]

0558 On Page: 410 Line: 14564,14586,14588 Section: unistd.h (2003 Ed.)

0559 On Page: 408 Line: 14475,14496,14498 Section: unistd.h (2001 Ed.)

0560 Add to the list of symbolic constants for sysconf()

0561 _SC_SS_REPL_MAX, _SC_TRACE_EVENT_NAME_MAX, _SC_TRACE_NAME_MAX,

0562 _SC_TRACE_SYS_MAX _SC_TRACE_USER_EVENT_MAX

0563 Rationale: These symbols were omitted.

0564 **Change Number:** XBD/TC2/D6/34 [XBD ERN 10]

0565 On Page: 413 Line: 14692 Section: unistd.h (2003 Ed.)

0566 On Page: 411 Line: 14602 Section: unistd.h (2001 Ed.)

0567 Remove the XSI margin marker and remove the shading

0568 from the following function:

0569 "int symlink(const char *, const char *);"

0570 Rationale: This was inconsistent with the XSH volume.

0571 **Change Number:** XBD/TC2/D6/35 [XBD ERN 11]

0572 On Page: 413 Line: 14471 Section: unistd.h (2003 Ed.)

0573 On Page: 404 Line: 14384 Section: unistd.h (2001 Ed.)

0574 Insert at the start of the symbolic constants list

0575 for pathconf() (before _PC_ALLOC_SIZE_MIN):

0576 "_PC_2_SYMLINKS"

0577 Rationale: This corresponds to the definition of POSIX2_SYMLINKS in

0578 the Shell and Utilities volume.

3. Changes to System Interfaces

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

0579 **Change Number:** XSH/TC2/D6/1 [XBD ERN 20]

0580 On Page: xxx Line: "ISO/IEC 8859" Section: Referenced Documents (2003 Ed.)

0581 On Page: xxvi Line: "ISO/IEC 8859" Section: Referenced Documents (2001 Ed.)

0582 Add after line starting "Part 10":

0583 "Part 11: Latin/Thai Alphabet"

0584 Add after line starting "Part 15":

0585 "Part 16: Latin Alphabet No. 10"

0586 **Change Number:** XSH/TC2/D6/2 [XSH ERN 167, TC2d5 ERN 9]

0587 On Page: 15,19 Line: 580-583,713-714 Section: 2.2.2 (2003 Ed.)

0588 On Page: 15,19 Line: 571-574,699-700 Section: 2.2.2 (2001 Ed.)

0589 On lines 580-583 (2003 Ed.); lines 571-574 (2001 Ed.)

0590 Change From:

0591 "Implementations may add symbols to the headers shown in the following

0592 table, provided the identifiers for those symbols begin with the

0593 corresponding reserved prefixes in the following table, and do not use

0594 the reserved prefixes `posix_`, `POSIX_`, or `_POSIX_`."

0595 To:

0596 "Implementations may add symbols to the headers shown in the following

0597 table, provided the identifiers for those symbols either:

0598 1. begin with one of the corresponding reserved prefixes in the table; or

0599 2. have one of the corresponding complete names in the table; or

0600 3. end in the string indicated as a reserved suffix in the table and

0601 do not use the reserved prefixes `posix_`, `POSIX_`, or `_POSIX_`, as long as

0602 the reserved suffix is in that part of the name considered significant

0603 by the implementation.

0604 Symbols that use the reserved prefix `_POSIX_` may be made visible by

0605 implementations in any header defined by IEEE Std 1003.1-2001."

0606 On lines 713-714 (2003 Ed.); lines 699-700 (2001 Ed.):

0607 Change From:

0608 "1. All identifiers that begin with an underscore and either an uppercase

0609 letter or another underscore are always reserved for any use by the

0610 implementation."

0611 To:

0612 "1. With the exception of identifiers beginning with the prefix `_POSIX_`,

0613 all identifiers that begin with an underscore and either an uppercase

0614 letter or another underscore are always reserved for any use by the

0615 implementation."

0616 Rationale:

0617 This change permits implementations to have symbols with the prefix

0618 `_POSIX_` visible in any header. A change in TC1 had disallowed this.

Technical Corrigendum 2

0619 **Change Number:** XSH/TC2/D6/3 [XSH ERN 1]

0620 On Page: 18 Line: 668 Section: 2.2.2 (2003 Ed.)

0621 On Page: 18 Line: 659 Section: 2.2.2 (2001 Ed.)

0622 Add the following line into the table after line 668 (2003 Ed.);

0623 after line 659 (2001 Ed.):

0624 "<math.h> FP_[A-Z] "

0625 Rationale:

0626 This text is added for consistency with the math.h page in the

0627 Base Definitions volume which states "Additional implementation-defined

0628 floating-point classifications, with macro definitions beginning with FP_

0629 and an uppercase letter, may also be specified by the implementation."

0630 **Change Number:** XSH/TC2/D6/4 [XSH ERN 99]

0631 On Page: 33 Line: 1337 Section: 2.4.3 (2003 Ed.)

0632 On Page: 33 Line: 1339 Section: 2.4.3 (2001 Ed.)

0633 Add "socketatmark()" to the table in 2.4.3 (the list of functions that shall

0634 be either reentrant or non-interruptible by signals and shall be

0635 async-signal-safe).

0636 Rationale:

0637 This was an omission when the socketatmark() function was added.

0638 **Change Number:** XSH/TC2/D6/5 [XSH ERN 109]

0639 On Page: 55 Line: 2267 Section: 2.9.5.2 (2003 Ed.)

0640 On Page: 55 Line: 2268 Section: 2.9.5.2 (2001 Ed.)

0641 Add "fdatasync()" into the table of functions after "fcntl()".

0642 Rationale:

0643 This was an omission. Since fdatsync() is often implemented

0644 in terms of fsync() it should also be in this list.

0645 **Change Number:** XSH/TC2/D6/6 [XSH ERN 76]

0646 On Page: 56 Line: 2277 Section: 2.9.5.2 Cancellation Points (2003 Ed.)

0647 On Page: 56 Line: 2278 Section: 2.9.5.2 Cancellation Points (2001 Ed.)

0648 Insert into the list of functions that "may have" a cancellation
0649 point (the second list of functions in this section)
0650 in the appropriate order:

0651 "access
0652 asctime
0653 asctime_r
0654 ctime
0655 ctime_r
0656 fmtmsg
0657 fpathconf
0658 fstat
0659 getaddrinfo
0660 gethostid
0661 getnameinfo
0662 getopt [with footnote if opterr is nonzero]
0663 link
0664 localtime
0665 localtime_r
0666 lstat
0667 mktime
0668 pathconf
0669 posix_openpt
0670 stat
0671 strerror_r
0672 strftime
0673 symlink
0674 sync
0675 tzset
0676 wcsftime
0677 wordexp"

0678 **Change Number:** XSH/TC2/D6/7 [XSH ERN 77]

0679 On Page: 57 Line: 2336-2353 Section: 2.9.5.3 (2003 Ed.)

0680 On Page: 57 Line: 2337-2354 Section: 2.9.5.3 (2001 Ed.)

0681 Change From:

0682 "Each thread maintains a list of cancellation cleanup handlers. The
0683 programmer uses the pthread_cleanup_push() and pthread_cleanup_pop()
0684 functions to place routines on and remove routines from this list.

0685 When a cancellation request is acted upon, the routines in the list
0686 are invoked one by one in LIFO sequence; that is, the last routine
0687 pushed onto the list (Last In) is the first to be invoked (First
0688 Out). The thread invokes the cancellation cleanup handler with
0689 cancellation disabled until the last cancellation cleanup handler
0690 returns. When the cancellation cleanup handler for a scope is
0691 invoked, the storage for that scope remains valid. If the last
0692 cancellation cleanup handler returns, thread execution is terminated
0693 and a status of PTHREAD_CANCELED is made available to any threads
0694 joining with the target. The symbolic constant PTHREAD_CANCELED
0695 expands to a constant expression of type (void *) whose value
0696 matches no pointer to an object in memory nor the value NULL.

0697 The cancellation cleanup handlers are also invoked when the thread
0698 calls pthread_exit().

0699 A side effect of acting upon a cancellation request while in a
0700 condition variable wait is that the mutex is re-acquired before
0701 calling the first cancellation cleanup handler. In addition, the
0702 thread is no longer considered to be waiting for the condition and
0703 the thread shall not have consumed any pending condition signals on
0704 the condition.

Technical Corrigendum 2

0705 A cancellation cleanup handler cannot exit via `longjmp()` or
0706 `siglongjmp()`."

0707 To:

0708 Each thread maintains a list of cancellation cleanup handlers. The
0709 programmer uses the `pthread_cleanup_push()` and `pthread_cleanup_pop()`
0710 functions to place routines on and remove routines from this list.

0711 When a cancellation request is acted upon, or when a thread calls
0712 `pthread_exit()`, the thread first disables cancellation by
0713 setting its cancelability state to `PTHREAD_CANCEL_DISABLE` and its
0714 cancelability type to `PTHREAD_CANCEL_DEFERRED`. The cancelability
0715 state shall remain set to `PTHREAD_CANCEL_DISABLE` until the thread
0716 has terminated. The behavior is undefined if a cancellation cleanup
0717 handler or thread-specific data destructor routine changes the
0718 cancelability state to `PTHREAD_CANCEL_ENABLE`.

0719 The routines in the thread's list of cancellation cleanup handlers
0720 are invoked one by one in LIFO sequence; that is, the last routine
0721 pushed onto the list (Last In) is the first to be invoked (First
0722 Out). When the cancellation cleanup handler for a scope is invoked,
0723 the storage for that scope remains valid. If the last cancellation
0724 cleanup handler returns, thread-specific data destructors (if any)
0725 associated with thread-specific data keys for which the thread has
0726 non-NULL values will be run, in unspecified order, as described for
0727 `pthread_key_create()`.

0728 After all cancellation cleanup handlers and thread-specific data
0729 destructors have returned, thread execution is terminated. If the
0730 thread has terminated because of a call to `pthread_exit()`, the
0731 `value_ptr` argument is made available to any threads joining with the
0732 target. If the thread has terminated by acting on a cancellation
0733 request, a status of `PTHREAD_CANCELED` is made available to any
0734 threads joining with the target. The symbolic constant
0735 `PTHREAD_CANCELED` expands to a constant expression of type `(void *)`
0736 whose value matches no pointer to an object in memory nor the value
0737 `NULL`.

0738 A side effect of acting upon a cancellation request while in a
0739 condition variable wait is that the mutex is re-acquired before
0740 calling the first cancellation cleanup handler. In addition, the
0741 thread is no longer considered to be waiting for the condition and
0742 the thread shall not have consumed any pending condition signals on
0743 the condition.

0744 A cancellation cleanup handler cannot exit via `longjmp()` or
0745 `siglongjmp()`."

0746 **Change Number:** XSH/TC2/D6/8 [XSH ERN 89]

0747 On Page: 58 Line: 2379 Section: Threads (2003 Ed.)

0748 On Page: 58 Line: 2380 Section: Threads (2001 Ed.)

0749 Insert new section:

0750 "2.9.8 Use of Application-Managed Thread Stacks

0751 An "application-managed thread stack" is a region of memory allocated
0752 by the application, for example, memory returned by the `malloc()`
0753 or `mmap()` functions, and designated as a stack through the act of
0754 passing an address related to that memory as the "stackaddr" argument to
0755 `pthread_attr_setstackaddr()` (obsolete) or by passing the address and size
0756 of the stack, respectively, as the `stackaddr` and `stacksize` arguments to
0757 `pthread_attr_setstack()`. Application-managed stacks allow the application
0758 to precisely control the placement and size of a stack.

0759 The application grants to the implementation permanent ownership of and
0760 control over the application-managed stack when the attributes object in
0761 which the `stack` or `stackaddr` attribute has been set is used, either by
0762 presenting that attributes object as the 'attr' argument in a call to
0763 `pthread_create()` that completes successfully, or by storing a pointer to

0764 the attributes object in the sigev_notify_attributes member of a 'struct
 0765 sigevent' and passing that 'struct sigevent' to a function accepting
 0766 such argument that completes successfully. The application may
 0767 thereafter utilize the memory within the stack only within the normal
 0768 context of stack usage within or properly synchronized with a thread
 0769 that has been scheduled by the implementation with stack pointer
 0770 value(s) that are within the range of that stack. In particular, the
 0771 region of memory cannot be freed, nor can it be later specified as the
 0772 stack for another thread.

0773 When specifying an attributes object with an application-managed stack
 0774 through the sigev_notify_attributes member of a 'struct sigevent', the
 0775 results are undefined if the requested signal is generated multiple
 0776 times (as for a repeating timer).

0777 Until an attributes object in which the stack or stackaddr attribute has
 0778 been set is used, the application retains ownership of and control over
 0779 the memory allocated to the stack. It may free or reuse the memory as
 0780 long as it either deletes the attributes object, or before using the
 0781 attributes object replaces the stack by making an additional call to the
 0782 same function, either pthread_attr_setstackaddr() or
 0783 pthread_attr_setstack(), that was used originally to designate the stack.
 0784 There is no mechanism to retract the reference to an application-managed
 0785 stack by an existing attributes object.

0786 Once an attributes object with an application-managed stack has been
 0787 used, that attributes object cannot be used again by a subsequent call
 0788 to pthread_create() or any function accepting a 'struct sigevent' with
 0789 sigev_notify_attributes containing a pointer to the attributes object,
 0790 without designating an unused application-managed stack by making an
 0791 additional call to the function originally used to define the stack,
 0792 pthread_attr_setstack() or pthread_attr_setstackaddr()."

0793 **Change Number:** XSH/TC2/D6/9 [XSH ERN 138]

0794 On Page: 92 Line: 3558-3559 Section: abort (2003 Ed.)
 0795 On Page: 92 Line: 3559-3560 Section: abort (2001 Ed.)

0796 In the APPLICATION USAGE section
 0797 Change From:
 0798 "Catching the signal is intended to provide the application writer with
 0799 a portable means to abort processing, free from possible interference
 0800 from any implementation-defined functions."

0801 To:
 0802 "Catching the signal is intended to provide the application writer with
 0803 a portable means to abort processing, free from possible interference
 0804 from any implementation-supplied functions."

0805 **Change Number:** XSH/TC2/D6/10 [XSH ERN 2]

0806 On Page: 105 Line: 3919 Section: aio_cancel (2003 Ed.)
 0807 On Page: 104 Line: 3898 Section: aio_cancel (2001 Ed.)

0808 In the RETURN VALUE section
 0809 Change From:
 0810 "The aio_cancel() function shall return the value AIO_CANCELED to the
 0811 calling process if the requested operation(s) were canceled."

0812 To:
 0813 "The aio_cancel() function shall return the value AIO_CANCELED
 0814 if the requested operation(s) were canceled."

0815 Rationale:
 0816 The use of the term "to the calling process" was unnecessary and
 0817 incorrect.